ANALYTIC METHODS IN FINANCE WITH APPLICATIONS TO PORTFOLIO AND RISK MANAGEMENT

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Supervisor

Dr. Nuttanan Wichitaksorn

Prof. Jiling Cao

April 2023

By

Rewat Khanthaporn

School of Engineering, Computer and Mathematical Sciences

Abstract

The thesis studies and develops an investment portfolio strategy using a regular-vine-based forecasting model in period of the recent COVID-19 crisis. The model parameter estimation technique uses families of Bayesian inference and variational Bayes inference. The optimisation model uses family of machine learning algorithms. Overall, the thesis comprises three papers in which the ultimate outcome in paper three is the solution of dynamic portfolio allocation. Prior to that, the first two papers develop the multivariate asset returns forecasting models using the inference function of margins method and then apply it to the third paper in a portfolio optimisation model. The full details of each paper are provided in their abstracts: Chapter 3 for paper one, Chapter 4 for paper two and Chapter 5 for paper three. A brief outline of the three papers can be stated as follows.

The first paper studies a univariate forecasting model using the hybrid of asymmetric generalised autoregressive conditional heteroskedasticity and intertemporal capital asset pricing, with the following innovations: (1) a mixture of two generalised Pareto distributions and a Gaussian distribution; and (2) generalised error distribution. The Griddy Gibbs sampling algorithm in the Bayesian Markov chain Monte Carlo with parallel computing is used for the model parameter estimation. The study demonstrates the proposed model and estimation method through both simulation and empirical experiments among the benchmarks. It proves that the proposed model statistically outperforms competing models in the return forecasting under the conditions of market turmoil during the COVID-19 period.

The second paper extends the first paper from the univariate forecasting model to a

multivariate forecasting model using high dimensional data and up to 100 dimensions where the comovement model is a regular vine model. The paper initiates a magnitude 13 bivariate copula candidate for the pair structure well-known in the literature of quantitative risk management. While the estimation techniques for the current paper explore another Bayesian Markov chain Monte Carlo, which is random-walk Metropolis-Hasting sampling, and, in Bayesian machine learning, variational Bayes with (and without) latent variables and data augmentation. Both simulation results and empirical results show satisfactory outcomes, since the proposed model and its estimation can outperform the traditional model.

The third paper extends multivariate regular vine forecasting model to the problem of dynamic optimal asset allocation in variate optimisation models. The study introduces evolutionary optimisation algorithms, including a genetic algorithm and a clonal selection algorithm, to optimisation problems. There are two main scenarios in optimisation problems which correspond to three model performance indicators: (1) the reward-risk indicator, (2) the diversity indicator, and (3) the convergence indicator. In addition, stock selection analysis is also applied to the optimisation problem. The empirical studies show that the proposed vine-copula-based forecasting model performs well in optimisation problems in terms of performance measures. Furthermore, based on the scenario experiment, the paper mathematically reveals that the financial market dependence structure has been disrupted as if a new normal has been established since the impact of the COVID-19 pandemic.

Contents

List of Tables

List of Figures

Attestation of Authorship

I hereby declare that this submission is my own work and

Signature of candidate

Co-authored Works

The contributions of the co-authors for the following chapters/papers are indicated below:

Chapter 3, published in the Journal of Applied Economics:

Khanthaporn, R. and Wichitaksorn, N. (2022). Modelling and forecasting COVID-19 stock returns using asymmetric GARCH-ICAPM with mixture and heavy-tailed distributions. *Applied Economics*, 1-20.

- 85%: Rewat Khanthaporn
- 15%: Dr. Nuttanan Wichitaksorn

Chapter 4, pending submission:

Khanthaporn, R. and Wichitaksorn, N. (2022). Variational Bayes with Latent Variables and Data Augmentation for High Dimensional Mixed Regular Vine Dependence Structure and Mixture Innovation Model in Financial Time Series.

- 85%: Rewat Khanthaporn
- 15%: Dr. Nuttanan Wichitaksorn

Chapter 5, pending submission:

Khanthaporn, R. and Wichitaksorn, N. (2022). Regular Vine Copula Model for Investment Portfolio Optimisation in Turbulent Financial Markets using Computational Intelligence.

- 85%: Rewat Khanthaporn
- 15%: Dr. Nuttanan Wichitaksorn

We, the undersigned, hereby agree to the percentage of participation to the chapters/papers identified above.

Rewat Khanthaporn

Dr. Nuttanan Wichitaksorn

Publications

Khanthaporn, R. and Wichitaksorn, N. (2022). Modelling and forecasting COVID-19 stock returns using asymmetric GARCH-ICAPM with mixture and heavy-tailed distributions. *Applied Economics*, 1-20.

CONFERENCE PRESENTATION

AUT Mathematical Sciences Symposium [2018] [2019] [2023]

The Junior International Society for Bayesian Analysis Conference (j-ISBA) [2021]

Australia and New Zealand Statistical Conference (ANZSC) [2021]

Mathematics and Statistics Postgraduate Conference (NZMSP) [2019] [2020]

Acknowledgements

I would like to acknowledge many people who supported me in the preparation of thesis. First, I would like to express my sincere gratitude to my supervisors, Dr Nuttanan Wichitaksorn and Professor Jiling Cao, who guided me through the current thesis during regular supervisory meetings, particularly during the difficult times of the COVID-19 pandemic. My supervisors were not only supportive from an academic perspective but also in the matter of finance. It has been my pleasure to work with them and I have learned such a lot from them, especially Dr Nuttanan Wichitaksorn, who was always patient and provided in-depth comments and thesis direction. I also wish to thank the anonymous journal reviewers and editors of Applied Economics who provided critical comments and made sure that my first paper was presented based on factual details.

I would like to thank my lovely family for their full support in both finance and mental health. Again, thank you to my dear wife (Eve), my lovely sister (Oil) and my big brother (Tee); and special thanks to my Thai community who always cheered up and supported me with food during the study.

Last, but not least, to my mum and dad who were always my inspiration even though they passed away, but were always in my heart with love. Without them I would not have been able to complete my doctoral studies.

Rewat Khanthaporn
April 2023

Dedication

I dedicate this thesis to my super mum and dad who are always in my heart.

Part I Theoretical Framework

Chapter 1

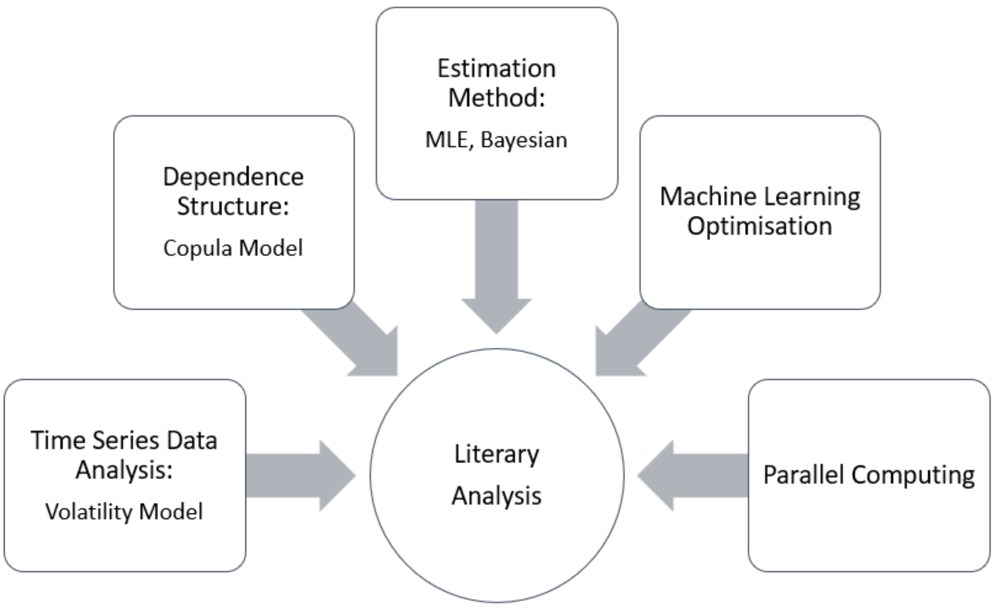
Introduction

This first paper introduces the theoretical framework and the important literature related to analytic methods in finance with applications to portfolio and risk management. Section 1.1 introduces an essential concept of modern portfolio optimisation, the problem of mean-variance portfolio optimisation (MVPO). The next two sections present the literature on the mathematical models to capture the important facts of financial time series data, which leads to the development of the proposed forecasting models of this thesis. More precisely, the forecasting model is one of the most important elements in the problem of MVPO since it is a significant influence on the solution of the optimisation problem. There are two components in the forecasting model, the univariate model and the multivariate model. Therefore, Section 1.2 covers the literature on the univariate model, especially, the family of conditional generalised autoregressive conditional heteroscedasticity (GARCH). Then, in Section 1.3, the literature review focuses on the multivariate model, especially the copula dependence model.

In Section 1.4, the chapter introduces the literature of another crucial component of a forecasting model, the model parameter estimation technique, which is also a massive and active research area. There are wide literature related to estimation methods. One of the most popular estimation methods is the maximum likelihood method while one of the alternative estimation methods is Bayesian inference. Hence, it is presented in

Section 1.4, where Subsection 1.4.1 is the literature on the maximum likelihood method, Subsection 1.4.2 covers Bayesian inference, finally, Subsection 1.4.3 is the demonstration of some relevant time series models and estimation method performance measures in the financial literature. Then, Section 1.5 introduces the literature on machine learning algorithms in investment portfolio optimisation models. Finally, the literature on the parallel computing technique is presented in Section 1.6, where it is an essential algorithm to reduce computational time in the complicated algorithms described in Section 1.4 and Section 1.5. See Figure 1.1 for the overall framework of Chapter 1 regarding the relevant theoretical framework and the literature on analytic methods in finance with applications to portfolio and risk management.

Figure 1.1: Literary Analysis in Quantitative Portfolio Risk Management



1.1 Modern Portfolio Theory

All securities carry with them some level of risk. In general, in the financial market, the higher the risk, the greater the expected asset returns. By definition, risk management is a process of identifying, analysing and addressing potential risks in order to achieve investment objectives at the end of the investment period. This is the main purpose of portfolio risk management, to ensure that the potential losses in investment never exceed acceptable boundaries or are not over the level of risk tolerance of investors.

Another important factor of portfolio risk management is that it can increase or decrease risk depending on the investment objectives. For example, the risk level in an investment portfolio is adjustable depending on the investors' risk tolerance; indeed, the higher the risk level the investors perceive, the greater the expected return the investors should obtain.

Risk is "the possibility of something bad happening" or "something bad that might happen" (*Cambridge Advanced Learner's Dictionary*, 2013). In financial institutions, there are two well-known risk measures including value at risk (VaR) and conditional value at risk (CVaR). VaR is the maximum amount of potential loss at the end of an investment period with a given probability. In statistics, VaR at level $\alpha \in (0, 1)$ is the $1 - \alpha$ quantile of the loss distribution or

$$VaR_{1-\alpha} = q_u(F_L)$$

where $q_u(F_L)$ is the quantile function of F_L. For instance, let us assume the asset return is normally distributed and $\alpha = 0.05$. VaR at $\alpha = 0.05$ for the investment portfolio is

$$\mathrm{VaR}_{0.05} = \ Amount\ of\ position * (1.65\sigma_p + \overline{R}_p).$$

Then assume that the expected monthly portfolio return $\overline{R}_p$ is 2.24%, the monthly portfolio risk σ_p is 14.98% and the investment amount is \$1,000. Therefore, $\mathrm{VaR}_{0.05}$ for a month is \$269.58. This means that there is only a 5% chance that the investment loss will be more than \$269.58 and the corresponding $\mathrm{VaR}_{0.05}$ for the 12- month investment is \$933.82. While CVaR or ES (Expected Shortfall) is the expectation of the tail of the loss distribution above the $1 - \alpha$ level of confidence. Hence, CVaR can be written in the form of

$$\mathrm{CVaR}_{1-\alpha} = \frac{1}{\alpha}\int_{1-\alpha}^{1} q_u(F_L)du$$

For example, when the asset return is normally distributed, the monthly CVaR_α is

$$\mathrm{CVaR}_\alpha = \mathit{Amount\ of\ position} * \left(\frac{f(\mathrm{VaR}_\alpha)}{\alpha}\sigma_p + \overline{R}_p\right)$$

where $f(\cdot)$ is a normal density function. Therefore, $\mathrm{CVaR}_{0.05}$ for a month is \$331.39.

In modern portfolio theory, the objective function of MVPO is to maximise the investment portfolio return and simultaneously minimise investment portfolio risk. Risk must be considered in the investment portfolio and can be measured by standard deviation. Alternatively, VaR and CVaR could be parallelly calculated. The characteristics of a portfolio, in general, are the standard deviation and expected return of the portfolio that, are respectively, given by

$$\begin{aligned}
\sigma_p &= \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{ij}} \\
&= \sqrt{\sum_{i=1}^{n} w_i^2\sigma_i^2 + \sum_{i=1, i\neq j}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{ij}} \\
&= \sqrt{\sum_{i=1}^{n} w_i^2\sigma_i^2 + 2\sum_{i=1, i>j}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{ij}} \\
\overline{R}_p &= \sum_{i=1}^{n} w_i E(R_i)
\end{aligned}$$

where n is the number of assets in the investment portfolio, w_i is the proportion of the i^{th} assets in the portfolio p, $E(R_i)$ are the expected rates of return of the i^{th} assets, and σ_{ij} is the covariance between the returns of the assets i^{th} and j^{th}. ρ_{ij} is the correlation coefficient between the returns of the assets i^{th} and j^{th}. The correlation coefficient is defined as

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i\sigma_j}.$$

We can also write the standard deviation and the expected returns of the portfolio in matrix form:

$$\sigma_p = \sqrt{W'\Omega W},$$
$$\overline{R}_p = W'\overline{R}_i,$$

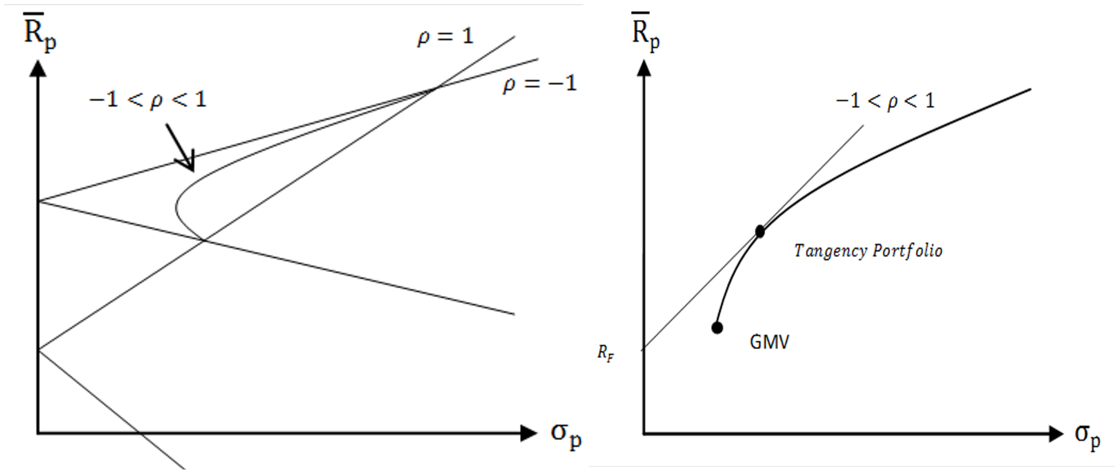
where W is a weight column vector of the assets in the portfolio, $\overline{R}_i$ is the expected returns row vector of the assets in the portfolio and Ω is an n by n variance-covariance matrix of the asset returns in the portfolio.

The study requires the investor to invest all the money in the portfolio, $\sum w = 1$, when there are no short sales. In the case of $-1 < \rho_{i,j} < 1$ and the simplest two assets of the portfolio, the relationship is not linear. It is a hyperbolic function, therefore, it is no longer possible to create a riskless portfolio in order that the portfolio with no short selling has a minimum volatility, where $\sigma_p > 0$. We can differentiate σ_p^2 with respect to W_i, i.e., W_1. The optimal portfolio weight of W_1^* is

$$\frac{d\sigma_p^2}{dW_1} = 0 \Rightarrow W_1^* = \frac{\sigma_2^2 - \rho_{12}\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2 - 2\rho_{12}\sigma_1\sigma_2}.$$

Note that the more negative the correlation coefficient, the larger the diversification benefits. Similarly, this can be applied to a general case with n risky assets portfolio optimisation problems by differentiating σ_p^2 with respect to W_i and then obtaining an efficient portfolio frontier with n risky assets. For any combinations of the two assets in the portfolio, rational investor would choose a portfolio represented by the positively sloped lines, because it gives the highest expected return with the lowest level of volatility. Therefore, it is called the efficient mean-variance portfolio set and at a point that gives the lowest level of volatility is called the global minimum variance (GMV) of the portfolio as shown in Figure 1.2.

Figure 1.2: Efficient Mean-Variance Portfolio Set



When short selling is allowed in an investment portfolio, it means that an investor can often sell an asset that he or she does not own. This process involves taking a negative fraction in an asset. However, a portfolio is where a short sell is allowed, in practice, most institutional investors do not short sell and many institutions are even forbidden to do this by law in many countries, therefore, this thesis will not consider this in this study. In the next section, the study introduces the concepts and literature review of univariate model in forecasting financial data, where mainly focused on the GARCH family.

1.2 Conditional Volatility: GARCH Family and Related Models

The conditional volatility model is a crucial model in financial time series analysis. Clearly, the (generalised) autoregressive conditional heteroscedasticity model, (G)ARCH family, is one of the most important in the conditional volatility models since Engle

(1982) introduced the ARCH model with the following specification:

$$y_t = \varepsilon_t = \sigma_t z_t$$
$$\sigma_t^2 = \xi + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2$$

where t represents a time index variable, $\varepsilon_t \sim N(0, \sigma_t^2)$, $z_t \sim i.i.d.N\,(0, 1)$ or independently and identically distributed standard normal, $q > 0$, $\xi > 0$, $\alpha_i \geq 0$. Bollerslev (1986) extended the model to the generalised ARCH(p,q) or GARCH(p,q) model,

$$\sigma_t^2 = \xi + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2,$$

where $\varepsilon_t \sim N(0, \sigma_t^2),\ p \geq 0, q > 0,\ \xi > 0,\ \alpha_i \geq 0, \beta_j \geq 0,\ 0 \leq \alpha_i + \beta_j \leq 1,\ 1 \leq i \leq q$ and $1 \leq j \leq p$ to ensure that the conditional variance σ_t^2 is almost surely strictly positive or a strictly stationary process. The estimated parameters from y_t are $\hat{\theta} = \{\hat{\xi}, \hat{\alpha_i}, \hat{\beta}_j, 1 \leq i \leq q, 1 \leq j \leq p\}$. Since then, the GARCH model has been evolving due to the properties that fulfill the variety of probability distributions of financial asset returns. Manifold extensions of the GARCH family have been proposed to capture the important financial market characteristics: volatility clustering, non-normal, fat-tail, and asymmetrically distributed properties; see Engle (2004) for a survey.

Among others, Nelson (1991) suggests the exponential GARCH (EGARCH) model while the GJR-GARCH is introduced by Glosten, Jagannathan and Runkle (1993). Zakoian (1994) proposes the threshold GARCH (TGARCH) whereas Bildirici and Ersin (2009) study the GARCH family models, including GARCH, EGARCH, GJR-GARCH, TGARCH, Nonlinear-GARCH (NGARCH), Simple-Asymmetric-GARCH (SA-GARCH), Power-GARCH (PGARCH), Asymmetric-Power-GARCH (APGARCH), and Nonlinear-Power-GARCH (NPGARCH). The models are also extended to an Artificial Neural Network called ANN-GARCH-type family.

The autoregressive moving average (ARMA) model is one of the most well-known time series models. The ARMA model attempts to capture more of the serial correlation present within financial asset returns. The AR part involves the variable of regressing on its own lagged time. The MA part involved the error term of modelling as a linear function at various times in the past. However, the ARMA model is not conditionally heteroskedastic as it does not take into account volatility clustering property in financial asset returns. Many researchers therefore combine the ARMA model and the GARCH model because the GARCH model fulfills volatility clustering property, so is called the ARMA-GARCH model. For instance, Mendes et al. (2016) examine the ARMA-GARCH model to cope with the multiobjective portfolio optimisation problem. The univariate ARMA(r,m) model has the following specification:

$$y_t = c + \sum_{i=1}^{r} \varphi_i y_{t-r} + \sum_{j=1}^{m} \omega_j \varepsilon_{t-m} + \varepsilon_t$$

where r and m represent the orders of the autoregressive and moving average models, respectively, $\varepsilon_t \sim N(0, \sigma_t^2)$. For instance, the return of i^{th} asset is a particular univariate ARMA(1,1)-GARCH(1,1) process and is demonstrated as

$$y_t = c + \varphi_1 y_{t-1} + \omega_1 \varepsilon_{t-1} + \varepsilon_t$$

$$\sigma_t^2 = \xi + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2.$$

The estimated parameters of y_t are $\hat{\theta} = \{\hat{c}, \hat{\varphi_1}, \hat{\omega_1}, \hat{\xi}, \hat{\alpha_1}, \hat{\beta_1}\}$.

1.2.1 Conditional Volatility with Non-Normal Innovation

Cai (1994) and Hamilton and Susmel (1994) initiate ARCH and regime switching (MS-ARCH) models to financial time series analysis. The assumption of the regime approach is based on different circumstances in the financial market. Therefore, the volatility model with only a single regime might affect the precision of the volatility estimate and forecast, for example, during a crisis and a non-crisis. These are two regimes model. Later, a tractable MS-GARCH model is introduced in Gray (1996) and

Bauwens, Preminger and Rombouts (2010) also propose the MS-GARCH model where the main difference is that each finite regime has the constant conditional mean and the constant conditional variance. Billio, Casarin and Osuntuyi (2016) again propose the MS-GARCH model with the Gibbs sampling method. The standard MS-GARCH(p,q) model is

$$\sigma_t^2 = \xi(R_t) + \sum_{i=1}^{q} \alpha_i(R_t)\varepsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j(R_t)\sigma_{t-j}^2$$

where z_t is a vector of an independently and identically distributed normal random variable with zero mean and unit variance. Indeed, GARCH(p,q) properties in MS-GARCH(p,q) satisfy ordinary GARCH(p,q), while R_t is a hidden Markov chain process with finite state space = $\{1, 2, 3, \ldots, d\}$, which is also called an unobservable/latent variable and its transition probabilities and stationary distribution are unknown. Note that the states are called regimes in econometric literature. For example, there are two finite state spaces, $s = \{1, 2\}$, $s \in$ and then $\{R_t = s\}$. y_t satisfies a GARCH function with parameter $\xi(s) > 0, \alpha_i(s) \geq 0,\ 0 \leq \alpha_i(s) + \beta_j(s) \leq 1,\ 1 \leq i \leq q, \beta_j(s) \geq 0,\ 1 \leq j \leq p$, which are again to ensure that the conditional variance almost surely strictly positive or a strictly stationary process. Suppose z_t and R_t are independent and that the hidden Markov chain is stationary, aperiodic, and irreducible with transition probability $p(m,n) := P(R_t = n|R_{t-1} = m)$ and the stationary distribution $\pi(s) := P(R_t = s), 1 \leq s \leq d, \forall\pi(s) > 0, \forall s \in E$. There is only one observed variable y_t to draw statistical inference. The estimated parameters from y_t are $\hat{\theta} := \{\hat{p}(m,n), \hat{\xi}(s), \hat{\alpha}_i(s), \hat{\beta}_j(s), m \neq n, 1 \leq m, n, s \leq d, 1 \leq i \leq q, 1 \leq j \leq p\}$.

The MS-GARCH model was again studied by Chuffart (2015), and the model selection criteria is the focal point of interest.. Besides this, the Logistic Smooth Transition (LST) with the GARCH model was studied by Hagerud (1996) and González-Rivera (1998). The LST-GARCH model is similar to the well-known GJR-GARCH model in Glosten et al. (1993) excluding the indicator discrete function, which is replaced by the logistic function.

To the best of the knowledge, the GARCH-type family with symmetric innovation has been extensively explored such as the Gaussian and the Student-*t* innovation. Currently, many researchers publish research papers related to asymmetric innovation, which corresponds to the non-Gaussian distribution characteristic of financial asset returns. Among others, Wichitaksorn and Choy (2015) demonstrate the GARCH and asymmetric Laplace (ALD) innovation. In Ardia, Bluteau, Boudt and Catania (2018), the MS-GARCH has several different innovations, including both symmetric and asymmetric innovations, and considered Gaussian, the Student-*t*, the skew normal and the skew t distribution. Additionally, Caporale and Zekokh (2019) also investigate the GARCH family, including the standard-GARCH, EGARCH, GJR-GARCH, TGARCH, and MS-GARCH with the normal mixture innovation. The GARCH-EVT-Copula model is studied by Sahamkhadam, Stephan and Östermark (2018), where the EVT is the extreme value theory, to cope with the crucial characteristics of the fat-tail and non-normally distributed and time-varying dependence among asset returns. For further discussion in the univariate model including mixture distribution in statistics see Chapter 3.

This leads to other advanced research areas in which copula time-varying dependence models are explored in relation to the (non-)normality of asset returns. The concepts and literature related to copula dependence models will be primarily focused on their application in quantitative risk management.

Lastly, it is of significance to provide a brief overview of another crucial member of the GARCH family, particularly in the continuous-time literature. This variant is known as the continuous-time generalised autoregressive conditionally heteroskedastic model (COGARCH), also referred to as the diffusion model, diffusion probabilistic model, or score-based generative model, as studied by Klüppelberg, Lindner and Maller (2004). Among these, other well-known diffusion models are, for example, the GARCH diffusion approximation, see the research conducted by Nelson (1990) (also in Drost and Werker (1996), and Duan (1997)). Although the GARCH model involves a single noise sequence, the diffusion limit is driven by two independent Brownian motions, denoted as $W_t^{(1)}$ and $W_t^{(2)}$ for $t \geq 0$. Consequently, the diffusion limit of the GARCH(1,1) model takes the form

$$dG_t = \sigma_t dW_t^{(1)},$$
$$d\sigma_t^2 = \zeta(\gamma - \sigma_t^2) + \rho\sigma_t^2 dW_t^{(2)},$$

where $t \geq 0$ and G_t represents the logarithm of the asset price (P_t), denoted as P_t in the GARCH process. Notably, the characteristics of the diffusion limit diverge from those of the GARCH process itself due to the independent evolution of σ_t in the first equation relative to the Brownian motion $W_t^{(1)}$.

This establishment the linkage to a dynamic pricing model that connects with other crucial research areas such as option pricing, hedging, market completeness, and more. Additionally, this would be a further extended to encompass two significant discussions related to the thesis: (1) portfolio optimisation in continuous time within a complete market (theory); and (2) the challenges involved in deriving frequentist asymptotics for GARCH models (Das, Goswami & Rana, 2018; Cheng, Gao & Phillips, 2018; Augustyniak & Badescu, 2021). This includes the exploration of topics such as the recurrence equation and contract mapping (Ganapathy & Wah, 1992; Karypis & Kumar, 1993).

1.3 Copula-based Model in Portfolio Strategy

This section introduces the copula-based dependence model for forecasting financial asset returns, and is split into two subsections. The first subsection introduces the basic copula model, i.e., the bivariate copula model (Subsection 1.3.1) and the second subsection introduces the copula model in general circumstances, i.e. the multivariate copula model (Subsection 1.3.2).

1.3.1 Bivariate Copula Model and Dependence Measure

The copula model is an important and well-known model, which is used to describe time-varying dependence among asset returns in the financial market because it can describe either linear or non-linear relationships among assets (Fan & Patton, 2014; Patton, 2012). The varieties of the copula family (e.g., the elliptical, the Archimedean and the vine copulae) are very flexible. By allowing for arbitrary marginal distribution, copula models can generate more accurate estimates and forecasts of the distribution of individual asset returns in a portfolio. For instance, the estimation of the lower and/or upper tail of the distribution can then be extended to improve the performance of portfolio risk measures. See details of the Sklar theorem (Sklar, 1959) and the concepts of copula in Joe (2015) and McNeil, Frey and Embrechts (2015, pp. 220-274).

Among others, Breymann, Dias and Embrechts (2003) analysed the time-varying comovement of high-frequency foreign exchange time series with the two main families of elliptical and Archimedean copulae. The elliptical copulae are the Gaussian copula and the Student-t copula. In the bivariate case, the Gaussian copula and the Student-t copula, respectively, are

$$C_{\Sigma}^{Ga}(u) = \Phi_{\Sigma}(\Phi^{-1}(u_1), \Phi^{-1}(u_2))$$

where Φ^{-1} is an inverse cumulative Gaussian distribution function or inverse cdf of the Gaussian, Φ_{Σ} is the cdf of the Gaussian where Σ is a 2 by 2 correlation matrix, $u_i \sim Unif(0,1)$ is a continuous standard uniform random variable for all $i = 1, 2$ and

$$C_{v,\Sigma}^{t}(u) = t_{v,\Sigma}(t_v^{-1}(u_1), t_v^{-1}(u_2))$$

where t^{-1} is an inverse cdf of the Student-*t*. $t_{v,\Sigma}$ is the cdf of the Student-*t* distribution where $v > 2$ degrees of freedom and Σ is a 2 by 2 correlation matrix. In this study, the Archimedean copulae are analysed, including the Gumbel, the Clayton and the

Frank. Furthermore, Table 1.1 demonstrates 13 bivariate Archimedean cdf copulae in one parameter and two parameters, $C(u,v) = \varphi^{-1}(\varphi(u) + \varphi(v))$. One-parameter Archimedean copulae are Clayton, Frank, Gumbel, Galambos and Joe/B5 and two-parameter Archimedean copulae are BB1, BB3, BB4, BB5, BB6, BB7, BB8, BB9 and BB10, which come from the relevant copula in the literature. It is worth mentioning that $C(u,v;\theta,\delta) = G(x,y;\theta,\delta)$ where $G(x,y;\theta,\delta)$ is a bivariate survival function and x,y are monotonically decreasing transforms. Therefore, the conditional cdf and copula pdf are, respectively,

$$\begin{aligned} C_{2|1}(v|u;\theta,\delta) &= \frac{\partial G}{\partial x}.\frac{\partial x}{\partial u} \\ c(u,v;\theta,\delta) &= \frac{\partial^2 G}{\partial x \partial y}.\frac{\partial x}{\partial u}.\frac{\partial y}{\partial v}. \end{aligned}$$

where $\varphi(.)$ is a strictly increasing and continuous function. A parameter can be calculated in the expression for the copula-based dependence measure, Kendall's τ in the case of the Archimedean was reduced to $\tau = 1 + 4\int_0^1 \frac{\varphi(u)}{\varphi\prime(u)}du$ or a sample estimate of τ, $\hat{\tau} = P\left[(x_i - x_j)(y_i - y_j) > 0\right] - P\left[(x_i - x_j)(y_i - y_j) < 0\right] = (c-d)/nC2$ where *c* is a number of concordatory pairs, *d* is a number of dis-concordatory pairs, $nC2$ is a total number of pairs and $nC2$ is $\frac{n!}{(n-2)!2!}$ (n choose 2). Alternatively, numerical approximation can be conducted via a Monte-Carlo simulation through two-variable integrals as stated in equation 1.1 for Kendall's τ and equation 1.2 for Spearman's ρ_s, respectively. For the closed form solution of the copula correlation coefficient, see Table 1.2. Figure 1.3 demonstrates an example of the Gaussian dependence measure. For further cdf copulae and their properties, see Joe (2015, Chapter 4).

$$\hat{\tau}(C) = 1 - 4\int_0^1 \int_0^1 C_{1|2}(u|v)C_{2|1}(v|u)dudv \tag{1.1}$$

$$\hat{\rho_s}(C) = 3 - 12\int_0^1 \int_0^1 uC_{2|1}(v|u)dudv \tag{1.2}$$

Table 1.1: Archimedean Parametric Copula Family

Name of Copula	**Bivariate Copula** $C(u,v), 0 \leqslant u, v \leqslant 1$	**Parameter** θ, δ
One-parameter		
Gumbel	$\exp[-((-\log(u))^{\theta} + (-\log(v))^{\theta})^{1/\theta}]$	$\theta \geqslant 1$
Clayton	$[max(u^{-\theta} + v^{-\theta} - 1, 0)]^{-1/\theta}$	$\theta \geqslant -1, \theta \neq 0$
Frank	$-\frac{1}{\theta} \ln\left(1 + \frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{e^{-\theta}-1}\right)$	$-\infty < \theta < \infty, \theta \neq 0$
Joe/B5	$1 - [(1-u)^{\theta} + (1-v)^{\theta} - (1-u)^{\theta}(1-v)^{\theta}]^{1/\theta}$	$\theta \geqslant 1$
Galambos	$uv \exp\{[(-\log u)^{-\theta} + (-\log v)^{-\theta}]^{-1/\theta}\}$	$0 \leqslant \theta < \infty$
Two-parameters		
BB1	$(1 + [(u^{-\theta} - 1)^{\delta} + (v^{-\theta} - 1)^{\delta}]^{1/\delta})^{-1/\theta}$	$\theta > 0, \delta \geqslant 1$
BB3	$exp\{-[\delta^{-1} log(e^{\delta \tilde{u}^{\theta}} + e^{\delta \tilde{v}^{\theta}} - 1)]^{1/\theta}\}, \tilde{u} = -log(u), \tilde{v} = -log(v)$	$\theta \geqslant 1, \delta > 0$
BB4	$(u^{-\theta} + v^{-\theta} - 1 - [(u^{-\theta} - 1)^{-\delta} + (v^{-\theta} - 1)^{-\delta}]^{-1/\delta})^{-1/\theta}$	$\theta \geqslant 0, \delta > 0$
BB5	$exp\{-[x^{\theta} + y^{\theta} - (x^{-\theta\delta} + y^{-\theta\delta})^{-1/\delta}]^{1/\theta}\}, x = -log(u), y = -log(v)$	$\theta \geqslant 1, \delta > 0$
BB6	$1 - (1 - exp\{-[(-log(1 - \bar{u}^{\theta}))^{\delta} + (-log(1 - \bar{v}^{\theta}))^{\delta})^{\delta}]^{1/\delta}\})^{1/\theta}, \bar{u} = 1 - u, \bar{v} = 1 - v$	$\theta \geqslant 1, \delta \geqslant 1$
BB7	$1 - (1 - [(1 - \bar{u}^{\theta})^{-\delta} + (1 - \bar{v}^{\theta})^{-\delta} - 1]^{-1/\delta})^{1/\theta}, \bar{u} = 1 - u, \bar{v} = 1 - v$	$\theta \geqslant 1, \delta > 0$
BB8	$\delta^{-1}(1 - \{1 - \eta^{-1}[1 - (1 - \delta u)^{\theta}][1 - (1 - \delta v)^{\theta}]\}^{1/\theta}), \eta = 1 - (1 - \delta)^{\theta}$	$\theta \geqslant 1, 0 < \delta \leqslant 1$
BB9	$exp\{-[(\delta^{-1} - log(u))^{\theta} + (\delta^{-1} - log(v))^{\theta} - \delta^{-\theta}]^{1/\theta} + \delta^{-1}\}$	$\theta \geqslant 1, \delta > 0$
BB10	$uv[1 - \delta(1 - u^{\theta})(1 - v^{\theta})]^{-1/\theta}$	$\theta > 0, 0 \leqslant \delta \leqslant 1$

Table 1.2: Closed Form Solution for Kendall's τ and Spearman's ρ_s in the Copula Family Using Linear Correlation Coefficient ρ

Copula	**Kendall's τ**	**Spearman's ρ_s**
Gaussian	$2\pi^{-1}\arcsin(\rho)$	$6\pi^{-1}\arcsin(\rho/2)$
Student-*t*	$2\pi^{-1}\arcsin(\rho)$	$6\pi^{-1}\arcsin(\hat{\rho}/2), \hat{\rho} = 2B((\rho+1)/2, a(\nu), a(\nu)) - 1$
Gumbel	$(\theta-1)/\theta$	*numerical approximation*
Clayton	$\theta/(\theta+2)$	*numerical approximation*
Frank	$1+4\theta^{-1}[D_1(\theta)-1]$	$1+12\theta^{-1}[D_2(\theta)-D_1(\theta)], D_k(x) = kx^{-k}\int_0^x t^k(e^t-1)^{-1}dt$
Joe/B5	$1-2a(2-\theta)^{-1}$	*numerical approximation*
BB1	$1+2/(\delta(\theta+2))$	*numerical approximation*

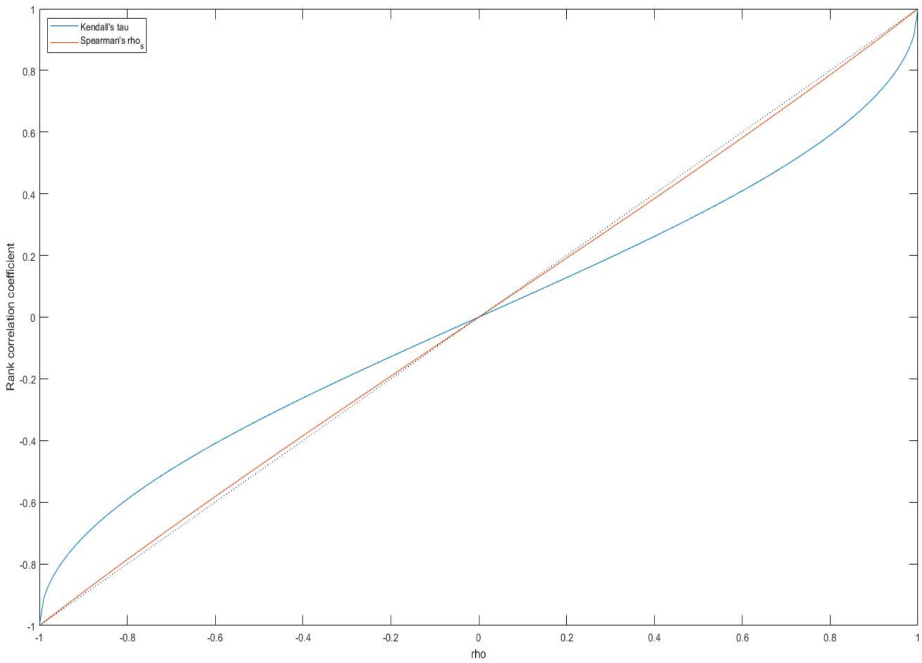
Note that $a = \psi(2) - \psi(2/\theta+1)$. B is the incomplete beta function, $a(\nu)$ is Spearman's ρ_s approximation for a Student-*t* copula with ν degrees of freedom. ψ is the digamma function.

Note that the parametric copula function is a well-known function in the application of portfolio strategy because it can determine the (non-)linear dependence structure of time series. When the investment portfolio has more than two assets, the bivariate dependence model may not be applicable. Therefore, the general multivariate copula has been initiated in the literature and leads to a discussion of the application of the multivariate copula function in an $n-$dimensional asset portfolio allocation.

1.3.2 Multivariate Copula Model

This subsection presents a literature review of the copula model as applied to portfolio management. Since Sklar (1959) proposed the copula theory in 1959, Sklar's Theorem has been broadly accepted to capture the dependence structure between financial assets and can combine non-Gaussian distributed characteristics such as asymmetry, fat-tails and a (non-)linear correlation to the model for asset forecasts. Moreover, the various parametric copula families (the elliptical and the Archimedean copulae) are very flexible. To be precise, it allows the use of any marginal distributions which could produce a better result for the forecasting of a multivariate distribution in an investment portfolio, i.e., the lower and/or upper tails of the distribution, and then it improves the performance of estimating portfolio risk, i.e., VaR and CVaR. The elliptical Gaussian and Student-*t*

Figure 1.3: Kendall's τ and Spearman's ρ_s against Linear Correlation Coefficient ρ of Bivariate Gaussian Copula



copulae in multivariate cases are, respectively,

$$C_{\Sigma}^{Ga}(u) = \Phi_{\Sigma}(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d))$$
$$C_{v,\Sigma}^{t}(u) = t_{v,\Sigma}(t_v^{-1}(u_1), \ldots, t_v^{-1}(u_d)).$$

In the literature, In the literature, there are numerous studies on bivariate copulae including, for example, Breymann et al. (2003) study modelling a bivariate deseasonalised dependence structure, where the copula functions are Gumbel, Student-*t*, Clayton, Frank and Gaussian copulae for different high-frequency data sets including 1, 2, 4, 8, and 15 hour(s) and 1 day to the log foreign exchange (FX) middle return. A two-stage semi-parametric procedure was applied to fit parametric copula functions and the best fitting model was selected by the Akaike information criterion (AIC). The results indicate that the Student-*t* copula outperforms the other copula functions and provides the best overall description of the FX return time series among all the candidates, as evidenced by the AIC and tests of goodness-of-fit. Furthermore, this study suggests that the Clayton and the survival Clayton copulae are, respectively, suitable for analysing the lower tails and the upper tails of the multivariate excesses of hourly returns.

The copula models were applied to various marginal models, for instance, the family of the ARMA-GARCH and the generalised autoregressive score (GAS). Among others, the GAS and a generalised autoregressive score with realised measures (GRAS) were studied by De Lira Salvatierra and Patton (2015), Fengler and Okhrin (2016) and Oh and Patton (2018). The copula procedure was proposed by Fengler and Okhrin (2016) as follows: firstly, the constant copula model was performed using a rolling window technique under a local change point detection algorithm. Secondly, three dynamic conditional copula models with marginal models were applied, i.e., the copula-GARCH-DCC model, the dynamic copula models, the GAS model and the GRAS model. Finally, two standard realised covariance models were applied. In total, there were 26 copula-based models employed with different marginal functions.

Besides, the additional marginal model of De Lira Salvatierra and Patton (2015) is in the GARCH family as the GJR-GARCH-DCC with a skewed-t distribution model, whereas, Oh and Patton (2018) also considered the AR-GJR-GARCH with skewed-t distribution, three copula functions, namely: 1) the Gaussian copula function as a base line model; 2) a rotated (or survival) Gumbel copula function, which represented a lower-tail dependence and an asymmetrical distribution; and 3) a standard Student-t copula function, which represented both tail dependence and a symmetrical distribution. By contrast, a factor copula was used in the study of Oh and Patton (2018), which considered the three different models for the Monte Carlo simulation including a dynamic equidependence model (G = 1), a dynamic block equidependence model (G = 10), and a dynamic heterogenous dependence model (G = N).

Prior to that, the factor copula was also studied by Creal and Tsay (2015) in order to solve two challenging problems: the first was the computation of the large cross-sectional dimension of the observed density functions and the second was the likelihood of a high-dimensional integral over the path of the latent state variable in a particle Gibbs sampler method applied to 200 financial assets, where the candidate copula functions with time-varying correlation matrices are the Gaussian, Student-t, grouped Student-t, and generalised hyperbolic copulae. Moreover, Creal and Tsay (2015) summarised the advantages of factor copula, which include tractability and flexibility over time, as well as having heavy tails in high dimensions. Furthermore, numerical results proved that the grouped Student-t copula with a time-varying factor function was the best fit to the data when compared with the other competing models among 200 unbalanced stocks and CDSs from US firms.

De Lira Salvatierra and Patton (2015) argued that the Student-t GRAS model is the best among the competing models, providing a robust result in a portfolio allocation problem. There was an empirical evidence that the advantage of high frequency data for a stochastic copula model of the distribution daily returns, including a realised measure, significantly improved the performance of the model. More precisely, The GRAS model performs well in capturing both higher and lower dependency values. In addition, given the out-of-sample test results, as well, The GRAS-copula model demonstrates superior performance over the constant copula model and GAS model in statistical tests for density forecast accuracy.

In the study of Fengler and Okhrin (2016), the results of an in-sample parameter estimation indicate that the Clayton copula and the rotated Gumbel copula are important for daily VaR portfolio analysis. Then, the out-of-sample backtesting risk measure analysis was performed, and the numerical results found that the new proposed realised copula model (RCop) was the most favourable among the various models compared, forecasting daily VaR at a 99% confidence level. In other words, the RCop model is better at capturing the 1% quantile of the profit and loss distribution of a realistic portfolio investment rather than the 5% and 10% quantiles of the profit and loss distribution. By contrast, the empirical results of Oh and Patton (2018) showed that the proposed model of dynamic joint distribution via a factor copula can be feasibly applied in 100 or more dimensions and is much better than previous research results given by statistical testing.

Additionally, Oh and Patton (2018) argued that systemic risk increased during the 2008 financial crisis and, interestingly, was still relatively high in the pre-crisis period.

In addition, the DCC-EVT marginal model was studied with the elliptical copulae for VaR risk forecasting for four asset portfolios in Berger (2013), where the EVT distribution is the generalised Pareto distribution (GPD) . However, the GPD was not the only innovation in the study of Berger (2013); in addition, the most two common distributions were determined, namely, the Gaussian and Student-*t* distributions. The empirical results confirm the sufficiency and precision of the proposed DCC-copula model compared to the competing static copula model. The results also highlight the crucial role of the marginal function's condition in VaR forecasting. More to the point, the copula distribution is preferable because it does not restrict the choice of marginal function, such as the Gaussian, Student-*t* and EVT margins where the paper emphasised that the best distribution function of the margin was the GPD. Again, for the VaR estimate and forecast, the DCC-copula-EVT model demonstrated superior statistical performance compared to all other competing models, particularly during periods of market turbulence.

In 2014, the Archimedean (Clayton, Gumbel and Frank) copula function was recommended among the choice of multivariate distributions used to assess portfolio performance in the study of Kakouris and Rustem (2014), as given by three optimisation models: 1) the Gaussian Portfolio (GP) (case-based optimisation problem, using Gaussian copula-CVaR model); 2) a more abstract portfolio, which is the Worst Case Portfolio (WCP) (a mixed copula-CVaR model); and 3) the Worst Case Markowitz Portfolio (WCMP) model. The study stated that CVaR risk was an alternative coherent risk measure and provided ample information on the tail of the loss distribution while VaR was not a coherent risk measure; hence, it may be obtained through many local extrema, which cause technical issues in the optimisation problem. The empirical results showed that the GP was more favourable when represented in a static portfolio. An equally weighted portfolio (EWP) was added to the performance analysis and revealed that the MD2, AR, TR, CVaR and volatility of the WCP were higher than the competing models while the WCMP's performance results were similar to the GP's performance in a dynamic portfolio. Overall, the WCP can outperform both the GP and the WCMP in out-of-sample forecasts, except for the volatility and the CVaR. Furthermore, a WCP using a mixed copula-CVaR proved to be beneficial over the 2008 crisis.

With regard to the GARCH family, there are many studies of the GARCH family in copulae in the literature. Among others, the hybrid of a heterogeneous autoregressive (HAR) model and a GARCH-DCC model was determined as the marginal model in symmetric copula models (the Student-t, Clayton, Frank and Gumbel copulae) and benchmark copula models (the independence copula and the multivariate Student-t distribution) (Oh & Patton, 2016). The study showed that there was a strong evidence for fat tails and all rejected the null hypothesis for zero pair-wise correlations. Based on the t-test comparison, it was found that the in-sample and out-of-sample models showed that the copula-based and HAR model performed with the best fit to the 104 assets. More precisely, the jointly symmetric Student-t-copula and HAR model performed as the best fitting model to the data, while the second best is the Clayton-copula-HAR model.

An approach of a component generalised autoregressive conditional heteroscedasticity type model (CGARCH) with copula (the ARMA(1,1)-CGARCH(1,1)-EVT-bivariate-Copula approach) was initiated to estimate the VaR and CVaR in quantitative risk management using data obtained at five-minute intervals for three pairs of stock indices, where the candidate of the bivariate-co-movement function as a joint model was the four copula functions including Student-t, Clayton, Gumbel and BB1 copulae (Karmakar & Paul, 2018). The GARCH-EVT-Copula framework has become the natural alternative for the VaR and CVaR risk studies because the flexibility of the model allowing asset returns has non-linear correlation and time-variability, skewness, and fat tails. The out-of-sample backtesting evidence, from three years of high frequency data, disclosed that, for the VaR analysis, the CGARCH-EVT-Copula model statistically outperformed the CGARCH-EVT model. Furthermore, the study emphasised the superiority of the results of the CGARCH-EVT-Copula type model over the competing models in the case of the high-frequency VaR and CVaR portfolio risk measure.

D'Amico and Petroni (2018) used the GARCH families were also used as a benchmark (GARCH (1,1), GARCH(2,1), GARCH(1,2)) for a synthetic autocorrelation function among six selected stocks on the Italian Stock Exchange ('BorsaItaliana'). They proposed a new stochastic-multivariate model called weighted-indexed semi-Markov chain processes (WISMC model) to copula model to discover Marginal VaR, VaR and CVaR for stock volatility forecasting. Given the mean percentage error (MPE) for forecasting intraday return volatility, the proposed WISMC-Copula model exhibits superior forecasting performance compared to all benchmark GARCH models. Furthermore, it was stated that, due to the Komogorov–Smirnov statistical test and the descriptive statistics, the empirical result confirmed that the distribution of real data and the proposed model can be considered as the same distribution. From the portfolio risk management point of view, the results of VaR estimation for real and synthetic data are almost identical under an equally weighted mean-VaR optimisation framework.

The study of F. Li and Kang (2018) also used the GARCH family, in which the specified marginal distribution of GARCH(1,1) incorporated the autoregressive conditional double Poisson ACDP(1,1) model. The proposed marginal model was applied to the copula families in the problem of an equally weighted five-stock portfolio, with the vine copula model selected in this case along with the massive copula candidates which are the Gaussian, Student-*t*, Clayton, Frank, Gumbel, Joe, BB1, BB8, Survival Clayton and Survival BB8 copulae, using the maximum likelihood method for each pair-structure parameter estimate and selection. Prior to that, the vine copula and GARCH(1,1)-Student-*t* model were also studied by So and Yeung (2014), who considered only three bivariate copulae for the vine copula, the Student-*t*, Clayton, and Gumbel copulae. In F. Li and Kang (2018), the empirical results revealed that Gaussian, Clayton and Frank copula were the most frequently selected functions in vine copula-based models. For out-of-sample VaR comparisons, the Joe-Clayton copula model can capture and was the most suitable for reflecting the VaR deviation when the DCC-GARCH model was the marginal model for an equally weighted portfolio of S&P 100 and S&P 600 returns.

The DCC-GARCH-Joe-Clayton-Copula model offered several advantages, including the ability to simulate and replicate data studies, thereby improving our understanding of bivariate co-movements. Additionally, the model improved risk management by enhancing the accuracy of estimating and predicting the density function, which in turn allowed for a more precise evaluation of portfolio VaR risk. Empirically, the model of the vine copula with the GARCH-Student-*t* of So and Yeung (2014) showed that the risk level of stocks increased when influenced by the increase in the correlation of many pairs of stocks.

Again, the vine copula family was selected in Bekiros, Hernandez, Hammoudeh and Nguyen (2015) who studied optimal asset allocation in multiple risk measures and specific market conditions on two mining stock portfolios on the Australian Securities Exchange (ASX). Bekiros et al. (2015) explored three vine copulae, namely the regular vine (R-vine) and the two special cases of the vine copula, the canonical vine (or C-vine) and the drawable vine (or D-vine) through the risk measures of the variance, the mean absolute deviation (MAD), the minimising regression (Minimax), the conditional Value-at-Risk (CVaR) and the conditional Drawdown-at-Risk (CDaR) measures. There were 20 daily dependent risk assets in each portfolio. Two investment portfolios were examined: one was considered low-risk (gold stocks), while the other was considered high-risk (iron ore/nickel stocks). The analysis covered three financial periods and considered the following scenarios: the pre-GFC (Global Financial Crisis) period, the GFC period, and the post-GFC period. To model the distribution, the first step was an ARMA(1,1)-GARCH(1,1) process with Student-*t* distribution employed to log return filtration. The second step used in the pair vine copula function employed the standardised residuals from the first step transformed via a probability integral transform to become the copula data. The empirical results demonstrated that the risk-return analysis of the gold portfolio with R-vine copula and the iron ore-nickel portfolio with C-vine could outperform the traditional mean-variance optimisation, thanks to its ability to capture the complexity of dependence structures and confirmed from the goodness-of-fit testing based on the Cramer-von Mises (CvM) and Kolmogorov-Smirnov statistical test. Besides this, optimal asset weight allocation in each portfolio was provided by the multiple risk measure where the average of weight given by risk measures showed that some stocks could be recommended as the good candidates for an investment choice for mining investors.

The copula model was also used in the portfolio optimisation problem where the proposed copula families were again both the elliptical and Archimedean copulae in the non-dominated sorting genetic algorithm II (NSGAII) algorithm and the strength Pareto evolutionary 2 (SPEA2) algorithm employed in a portfolio of up to 200 S&P stocks in Babaei, Sepehri and Babaei (2015). In this study, a dynamic stochastic financial data model with α-stable distribution was a marginal model. The results showed that the Student-*t* copula gave a superior result over the other copulae. The Gaussian and Student-*t* copulae were more likely best fitted to the data than the Clayton, Frank, and Gumbel copulae, according to log-likelihood, BIC and AIC values. More precisely, according to goodness-of-fit test, in all cases, the Student-*t* copula can outperform the Gaussian copula at a 1% significance level. Furthermore, this study also found that Gaussian marginal distribution and historical distribution led to an extreme risk of underestimation.

The DCC model was studied again as a marginal function of the Student-*t*-copula model by Al Janabi, Arreola Hernandez, Berger and Nguyen (2017) in non-linear liquidity-adjusted VaR optimisation algorithms. The proposed model was investigated through four illiquid market scenarios and an interesting benchmark mean-VaR-CCC-GARCH model. The classical Markowitz approach was applied to prove the robustness of the proposed model. Portfolio allocation was subjected to four realistic scenarios as follows: scenario 1 the absence of short selling without (and with) a restrictive portfolio proportion in developed and emerging markets in a total of 12 indices; scenario 2 the optimum with budget constraint; scenario 3 the impact of short selling on the portfolio optimisation solution; and scenario 4 model flexibility by defining the different degree of liquidity in each market approach. The research discovered that the DCC-Student-*t*-copula model can outperform the benchmark model in terms of liquidity risk and a risk-adjusted return. An out-of-sample experiment showed the proposed model had the best portfolio performance (compared with competing models) over the mean-VaR efficient, risk-parity and equally weighted portfolio.

The GARCH-EVT-copula and ARMA-GARCH-EVT-copula models were again studied by Sahamkhadam et al. (2018) who claimed that they were the two best models for forecasting expected returns and standard deviation in a portfolio when the optimisation problems were Min-CVaR (minimising the conditional Value-at-Risk), GMV (minimising the variance) and CET (maximising the Sharpe ratio). The proposed copulae were the elliptical copulae (Student-*t* and Gaussian) and Archimedean copulae (Clayton, Frank, and Gumbel). The data were the daily adjusted logarithmic returns of 10 stock indices, with a rolling window of 1,260 days out of 5,218 days during 1996-2016 employed for the out-of-sample parameter estimation. The result of the out-of-sample test indicated that the best model was the GARCH-EVT-copula model and the optimisation problem was the CET portfolio. The study also found that the portfolio risk of the elliptical copula function was a that of preferable dependence function because it perceived a lower portfolio CVaR risk than the Archimedean copula function.

The GARCH family and copula were employed again in a study where the GARCH family was an asymmetric GJR-GARCH with ARMA model and the copula model was the R-vine copula; these models were applied to the optimisation problem of two variants of stable tail-adjusted return ratios (STARR) (Goel, Sharma & Mehra, 2019). The copula candidates for the regular vine were the Gumbel, Frank, Clayton, Gaussian and Joe copulae. The competing model was the standard Gaussian copula. The Markowitz model and marginal GARCH model were considered among the choice of the indicators, which were excess mean return (EMR) from the 1/m naive portfolio, the Sortino ratio, the Rachev ratio, the VaR ratio, and the Treynor ratio. The worst-case schemes of the robust STARR ratio optimisation approach, with the R-vine copula, resulted in a superior performance in capturing the time-varying structure of the market compared to the standard Gaussian copula

The copula function was studied in another marginal model, the conditional heteroscedastic autoregressive nonlinear model, also known as the multivariate non-parametric conditional heteroscedastic autoregressive nonlinear (CHARN) (Neumeyer, Omelka & Hudecová, 2019). The copula functions consisted of Clayton, Frank, Gumbel, Gaussian, and Student-*t* copulae under the Kendall's tau inversion (IK) and the maximum pseudo-likelihood (MPL) procedure. The empirical results demonstrated that the Student-*t* copula was outstanding among the competing copulae. On the other hand, simulation results across a range of correlations indicated that the IK method was the preferred approach when data dependence was particularly strong while the MPL method was a more suitable method for low tau. Furthermore, the standard deviation of the MPL was almost always a slightly smaller value than for the IK method. The bias of the IK method can be significantly less than the MPL method. The MPL method also works well with the elliptical copula for all co-movements.

Now, it is worth to mentioning why this study proposes GARCH-type model rather than a stochastic volatility (SV) model in multivariate copula analysis (Chapter 4 and Chapter 5). Even if, in the literature in the past two decades, the SV model has demonstrated a tendency to outperform the GARCH model. It is wildly accepted that SV classes have been considered to offer a higher goodness-of-fit and flexibility than GARCH classes where the main difference lies in the fact that SV classes introduce a stochastic latent variable with unexpected noise (see, among others, Jacquier, Polson and Rossi (1994); Kim, Shephard and Chib (1998); Carnero, Peña and Ruiz (2004); X. Wang, Zhang and Zhao (2017); Agbeyegbe (2022)). However, in the multivariate copula case, SV framework may provide incorrect results in describing the co-movements between financial returns when dealing with non-linearity and asymmetric tail-dependence, especially, the case of the distribution has higher lower tail-dependence than upper tail-dependence(Hafner & Manner, 2012; Joe, Li & Nikoloulopoulos, 2010).

In conclusion for the analysis of the copula literature in the portfolio optimisation problem, the use of daily data is more widely studied than the use of intraday data. Almost all of the published papers use daily time series data as representative of lower-frequency data in their proposed model, and in particular, the more variety of the proposed models, and apply their proposed model to a various markets around the world, again using different techniques for model parameter estimation methods. According to the literature, the copula function can be combined with any marginal functions and can improve the performance of estimates and forecasts of finance asset returns. Consequently, copula can improve the optimal asset weight in portfolio strategy. Further discussion of copula concepts, especially the advanced vine copula, and further review of the literature, see Chapter 4 and Chapter 5. In the next section, another important research area in literature of the model parameter estimation method is introduced with a more detailed discussion of the class of efficient Bayesian MCMC algorithms. Furthermore, Chapter 4 discusses another current popular model, the parameter estimation method, which is a class of Bayesian machine learning algorithms. Note that both classes of estimation methods are advanced mathematical algorithms that used in this thesis.

1.4 Model Parameter Estimation

This section introduces the model parameter estimation method in this thesis. There are three main parts in this section including: 1) introduction to the traditional estimation method which it is the maximum likelihood method (MLE); 2) introduction to the Bayesian MCMC method with some literature; and 3) demonstration of the Bayesian MCMC and MLE estimation method of performance among the various models in the literature. Further intensive discussion and more literature are provided in Chapters 3 and 4.

The model parameter estimation method is another vital key to obtaining robust and superior forecasting performance. To the best of knowledge, the MLE and the Bayesian

MCMC are broadly accepted to evaluate model parameters in risk management. Besides this, another parameter estimation method is, for instance, the method of moments in Oh and Patton (2018). See the survey, among others, in Chang, McAleer and Wong (2018); Kolm, Tütüncü and Fabozzi (2014); Pareek and Thakkar (2015); and Patton (2012).

While the Maximum Likelihood Estimation (MLE) is a standard parameter estimation method, the Bayesian Markov Chain Monte Carlo (MCMC) approach provides higher accuracy in calibrating model parameters. However, the Bayesian MCMC comparatively requires further mathematical complexity and more computational time. The well-known Bayesian MCMC algorithms are the Gibbs sampling algorithm, the Metropolis-Hasting algorithm, and the Griddy-Gibbs algorithm (Billio et al., 2016). The essential part of estimation mechanisms for both of MLE and Bayesian MCMC is to formulate the likelihood function. Mathematically, the likelihood function is the joint density of random variable,

$$L(\theta) = \prod_{t=1}^{T} f(y_t|\theta). \tag{1.3}$$

In MLE, the parameter estimate $\hat{\theta}$ is obtained by maximizing the likelihood function and setting its first-order derivative with respect to θ to zero. To further verify the estimator value and maximise the likelihood function is by calculating the negative value of the second order derivatives. Also, the *fmincon* function in MATLAB can be used to solve this problem. Using Bayes' Theorem, the posterior density is given by

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)} \propto p(\theta)L(\theta) \tag{1.4}$$

where $p(\theta|y)$ is the posterior density. $p(y|\theta)$ is the likelihood function $L(\theta)$, $p(\theta)$ is the prior density, and $p(y)$ is the marginal density. See more details on pages 11-12 in Koop, Poirier and Tobias (2007). Note that the example of a random walk chain Metropolis-Hasting (rw-MH) algorithm to the regression model with ARCH(p) and normal errors can be found in Koop et al. (2007), exercise 17.5. For further discussion on the Bayesian MCMC algorithm, see Chapters 3 and 4. The further section discusses methods for parameter estimation in the past decade, with a particular focus on the MLE and Bayesian MCMC methods.

1.4.1 Maximum Likelihood

Among others, the use of the MLE is still a popular method in parameter estimation. Xie (2009) emphasised the reason why the MLE was still used and widely accepted by researchers given by the study of consistency of maximum likelihood estimator through the MS-GARCH model. Sahamkhadam et al. (2018) studied portfolio optimisation problems and used the MLE estimator for the GARCH-EVT-copula model and the ARMA-GARCH-EVT-copula model. Fengler and Okhrin (2016) used the MLE as a benchmark estimator in the study of managing risk with a realised copula parameter. Weiß and Supper (2013) use the MLE to solve parameters on the autoregressive conditional double Poisson model, and to the problem of an equally-weighted portfolio in a multivariate dependence model for forecasting liquidity-adjusted intraday VaR with vine copulae. Goel et al. (2019) also used the MLE to obtain optimal copula parameters in the proposed robust portfolio optimisation models. Jose Rodriguez and Ruiz (2012) used the MLE in the study of cryptocurrencies volatility via the GARCH-type model in order to demonstrate the superior model of volatility.

Moreover, the new MLE solver has been invented to answer complicated and specific problems in the financial market, since high dimension portfolio problem cannot be solved by the ordinary MLE. Among others, Oh and Patton (2016) used a maximum composite likelihood estimation method to solve the jointly symmetric copulae model. Segnon and Trede (2018) used two steps of MLE called the inference function of margins (IFM) method to the copula-MSM model and the GARCH-copula model. In the beginning step, the parameters of the univariate marginal functions were estimated. Given the estimate for the margins, the copula parameters were estimated in the second step, see details of the IFM method in Al Janabi et al. (2017). Bernardi and Catania

(2018) also used the IFM method for the study of portfolio optimisation under the first step, which was by maximum likelihood of the univariate margins GAS functions.

In the second step, they used the expectation-maximisation (EM) algorithm to deal with the latent variable of the Markov switching model. Babaei et al. (2015) and Berger (2013) also used the IFM method to obtain the parameters of the marginal and the copula functions especially in multi-objective portfolio optimisation problems and the VaR forecasting. The demonstration of the recent MLE approaches is in equation 1.3. The parameters' estimator $\hat{\theta}$ can be obtained via maximising the likelihood function by the first order derivatives with respect to θ to zero.

$$\frac{\partial L(\hat{\theta})}{\partial \hat{\theta}} = 0$$

To verify the estimators' value is a maximum of the likelihood function by calculating the negative value of the second order derivatives.

$$\frac{\partial^2 L(\hat{\theta})}{\partial \hat{\theta}^2} < 0$$

MATLAB function either *fmincon* or *fminsearch* can be used to solve this problem.

1.4.2 Bayesian MCMC

The Bayesian MCMC provides a better model estimator through Bayes' Theorem, (equation 1.4), which is a conjugate prior (belief) density and likelihood function. However, it takes more computational time than the MLE. Among others, Creal and Tsay (2015) used the particle Gibbs sampling algorithm to handle the high-dimensional daily returns of credit default swaps and equities dynamics stochastic copula models. In their study, Wichitaksorn and Choy (2015) employed the Griddy Gibbs (GG) sampling algorithm and parallel computing techniques to solve GARCH models incorporating the asymmetric Laplace function. The aim of parallel computing was to accelerate the computational time of the Bayesian MCMC approach by enabling independent calculations to be performed concurrently, known as the Parallelisation-Bayesian approach.

The Gibbs Sampling algorithm is

1. Take some initial value $\hat{\theta}_k^{(0)}, k = 1, 2, \ldots, K$
2. Repeat For $t = 1, 2, \ldots$ and
   For $k = 1, 2, \ldots, K$ generate $\hat{\theta}_k^{(t)}$ from
   $P(\hat{\theta}_k^{(t)} | \hat{\theta}_1^{(t)}, \ldots, \hat{\theta}_{k-1}^{(t)}, \hat{\theta}_{k+1}^{(t-1)}, \ldots, \hat{\theta}_K^{(t-1)})$ where P is the full posterior cumulative density function.
3. Repeat step 2 until $\hat{\theta}_1^{(t)}$, $\hat{\theta}_2^{(t)}, \ldots,$ $\hat{\theta}_K^{(t)}$ does not change.

Furthermore, for the Bayesian MCMC algorithm see the Griddy Gibbs (GG) sampling algorithm in the univariate model in Chapter 3. Additionally, Billio et al. (2016) applied the forward filtering backward sampling (FFBS) to the Bayesian MCMC algorithm including the multiple-trial Metropolis FFBS approach and the multiple-trial metropolised independent sampler FFBS approach to solve the MS-GARCH model. Ardia et al. (2018) evaluated parameters of the MS-GARCH model by the MLE and the Bayesian MCMC, of which the Bayesian MCMC was the adaptive random-walk Metropolis sampler algorithm in the study of the VaR and the CVaR forecasting. F. Li and Kang (2018) used the Metropolis-Hastings algorithm to solve the copula-GARCH model and the copula-stochastic-volatility model. Wichitaksorn, Gerlach and Choy (2019) used the Metropolis-within-Gibbs algorithm to solve the problem of the elliptical-copula regression models with scale mixture of the Gaussian innovation. For a more detailed description of the rw-MH sampling estimation method in the Bayesian MCMC method, please refer to Chapter 4, and the variational Bayes estimation method in Bayesian machine learning and further literature is also in Chapter 4.

1.4.3 Performance of Estimation Method

This section presents preliminary simulation results of both Bayesian MCMC and MLE estimation methods; used in the same well-known models. The models include linear regression, GARCH(1,1)-Gaussian-Innovation, GARCH(1,1)-Asymmetric-Laplace-Innovation and GARCH(1,1)-Generalised-Pareto-Innovation.

a) Linear Regression Model

According to equation 1.4, the Gibbs sampling algorithm for the linear regression model, $Y = X\beta+\epsilon$, requires the posterior density function, where Y a vector of dependent variable is X is a matrix of regressor, β is a matrix of estimator and ϵ is a vector of error term or noise. A full posterior density function is a likelihood function that is conjugate to a prior distribution.

$$p(\sigma^2, \beta) \ \propto N(X\beta, \sigma^2) IG(a, b) N(\mu, \tau^2)$$

where σ and τ is standard deviation of Gaussian distribution, μ is mean of Gaussian distribution and a is shape parameter and b is scale parameter of inverse gamma (IG) distribution. The first step draws σ^2 from the posterior distribution $\sigma^2|\beta, Y \sim N(X\beta, \sigma^2)IG(a, b)$ where $Y = X\beta, N(X\beta, \sigma^2)$ and is the likelihood function and denotes that $IG(a, b)$ is a belief/prior distribution. This can lead to a simplification of the posterior probability density function of σ^2 as follows:

$$\sigma^2|Y, \beta \ \backsim \ IG(a_n, b_n), \ a_n = \frac{n}{2} + a, b_n = (\frac{1}{2}(Y - X\beta)'(Y - X\beta) + b^{-1})^{-1}$$

The second step draws β given $\beta|\sigma^2, Y \sim N(X\beta, \sigma^2)N(\mu, \tau^2)$. After performing the necessary algebraic manipulations, the equation can be transformed into the following formula.

$$\beta|\sigma^2, Y \sim N(m, s), m = (X'X)^{-1}X'Y, s = \sigma^2(X'X)^{-1}.$$

Table 1.3 presents the estimated parameters obtained using the Gibbs sampling algorithm and the MLE method based on the simulated data, where the true parameters are $\beta_0 = 1.1, \beta_1 = 0.5, \beta_2 = 0.3, \sigma = 2, n = 500$, repeating number for Gibbs sampling = 1,000 times and discard the first 500 value. The table reveals that both methods for estimating model parameters exhibit a similar performance in terms of the first and second moments.

Table 1.3: Estimation Result of Linear Regression Model with Gaussian Innovation

True Parameter		**Estimator**	
		MLE	Gibbs Sampling
β_0	1.100	1.100	1.103
Std.		0.106	0.112
β_1	0.500	0.445	0.444
Std.		0.084	0.084
β_2	0.300	0.366	0.365
Std.		0.129	0.162
σ	2.000	1.890	1.891
Std.		0.215	0.233

Note that Std. = standard deviation.

b) GARCH(1,1)-Gaussian-Innovation Model Mathematically, the likelihood function of GARCH(1,1)-Gaussian-Innovation Model can be formulated in order to solve the model parameters by maximising the function. Hence, the (log-)likelihood function of the conditional Gaussian density of the errors of the GARCH(1,1) is

$$\begin{aligned}\Lambda(\theta) &= \sum_{t=2}^{T} ln(f_{y_t|y_{t-1}}) \\ &= -\frac{(T-1)}{2} ln(2\pi) - \frac{1}{2}\sum_{t=2}^{T} ln(\xi + \alpha\epsilon_{t-1}^2 + \omega\sigma_{t-1}^2) - \frac{1}{2}\sum_{t=2}^{T} \frac{\epsilon_t^2}{\xi + \alpha\epsilon_{t-1}^2 + \omega\sigma_{t-1}^2}.\end{aligned}$$

The estimated parameters $\hat{\theta} := \{\hat{\xi}, \hat{\alpha}, \hat{\omega}, \hat{\mu_y}\}$ can be obtained through either the MLE or the Bayesian MCMC method.

The GG set-up has 200 grid points and the number of draws is 3,000, with the first 30% of draws used as burn-ins. The prior density function for $\xi \sim Unif(0,1), \alpha \sim Unif(0,1), \omega \sim Unif(0,1), \mu_y \sim Unif(-1,1)$. Table 1.4 demonstrates the result of

MLE, the rw-MH algorithm and the GG sampling algorithm according to data simulation, where the true parameters are $\xi = 0.01$, $\alpha = 0.12$, $\omega = 0.85$, $\mu_y = 0.05$ and $T = 2000$. The rw-MH set-up has 1,000 burn-ins and is replicated 100,000 times.

Table 1.4: Estimation Result of GARCH(1,1) with Gaussian Innovation

True Parameter		**Estimator**		
		MLE	**rw-MH**	**GG**
ξ	0.01	0.008	0.013	0.010
Std.		0.002	0.015	0.003
α	0.12	0.111	0.117	0.123
Std.		0.018	0.024	0.019
ω	0.85	0.863	0.766	0.843
Std.		0.015	0.260	0.021
μ_y	0.05	0.056	0.063	0.056
Std.		0.005	0.037	0.010

Note that Std. is standard deviation.

c) GARCH(1,1)-Asymmetric-Laplace-Innovation Model

An asymmetric distribution is introduced to the GARCH model in the form of GARCH(1,1) with asymmetric Laplace innovation (ALD). The log of likelihood function of the conditional density of the errors of the GARCH(1,1)-ALD is

$$\begin{aligned}\Lambda(\theta) &= ln[\prod_{t=2}^{T}\frac{1}{\sigma_t}f(z_t)] = \sum_{t=2}^{T}\ln(\frac{1}{\sigma_t}f\left(z_t\right)) \\ &= \frac{(T-1)}{2}\ln\left(1-2\tau+2\tau^2\right) - \sum_{t=2}^{T} ln\ \sigma_t + \sum_{t=2}^{T}\frac{y_t\left(1-2\tau+2\tau^2\right)^{\frac{1}{2}}}{\sigma_t\left(\tau - I\left(y_t \geq \mu_y\right)\right)},\end{aligned}$$

where $\sigma_t = \sqrt{\xi + \alpha y_{t-1}^2 + \omega\sigma_{t-1}^2}$. The estimated parameters are $\hat{\theta} := \{\hat{\mu_y}, \hat{\xi}, \hat{\alpha}, \hat{\omega}, \hat{\tau}, \hat{\delta}\}$ or $\hat{\theta} := \{\hat{\mu_y}, \hat{\xi}, \hat{\alpha}, \hat{\omega}, \hat{\tau}\}$ since a closed form solution of δ is known. The table 1.5 presents the result of MLE, the rw-MH algorithm and the GG sampling algorithm. The true parameters for the simulation are $\xi = 0.01$, $\alpha = 0.85$, $\omega = 0.12$, $\mu_y = 0.05$, $\tau = 0.3$ and $T = 3000$.

The rw-MH set-up is run with 1,000 burn-ins and replicated 50,000 times. The GG setup is configured with 200 grid points, 3,000 draws, and the first 30% of draws are used as burn-ins. The prior density function for $\xi \sim Unif\,(0,1)\,, \alpha \sim Unif\,(0,1)\,, \omega \sim Unif\,(0,1)\,, \mu_y \sim Unif\,(-1,1)\,, \tau \sim Unif\,(0,1)$.

Table 1.5: Estimation Result of GARCH(1,1) with ALD Innovation

True Parameter		**Estimator**		
		MLE	**rw-MH**	**GG**
μ_y	0.05	0.054	0.077	0.054
Std.		0.000	0.030	0.003
ξ	0.010	0.009	0.010	0.010
Std.		0.001	0.002	0.002
α	0.850	0.792	0.780	0.776
Std.		0.050	0.060	0.060
ω	0.120	0.137	0.134	0.117
Std.		0.020	0.030	0.030
τ	0.300	0.315	0.371	0.315
Std.		0.006	0.070	0.010

Note that Std. is standard deviation.

d) GARCH(1,1)-Generalised-Pareto-Innovation Model

The Generalised Pareto Distribution (GPD) is a crucial distribution in Extreme Value Theory (EVT) and provides a better fit for the tail distribution of financial data compared to a Gaussian distribution. Other distributions in the EVT family include the Gumbel (Type I extreme value), Fréchet (Type II extreme value), and Weibull (Type III extreme value) distributions. For further discussion on EVT, please refer to Section 3.2 in Chapter 3. The log of likelihood function of the conditional density of the errors of the GARCH(1,1)-GPD is

$$\Lambda(\theta) = -\sum_{t=2}^{T} \ln \sigma_t - (T-1)\ln\left((1-\xi)(1-2\xi)^{\frac{1}{2}}\right) + \left(-\frac{1}{\xi} - 1\right)\sum_{t=2}^{T} \ln\left(1 + \frac{\xi y_t}{\sigma_t(1-\xi)(1-2\xi)^{\frac{1}{2}}}\right),$$

where $\sigma_t = \sqrt{\omega + \alpha y_{t-1}^2 + \gamma\sigma_{t-1}^2},\ \beta > 0,\ \xi < 1/2$.

The estimated parameters can be denoted as $\hat{\theta} := \{\hat{\omega}, \hat{\alpha}, \hat{\delta}, \hat{\xi}, \hat{\beta}\}$ or $\hat{\theta} := \{\hat{\omega}, \hat{\alpha}, \hat{\delta}, \hat{\xi}\}$ f the closed form of $\hat{\beta}$ is known in the study. Note that models such as ARCH(p,q), GARCH(p,q), or the ARMA(r,m)-GARCH(p,q) can be easily generalised and extended

Table 1.6: Estimation Result of GARCH(1,1) with GPD Innovation

True Parameter		Estimator		
		MLE	rw-MH	GG
ω	0.010	0.010	0.017	0.010
Std.		0.002	0.020	0.003
α	0.850	0.809	0.658	0.809
Std.		0.060	0.200	0.200
γ	0.120	0.124	0.122	0.124
Std.		0.0200	0.030	0.040
ξ	0.250	0.272	0.283	0.272
Std.		0.020	0.030	0.030
β	0.530	0.523	0.559	0.523
Std.		0.020	0.200	0.060

Note that Std. is standard deviation.

to other models, such as the GJR-GARCH model. Table 1.6 shows the result of MLE, the rw-MH algorithm and the GG sampling algorithm according to data simulation, where the true parameters are $\xi = 0.25, \alpha = 0.85, \omega = 0.01, \gamma = 0.12, \beta = 0.53, T = 3000$.

The set-up conditions for the rw-MH algorithm include a burn-in period of 1,000 iterations, and a total of 50,000 replications. The set-up conditions for the GG algorithm include 200 grid points, 3,000 draws, and a burn-in period consisting of the first 30% of draws. The prior density function for $\xi \sim Unif(0, 0.5), \alpha(0,1), \omega\,(0,1)\,, \gamma \sim Unif\,(0,1)\,, \beta \sim Unif\,(0,1)$.

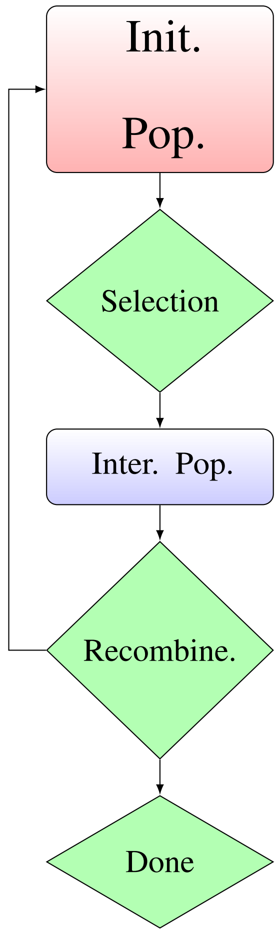
1.5 Machine Learning Portfolio Optimisation

This section presents the optimisation algorithm in machine learning used in this thesis where it can be useful for solving optimisation problems because they can quickly identify the best solution from a vast number of possibilities. This is especially important in complex optimisation problems where the traditional optimisation algorithms can be computationally expensive and time-consuming. Machine learning methods can learn patterns in the data and use this information to guide the optimisation process, leading to faster and more efficient solutions. Additionally, machine learning can be used to develop metaheuristic optimisation algorithms that can adapt to changing environments and handle noisy and incomplete data. Overall, machine learning methods can be a powerful tool for solving optimisation problems in a wide range of applications. In particular, the genetic algorithm (GA) and clonal selection algorithm (CSA) are presented in this section. For the survey of machine learning algorithm see Kalayci, Ertenlice and Akbay (2019, section 6), and for the survey of class of artificial immune systems algorithm, including CSA, see Timmis, Andrews, Owens and Clark (2008).

1.5.1 Genetic Algorithm

The genetic algorithms (GAs) are a massive class of algorithms in the stochastic evolutionary method where the search technique is inspired by the law of natural evolution of Charles Darwin (Holland, 1995). In general, a genetic algorithm consists of five main components:

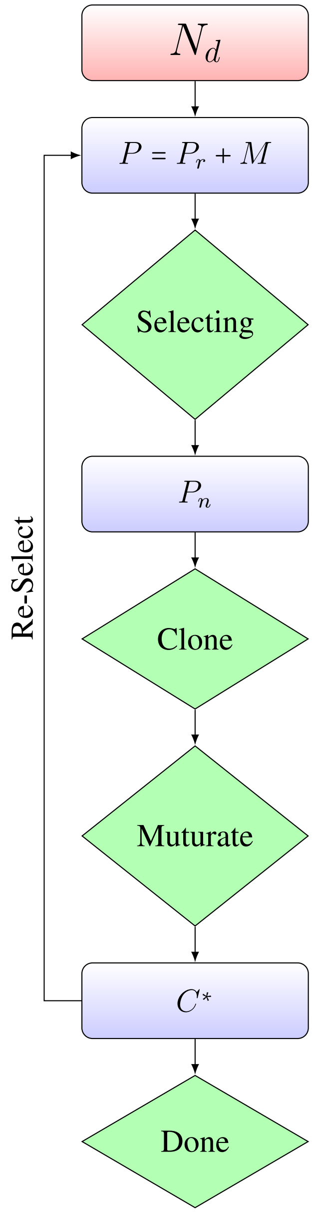
- An initial candidate set of genetic representation for potential solutions;
- A generation of an initial population for potential solution;
- An evaluation through a particular (fitness) function;
- Genetic processes that produce an offspring composition; and
- Valuation of the various parameters used by the algorithm: genetic operator, population sizes and so on.



The flowchart depicts the standard GA algorithm, where Init. Pop. is initial population, Inter. Pop. is intermediate population and Recombine. is recombination.

1.5.2 Clonal Selection Algorithm

The clonal selection algorithm (CSA) is one of the popular classes in an artificial immune systems algorithm (AIS) (De Castro & Von Zuben, 2002). The CSA uses shape-space formalism and immunology vocabulary, which describe Ag-Ab interaction and cellular evolution. The CSA performs searching methodology through the mechanisms of somatic mutation and receptor editing, by re-balancing the exploitation of the potential solution with the exploration of the search regime. The CSA flowchart is



The flowchart depicts the general idea of the CSA algorithm. The CSA algorithm can be summarised as follows:

(1) The initial population, $P = P_r + M$, is a set of potential solutions. P_r is the remaining population and M is an additional subset of memory cells.

(2) The potential solution (P) is then determined based on an affinity measure as the P_n best individual candidate, where $P_n \in P$.

(3) The P_n best potential solutions are cloned where the clone size is C, where C is

an increasing function of the affinity measure of the antigen.

(4) The C population in step (3) is submitted to a hypermutation scheme where the hypermutation is proportional to the affinity measure of the antibody and obtains a maturated antibody population C^*.

(5) Re-selection of C^* is processed to improve the individuals and replace the population P.

(6) Maintaining diversity by replacing d low affinity antibodies of the population.

Note that the mathematical scheme of the CSA and GA is not different. For further discussion of the mathematical scheme of the CSA and GA to portfolio management, algorithm implementation in optimisation problems and further literature reviews, see Chapter 5.

1.6 Parallel Computing Technique

This section provides a brief explanation of parallel computing techniques and their relevance, specifically in the context of Bayesian inference. Additionally, the literature related to parallel computing techniques in Bayesian inference will be discussed, where it is particularly important in Bayesian inference, as Bayesian MCMC methods require significant computational time to estimate the model parameters. This issue can be mitigated due to advanced technology i.e., a supercomputer among the parallel computing technique. This technique significantly reduces computational time on solving either the complicated mathematical model or the optimisation problem, especially the problem that belong to the tick-by-tick and high-dimensional time series data. For example, see Benczúr (2011); Fujiwara et al. (2018); Lam, Hu, Zhang and Ni (2019); Lopatka and Czyzewski (2014); Prudencio and Cheung (2012); and Sanchez-Vazquez, Avila-Costa and Cervantes-Pérez (2014).

The supercomputer is a computer with a high level of performance and can process a massive amount of data at the same time. The computing performance of the supercomputer is measured in flops (floating-point operations per second). Nowadays, in the data driven era, the supercomputer is a significant tool in data-intensive research in many fields such as astrophysics and space, meteorology, machine learning, big data analytics, time-series analytics, and modern finance. For instance, the OzSTAR supercomputer of the Centre for Astrophysics and Supercomputing has a theoretical peak speed with a Petaflops performance (Hurley, 2019). The Linux operating system, which is one of the most popular varieties of UNIX operating systems, is the main operating system of the OzSTAR supercomputer. Multi-tasking can be created via different methods, for example, through the message passing interface (MPI). MPI is a portable message-passing standard, which is developed by academia and industry researchers to function on a wide variety of parallel computing architectures. Portable message-passing programming languages are C, C++ and Fortran; see the details of the technical information on the OzSTAR in Swinburne University of Technology (2019) and also Linux tutorials for beginners in the Department of Electrical and Electronic Engineering at the University of Surrey (2019).

The next chapter is Chapter 2, the final chapter of Part I, Theoretical Framework. This chapter covers the research design, including the research gap, research questions and purpose, and provides an outline of the thesis. Next part is Part II where this part presents the main thesis studies where consisted with three papers in development to the portfolio optimisation. Note that the further and more intensive literature of thesis studies can be seen in this part, the Chapter 3, Chapter 4 and Chapter 5. Namely, it is the literature of asymmetric GARCH[1], mixture distribution[2], Bayesian inference[3], graph theory in multivariate analysis in a particular of copula model[4], variational Bayes

[1]See in Chapter 3

[2]See in Chapter 3

[3]See in Chapter 3 and Chapter 4

[4]See in Chapter 4 and Chapter 5

inference[5] and evolutionary algorithm in portfolio optimisation model[6].

[5] See in Chapter 4
[6] See in Chapter 5

Chapter 2

Research Design

2.1 Research Gap

The literature on portfolio risk management has rapidly and constantly developed in various shades of light. Some of the most active research questions are the study of conditional volatility models, non-normality, time-varying dependence, Bayesian inference and machine learning optimisation models. Therefore, this thesis will further explore these active research areas. Specifically, the study of conditional volatility models uses a popular class of generalised autoregressive conditional heteroscedasticity (GARCH), which is crucial for the conditional volatility model since it was introduced in Engle (1982) and generalised in Bollerslev (1986). The GARCH family in this thesis will be extended by incorporating a mixture distribution, which has received relatively little attention in previous research. Later, copulae will be incorporated into the model for the portfolio risk strategy, which are well-known to describe the dependence of financial time series. However, despite the widespread use of copula models in financial time series analysis, there has been relatively little research on their application to investment optimisation problems, particularly with respect to general vine copulae.

The parameter estimation method in this study is the efficient Bayesian Markov chain Monte Carlo (MCMC) method and variational Bayes method, which result in higher accuracy than the standard estimation method such as the maximum likelihood (MLE). However, Bayesian inference comparatively needs more computational time. Hence, a parallel computing technique will be applied to reduce the computational time of the Bayesian inference and all possible algorithms. Implementations in these research areas are still limited and require further investigation. The proposed forecasting models are initially based on asymmetric GARCH-R-Vine-Copula-Parallelisation-Bayesian with a mixture distribution approach and extended to the machine learning optimisation application in portfolio risk management.

2.2 Research Questions and Purpose

Based on the literature review and the significance of portfolio management, this study aims to address the research gap. Hence, the main research questions I aim to answer in this thesis are:

(1) What would be the performance of the model when implementing the Griddy Gibbs Bayesian MCMC method on ICAPM-EGARCH, incorporating a mixture distribution model? How does this compare to the performance of traditional models?

(2) What is the appropriate model to best describe the dependence structure among financial assets for portfolio management? For example, how can I obtain a better explanation of the relationship among asset returns to assess the portfolio risk in the financial market?

(3) How can I improve the computing performance of the estimation algorithm for the proposed portfolio strategy?

(4) What is the appropriate optimisation algorithm for portfolio management?

Therefore, this thesis applies advanced mathematical approaches to the dynamic portfolio allocation problem. In particular, the main contributions are as follows:

(a) The regular vine copula model was applied, which provided suitable forecasting of financial returns in multivariate analysis. Magnitude thirteen bivariate copula functions are a candidate in the regular vine copula model. This thesis also applied a leading asymmetric generalised autoregressive conditional heteroskedasticity model and intertemporal capital asset pricing model with mixture distribution to improve the copula-based forecasting model.

(b) This thesis proposes the efficient model parameter estimation methods including the Bayesian Markov chain Monte Carlo method and the variational Bayes method incorporated with a parallel computing technique.

(c) For the portfolio optimisation algorithm, the clonal selection algorithm and genetic algorithm in machine learning were applied to explore the optimal portfolio strategy.

2.3 Outline

To answer the above research questions and fulfil the purposes, the thesis is divided into three papers as follows:

Paper 1 implements an application of the Griddy Gibbs Bayesian MCMC method to ICAPM-EGARCH with the mixture distribution model to perform out-of-sample forecasts with the simulation experiment study. The innovation of this univariate model is the mixture distribution of two generalised Pareto distributions and a Gaussian distribution in the extreme value theory, while the benchmark innovation is generalised error distribution that can also capture extreme events. The empirical study of the proposed model and method is presented among seven stock markets, where the results are in the in-sample period before and during the COVID-19 pandemic and the out-of-sample period is during the COVID-19 pandemic to justify the use of the proposed model.

Paper 2 extends the first paper to the study of the time-varying dependence model in high n dimensions using the advanced regular vine copula model in graph theory. In the graph theory, the regular vine model can be represented in n by n matrix where three matrices are required in order to formulate the model. Note that the regular vine copula model with matrix representation is a generalisation of the vine copula family, where there are two-special subclasses of vine copula models that include canonical vine copula and drawable vine copula where the matrix approach of the regular vine model is also compatible. Besides this, another main contribution of this paper is the application of Bayesian inference and machine learning to the multivariate regular vine model with thirteen candidates of bivariate copula functions. The procedure of the algorithms is the Inference for the Margins method, also known as the IFM method. This paper proposes the extension of the variational Bayes method with (and without) data augmentation estimation of Bayesian machine learning scheme and random-walk Metropolis-Hasting estimation in Bayesian inference to the multivariate regular vine copula model. This paper presents the simulation experiments, and includes the proposed algorithm. Furthermore, this paper is a study providing evidence of the proposed algorithm in high dimensional data in the financial stock market.

Paper 3 represents the ultimate thesis extension of the application of optimal portfolio weight allocation in an investment portfolio strategy. This paper uses the proposed regular vine based model in Paper 2 as a forecasting model and follows the IFM method to formulate the optimisation problems and include the problem of CET optimisation, expected return/VaR optimisation and expected return/CVaR optimisation. The main constraint function I proposed is a realistic constraint where the investment position is only a long position. Finally, this paper initiates the clonal selection algorithm and genetic algorithm in computational intelligence to the proposed optimisation problems. The empirical study of the proposed optimal portfolio strategy with a variety of scenarios has determined and measured the portfolio strategy performance using the two important financial products - one is a cryptocurrency and another financial product is equity in up to a 35-dimensional portfolio optimisation problem. In addition, the study implements the fundamental analysis of stock selection to the proposed portfolio strategy, which is one of the successful and crucial keys in portfolio management.

It is worth mentioning that the parallel computing technique is applied to all proposed algorithms in this thesis. See the particular algorithm in each paper in Part II. All proposed models in the thesis generally outperform the competing model and the traditional model in the literature.

Part II Paper

Chapter 3

Modelling and Forecasting COVID-19 Stock Returns using Asymmetric GARCH-ICAPM with Mixture and Heavy-Tailed Distributions

3.1 Abstract

COVID-19 pandemic is an extreme event that created turmoil in stock markets around the world. This unexpected circumstance poses a critical question of whether the prevailing models can help predict the plummets of indices, hence the returns. This study aims to analyse and forecast the daily stock returns using various generalised autoregressive conditional heteroscedastic (GARCH) models with intertemporal capital asset pricing structure and innovation following (1) a mixture of generalised Pareto and Gaussian distributions and (2) generalised error distribution that can capture extreme events. We also employ the parallel griddy Gibbs (GG) sampling, which is a Markov chain Monte Carlo method, to facilitate parameter estimation. Our simulation study shows that the GG estimation method outperforms the benchmark quasi-maximum likelihood estimation method. We then proceed to the empirical study of seven stock markets where the results from the in-sample period before the COVID-19 pandemic justify the use of the proposed GARCH models. The out-of-sample forecasts during the early COVID-19 period also show satisfactory results.

Keywords: *COVID-19 pandemic, Bayesian Markov chain Monte Carlo, Asymmetric GARCH, ICAPM, Mixture distribution, Generalised Pareto distribution.*

3.2 Introduction

It is widely known that stock markets are prone to shocks and the COVID-19 pandemic is the most recent shock that generates the instantaneous risk that most markets faced but to different degrees. Given the prevalent risk, having a suitable model to analyse stock returns is a challenging task. The trade-off between risk and expected return of a financial asset is a fundamental problem in financial econometrics as its functional form is still debated in the literature. One of the leading models is the intertemporal capital asset pricing model (ICAPM), which is originally introduced by Merton (1973). The ICAPM is a linear function between the expected return and the variance of the market portfolio's return. Due to its simplicity, this model has been widely used and remains crucial in financial econometrics. For instance, Bali and Engle (2010) study ICAPM with dynamic conditional correlations and determined daily intertemporal relation between the expected return and risk of 30 stocks in the Dow Jones Industrial Average. Later, Engle, Lilien and Robins (1987) introduce the generalised autoregressive conditional heteroscedasticity (GARCH)-in-mean model and specified the linear function for the risk-return relationship in the bond market. The model is called GARCH with linear-in-variance risk premium or GARCH-ICAPM or GARCH-M. However, it does not provide a robust solution for the estimation.

Due to its conditional volatility component, since ARCH was invented by Engle

(1982) and extended by Bollerslev (1986), the (G)ARCH family becomes a crucial model in financial time series analysis and forecasting. The GARCH model has evolved due to its properties that can easily fit with various probability distributions of financial returns. Many extensions of the GARCH family have been proposed to capture the important financial market characteristics that are volatility clustering, non-normal, fat-tailed, and asymmetrically distributed properties; see, among others, Engle (2004) for a survey.

Within the GARCH family, the exponential-type GARCH family is the popular model. Geweke (1986) and Milhøj (1987) recommend the LogGARCH while Nelson (1991) suggests the exponential GARCH (EGARCH). Zakoian (1994) proposes the threshold GARCH (TGARCH), whereas Bildirici and Ersin (2009) study the GARCH family models including GARCH, EGARCH, GJR-GARCH, TGARCH, nonlinear-GARCH, simple-asymmetric-GARCH, power-GARCH, asymmetric-power-GARCH, and nonlinear-power-GARCH. The GARCH models are also extended, for example, to the artificial neural network, called the ANN-GARCH-type model. Besides, GJR-GARCH, introduced by Glosten et al. (1993), is a popular class under the exponential-type GARCH model.

It is worth discussing stochastic volatility (SV) and SV-in-mean models, given that they provide interesting alternative options for conditional volatility modeling (Taylor, 1986; Hull & White, 1987). Over the past two decades, these models have gained considerable popularity in mathematical finance literature, often surpassing the recognition of the GARCH family. However, the study employs the GARCH family due to its applicability in extending the univariate model to a sophisticated multivariate copula model. To be more precise, the GARCH family is well-suited for constructing copulae, such as extracting standardised residuals (Hafner & Manner, 2012).

GARCH(1,1) appears to be the most popular specification from its simplicity with decent forecasting performance, as the autocorrelation of the conditional volatility and the innovation does not usually have a long-lasting effect. As mentioned earlier, various distributions have been applied to the innovation of the GARCH family. One class of distributions, that might fit well with the bullish and bearish natures of stock markets, is a mixture of distributions. Given the COVID-19 pandemic, which is a rare event that produces extreme negative returns, having the innovation that follows the mixture of distributions with an extreme distribution, e.g., generalised Pareto distribution (GPD), might improve the GARCH performance in the analysis and forecasting under the severe shocks. We also compare the performance of the GPD innovation with that of the generalised error distribution (GED), which is a heavy-tailed distribution.

This study has three major contributions. First, a hybrid of the exponential-type GARCH and the ICAPM models are proposed that statistically fit better to the real data and provide insightful information on financial returns. Precisely, the proposed model reveals the empirical risk-free rate and risk premium of the financial market while capturing important volatility clustering characteristics of the returns. Further, it shows that this combined model is easier to handle the coefficient terms of ICAPM than those of the standard GARCH models. It is also a constraint-free model that can facilitate the estimation for inference. See, among others, Francq, Wintenberger and Zakoïan (2013), Francq, Wintenberger and Zakoïan (2018), and Hafner and Kyriakopoulou (2021) for the stochastic properties of the exponential-type GARCH model, which is compatible with the ICAPM model. Second, the study applies a mixture of distributions that include a GPD to the exponential-type GARCH innovation. The mixture with the GPD allows us to analyse the returns in a pre-specified quantile level so that the model can capture the tail behaviour or extreme events, i.e., the lower and upper tails in the case. See Sahamkhadam et al. (2018) and Caporale and Zekokh (2019), among others, for more details on the related mixture distributions. Third, the study implements the proposed models using the Bayesian griddy Gibbs (GG) method with a parallel computing technique that results in better accuracy performance than that of traditional QMLE.

Recently, a popular estimation method for the exponential-type GARCH model is a quasi-maximum likelihood (QMLE) approach, see Francq et al. (2013) and Hafner and Kyriakopoulou (2021). However, in general, the QMLE estimator is known for its dependence on initial conditions and is less statistically efficient for this type of model. Hence, the study uses the Bayesian Markov chain Monte Carlo (MCMC) that provides robustness and superiority of forecasting performance over that of the QMLE. Note, however, that the MCMC method requires further mathematical complexity and more computational time. See Chang et al. (2018), Kolm et al. (2014), Pareek and Thakkar (2015), and Patton (2012), among others, for the relevant survey of estimation method using the QMLE and the MCMC. To facilitate the estimation and reduce the computational time for the MCMC, many algorithms have been proposed. Andrieu, Doucet and Holenstein (2010) and Creal and Tsay (2015) use particle Gibbs sampling algorithm while Wichitaksorn and Choy (2015) and Ausín and Galeano (2007) use the GG sampling. Multiple-try Metropolis forward filtering backward sampling (FFBS) approach and the multiple-trial Metropolis independent sampler FFBS approach are applied in Billio et al. (2016). See also Ardia et al. (2018) and F. Li and Kang (2018) for the Metropolis-Hastings algorithm and its variations.

Hence, in this study, the study chooses the MCMC approach to estimate the proposed exponential-type GARCH-ICAPM-Mixture model. To mitigate the adverse effect of longer computational time, the study incorporates the parallel computing technique, into the GG algorithm. This allows us to compute a massive amount of data at the same time. See Benczúr (2011), Fujiwara et al. (2018), Lam et al. (2019), Lopatka and Czyzewski (2014), Prudencio and Cheung (2012), Sanchez-Vazquez et al. (2014) and Wichitaksorn and Choy (2015), among others, for more details on the parallel computing technique. Based on the simulation study results, the parallel-GG algorithm outperforms the QMLE. The empirical study using the data during the early period of the COVID-19 pandemic also shows the favorable forecasting performance of the proposed models and method.

The rest of the paper is organised as follows. Section 3.3 presents the class of exponential-type and GJR-GARCH with mixture distribution for the innovation and discusses the Bayesian MCMC method using the parallel computing technique. Section 3.4 performs the simulation study. Section 3.5 shows the empirical study results. Section

3.6 concludes.

3.3 Models and Estimation

This section introduces the proposed models where this study includes the ICAPM structure in the observation equation of GARCH, the exponential-type GARCH, and the GJR-GARCH, with the innovation following the mixture of the Gaussian distribution and the heavy-tailed GPD. In addition, the study facilitates the model estimation using the GG algorithm together with the parallel computing technique. To assess the performance of the proposed models and method, the study compares them with the EGARCH-ICAPM using the generalised error distribution (GED) innovation as the benchmark model and the quasi-maximum likelihood estimation as the benchmark method.

3.3.1 Models

We first consider the ICAPM given in Merton (1973), which is a linear-in-variance function of a financial return series. For $t = 1, ..., T$, let y_t denote a financial return series. Based on Hafner and Kyriakopoulou (2021), the ICAPM is then given by

$$y_t = \lambda_1 + \lambda_2 \mathbf{h}_t + \varepsilon_t, \tag{3.1}$$

where t is the time index, λ_1 is the expected risk-free rate with the hypothetical value of zero, λ_2 is the expected (positive) risk premium, $\mathbf{h}_t$ is the conditional volatility, $\varepsilon_t = \sqrt{\mathbf{h}_t} z_t$ is the innovation with mean zero and the conditional variance (or volatility) $\mathbf{h}_t$, and z_t is the independent and identically distributed (iid) random variable with zero mean and unit variance. Note that let λ_1 be an unknown and random that needs to be estimated while it is known and fixed in Hafner and Kyriakopoulou (2021). This ICAPM structure is then applied to all GARCH models implemented in this study.

The specifications the study used to model the conditional volatility $\mathbf{h}_t$ include the typical GARCH(p,q), the exponential-type GARCH that are EGARCH(p,q,r) and

LogGARCH(p,q), and the GJR-GARCH(p,q). Prior works indicated that the order lags for the innovation and volatility terms are best performing when $p = 1$ and $q = 1$, then follow that in this analysis. For illustrative purposes, the description below is the specification of all GARCH models used here in the details.

The typical GARCH(p,q) model is used as the starting model where the study follows Bollerslev (1986) and given by

$$\mathbf{h}_t = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \nu_j \mathbf{h}_{t-j},$$

where $p \geq 0, q > 0,\ \omega > 0,\ \alpha_i \geq 0, \nu_j \geq 0,\ 0 \leq \alpha_i + \nu_j \leq 1,\ 1 \leq\ i \leq\ p, 1 \leq\ j \leq\ q$ and p, q are the discrete value to ensure the conditional variance h_t being an almost surely strictly positive or a strictly stationary process. In this model, the parameter space of volatility coefficients is denoted as $\theta = \{\omega, \alpha_i, \nu_j, 1 \leqslant i \leqslant p, 1 \leqslant j \leqslant q\}$.

We then turn to a popular GARCH model extension, which is an exponential-GARCH(p,q,r) or EGARCH(p,q,r) of Nelson (1991) and given by

$$\log \mathbf{h}_t = \omega + \sum_{i=1}^{p} \gamma_i z_{t-i} + \sum_{j=1}^{q} \delta_j |z|_{t-j} + \sum_{k=1}^{r} \nu_k \log \mathbf{h}_{t-k},$$

where the conditional volatility is now log volatility. The parameter space is then $\theta = \{-\infty \leqslant \omega, \gamma_i, \delta_j \leqslant \infty, |\nu_k| < 1, 1 \leqslant i \leqslant p, 1 \leqslant j \leqslant q, 1 \leqslant k \leqslant r\}$. The "news impact" function $g(z_{t-1})$ of the EGARCH model is $\omega + \sum_{i=1}^{p} \gamma_i z_{t-i} + \sum_{j=1}^{q} \delta_j |z|_{t-j}$ while ν is the volatility persistence parameter, γ and δ are the shock asymmetry and the size effect, respectively. If $\gamma > 0$, that means positive shocks increase the volatility more than negative shocks of the same size and vice versa. For the review of the GARCH family including the exponential-type GARCH and the GJR-GARCH, see, among others, Bera and Higgins (1993), Hentschel (1995), and Jose Rodriguez and Ruiz (2012).

Another popular exponential-type GARCH is the LogGARCH(p,q) from Francq et al. (2018) and given by

$$\log \mathbf{h}_t = \omega + \sum_{i=1}^{p} \left(\alpha_i^{+} I_{\{\varepsilon_{t-i}>0\}} + \alpha_i^{-} I_{\{\varepsilon_{t-i}<0\}} \right) \log \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \nu_j \log \mathbf{h}_{t-j},$$

where I is an indicator function. The parameter space is now $\theta = \{-\infty \leqslant \omega, \alpha_i^{+}, \alpha_i^{-} \leqslant \infty, |\nu_j| < 1, 1 \leqslant i \leqslant p, 1 \leqslant j \leqslant q\}$ for the existence of a stationary solution. The news impact function of this LogGARCH model is $\omega + \sum_{i=1}^{p} \left(\alpha_i^{+} I_{\{\varepsilon_{t-i}>0\}} + \alpha_i^{-} I_{\{\varepsilon_{t-i}<0\}} \right) \log \varepsilon_{t-i}^2$. When $\alpha_i^{-} = \alpha_i^{+}$, the model becomes the symmetric LogGARCH process. Note that the LogGARCH has a well-known problem, i.e., if the ε_{t-i} term equals zero, the conditional volatility cannot be generated. For the discussion of the LogGARCH process, see, among others, Sucarrat and Escribano (2018), and Sucarrat, Gronneberg and Escribano (2016). For the recent studies of the exponential-type GARCH process, see, Hafner and Kyriakopoulou (2021), Caporale and Zekokh (2019), Francq et al. (2018), and Bildirici and Ersin (2009), among others.

The GJR-GARCH(p,q) by Glosten et al. (1993) is the last model the study used in the analysis. This model becomes better known among the GARCH family and given by

$$\mathbf{h}_t = \omega + \sum_{i=1}^{p} \left(\alpha_i + \gamma_i I_{\{\varepsilon_{t-i}<0\}} \right) \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \nu_j \mathbf{h}_{t-j}$$

where $\omega > 0, \alpha_i, \nu_j \geq 0,\ \nu_j + \gamma_i \geq 0$. α_i and γ_i are the size effect and shock asymmetry, respectively. Hence, the parameter space is $\theta = \{\omega, \alpha_i, \gamma_i, \nu_j, 1 \leqslant i \leqslant p, 1 \leqslant j \leqslant q\}$ while the news impact function of the GJR-GARCH is $\omega + \sum_{i=1}^{p} \left(\alpha_i + \gamma_i I_{\{\varepsilon_{t-i}<0\}} \right) \varepsilon_{t-i}^2$. See, among others, Goel et al. (2019), Caporale and Zekokh (2019), Oh and Patton (2018), Fengler and Okhrin (2016), and De Lira Salvatierra and Patton (2015), for the recent studies and discussion of this type of model.

3.3.2 Model Innovations

With all GARCH model specifications mentioned above, the study applies two types of innovations including the mixture distribution and the GED. The former is what the study expect to perform well in the COVID-19 situation while the latter is the benchmark for comparison. In the literature, the mixture distributions have been used in data-intensive research in many fields such as astrophysics and space, meteorology, machine learning, big data analytics, time-series analytics, and modern finance.

Mixture Distribution

The mixture distribution can be represented by either a finite set of cumulative distribution functions (cdf), $P_k(z_t)$, or probability density functions (pdf), $p_k(z_t)$, for $k = 1, \ldots, n$. Hence, the mixture distribution function and the mixture density function, respectively, are a sum such that

$$F(z_t) = \sum_{k=1}^{n} w_k P_k(z_t),$$

$$f(z_t) = \sum_{k=1}^{n} w_k p_k(z_t).$$

where weight $w_k > 0$ and $\sum w_k = 1$. In financial time series, the mixture distribution is popularly applied due to its flexibility in capturing asymmetrical tails of financial returns. If the mixing weight w_k is pre-specified, it can also represent the quantile level of the mixture distribution.

This study follows Sahamkhadam et al. (2018) that use the mixture of Gaussian and GPD densities. Precisely, the study have the GPD at the lower and upper tails to capture the extreme (rare) events while the Gaussian can represent the usual (normal) circumstances. We expect asymmetric-GARCH-ICAPM models with this mixture distribution can fit well with the returns under the COVID-19 pandemic. The mixture pdf is then given by

$$p(z_t) = \begin{cases} \frac{1}{\beta^L}\left(1 + \xi^L \frac{(z_t - \mu^L)}{\beta^L}\right)^{-\frac{1}{\xi^L} - 1} & , z_t \leq \Phi^{-1}(a) \\ \phi(z_t) & , \Phi^{-1}(a) < z_t < \Phi^{-1}(b) \\ \frac{1}{\beta^R}\left(1 + \xi^R \frac{(z_t - \mu^R)}{\beta^R}\right)^{-\frac{1}{\xi^R} - 1} & , z_t \geq \Phi^{-1}(b) \end{cases} \tag{3.2}$$

where a and b are a pre-specified quantile, $-\infty < \mu < \infty$ is a location parameter, $0 < \beta < \infty$ is a scale parameter, $-\infty < \xi < \infty$ is a shape parameter, and L and R denote the lower and upper tails, respectively. The support of z_t of GPD is $z_t \geq 0$ when $\xi \geq 0$ and $\mu \leq z_t \leq \mu - \beta/\xi$ when $\xi < 0$, and $0 \leq \xi < 0.5$ represents a fat tail and exists at least up to second moment.

The GPD consists of, in the sense of generalisation, an ordinary Pareto distribution when $\xi > 0$, an exponential distribution when $\xi = 0$, and a short-tailed, Pareto type II distribution when $\xi < 0$. We denote $X \sim GPD(\mu, \beta, \xi)$, therefore, $E(X) = \mu + \frac{\beta}{1-\xi}$ when $\xi < 1$ and $Var(X) = \frac{\beta^2}{(1-\xi)^2(1-2\xi)}$ where $\xi < 0.5$ to ensure the variance is defined, see, e.g., Hosking and Wallis (1987) and Singh and Guo (1995). While $\phi(\cdot)$ and $\Phi^{-1}(\cdot)$ are, respectively, the standard Gaussian pdf and inverse cdf.

GED Innovation

For the sake of comparison, the study also applies the GED innovation, proposed by Nadarajah (2005), to test its forecasting performance under extreme events. The GED is a useful distribution in time series analysis and is also known as the exponential power distribution. Let $X \sim GED(\mu, \beta, \xi)$. It includes the Gaussian and Laplace distributions, when $\xi = 2$ and $\xi = 1$, respectively. As $\xi \to \infty$, the pdf converges pointwise to a uniform pdf on $(\mu - \infty, \mu + \infty)$. When $\xi < 2$ represents fat-tail distribution and $\xi > 2$ represents lighter-tail distribution than normal.

With the observation equation 3.1, the GED pdf for the innovation is given by

$$p(z_t) = \frac{\xi}{2\beta\Gamma(1/\xi)} e^{-(|z_t-\mu|/\beta)^{\xi}}$$

where $-\infty < \mu < \infty$, $0 < \beta < \infty$ and $0 < \xi < \infty$ are the location, scale and shape parameter, respectively. The support of z_t is $-\infty < z_t < \infty$.

3.3.3 Posterior Inference

Likelihood Function and Prior

In the estimation of all GARCH-ICAPM models in the study, the study uses the Bayesian MCMC method as the study expects it to return better results than those of the traditional method, e.g., quasi-maximum likelihood estimation (QMLE). Results from the simulation study confirm this by showing that the Bayesian method fits better than the QMLE in terms of the Akaike information criterion (AIC) and Bayesian information criterion (BIC), see Section 3.4 for the estimation results. In addition, all of the model likelihoods are intractable, see equation 3.3. Using the sampling method from the Bayesian MCMC can conveniently get the parameter estimates than the mountain-climbing QMLE. Hence, with the Bayesian MCMC, the study needs to estimate the model parameters using the posterior density, which is proportional to the prior and the likelihood as $p(\theta|\, y) \propto p(\theta)\, L(y|\theta)$ where $p(\theta|\, y)$ is the posterior density, $p(\theta)$ is the prior density, and $L(y|\theta)$ is the likelihood function.

Following from the observation equation 3.1, let $z_t = (y_t - \lambda_1 - \lambda_2 h_t)/\sqrt{h_t}$. To standardise the innovation with the mixture distribution in equation 3.2, the study sets $\beta = (1-\xi)(1-2\xi)^{1/2}$. Therefore, the log-likelihood function of the proposed model with the mixture of Gaussian and two-sided GPD innovation is then

$$
\begin{aligned}
\ell(\theta) &= \sum_{t=1}^{T} log[q^L p_L^{GPD}(z_t) + (1-q^L-q^R)p^G(z_t) + q^R p_R^{GPD}(z_t)] \\
p_L^{GPD}(z_t) &= \frac{1}{\sqrt{\mathbf{h}_t}(1-\xi^L)(1-2\xi^L)^{1/2}}\left(1+\frac{\xi^L(y_t-\lambda_1-\lambda_2\mathbf{h}_t)}{\sqrt{\mathbf{h}_t}(1-\xi^L)(1-2\xi^L)^{1/2}}\right)^{-\frac{1}{\xi^L}-1}, \\
p^G(z_t) &= \frac{1}{(2\pi\mathbf{h}_t)^{1/2}} exp\left[\frac{-(y_t-\lambda_1-\lambda_2\mathbf{h}_t)^2}{2\mathbf{h}_t}\right], \\
p_R^{GPD}(z_t) &= \frac{1}{\sqrt{\mathbf{h}_t}(1-\xi^R)(1-2\xi^R)^{1/2}}\left(1+\frac{\xi^R(y_t-\lambda_1-\lambda_2\mathbf{h}_t)}{\sqrt{\mathbf{h}_t}(1-\xi^R)(1-2\xi^R)^{1/2}}\right)^{-\frac{1}{\xi^R}-1}
\end{aligned}
\tag{3.3}
$$

where q is a pre-specified quantile level or weight of a pdf, $\mathbf{h}_t$ is the conditional variance for the all GARCH family including GARCH(1,1) and GJR-GARCH(1,1) while $\mathbf{h}_t$ of EGARCH(1,1,1) and LogGARCH(1,1) is the conditional log-variance. Note that the pre-specified quantile q can be defined as the left and right boundary of the distribution tail where the study follows Sahamkhadam et al. (2018) and Z.-R. Wang, Chen, Jin and Zhou (2010) to set q^L and q^R at 0.10. However, to assess the ability of the proposed models in capturing the extreme events, the study also sets q at 0.01 and 0.05 in the simulation study.

For the EGARCH(p,q,r)-ICAPM model as that in Hafner and Kyriakopoulou (2021), with the standardised GED innovation where the study set $\beta = \sqrt{\Gamma(1/\xi)/\Gamma(3/\xi)}$, the log-likelihood function is then given by

$$\begin{aligned}\ell(\theta) = (T-1)\log\left(\frac{\Gamma(3/\xi)^{1/2}}{\Gamma(1/\xi)^{3/2}}\xi\right) - 2\sum_{t=2}^{T}\log\left(\sqrt{exp(\mathbf{h}_t)}\right) \\ - \sum_{t=2}^{T}\left(\left|\frac{y_t - \lambda_1 - \lambda_2 exp(\mathbf{h}_t)}{\sqrt{exp(\mathbf{h}_t)}}\right| / \sqrt{\frac{\Gamma(1/\xi)}{\Gamma(3/\xi)}}\right)^{\xi}\end{aligned} \tag{3.4}$$

where $\Gamma(\cdot)$ is a gamma function.

In the posterior inference, prior specifications are a crucial part as it determines the robustness and accuracy of the posterior statistics, e.g., mean and standard deviation. The parameters of interest include the risk premium, which is expected to be positive, the risk-free rate to be very close to zero, and the conditional volatility parameters and their innovation parameters. With different specifications of those parameters, the study needs to apply different priors to different model parameters. We set priors for the GARCH(1,1)-ICAPM-Mixture process, as

$$\begin{aligned}\lambda_1 &\backsim N\left(\mu_{\lambda_1}, \sigma^2_{\lambda_1}\right), & \omega &\backsim B(\alpha_\omega, \beta_\omega), \\ \lambda_2, &\backsim G\left(\alpha_{\lambda_2}, \beta_{\lambda_2}\right), & \alpha &\backsim B(\alpha_\alpha, \beta_\alpha), \\ \xi^L &\backsim G\left(\alpha_{\xi^L}, \beta_{\xi^L}\right), & \nu &\backsim G(\alpha_\nu, \beta_\nu), \\ \xi^R &\backsim B\left(\alpha_{\xi^R}, \beta_{\xi^R}\right), & &\end{aligned}$$

for the EGARCH(1,1,1)-ICAPM-Mixture process, as

$$\lambda_1 \backsim N\left(\mu_{\lambda_1}, \sigma^2_{\lambda_1}\right), \qquad \delta \backsim B\left(\alpha_\delta, \beta_\delta\right),$$
$$\lambda_2 \backsim N\left(\mu_{\lambda_2}, \sigma^2_{\lambda_2}\right), \qquad \nu \backsim B\left(\alpha_\nu, \beta_\nu\right),$$
$$\omega \backsim N\left(\mu_\omega, \sigma^2_\omega\right), \qquad \xi^L \backsim Unif\left(a_{\xi^L}, b_{\xi^L}\right),$$
$$\gamma \backsim N\left(\mu_\gamma, \sigma^2_\gamma\right), \qquad \xi^R \backsim Unif\left(a_{\xi^R}, b_{\xi^R}\right),$$

for the LogGARCH(1,1)-ICAPM-Mixture process, as

$$\lambda_1 \backsim N\left(\mu_{\lambda_1}, \sigma^2_{\lambda_1}\right), \qquad \alpha^- \backsim B\left(\alpha_{\alpha^-}, \beta_{\alpha^-}\right),$$
$$\lambda_2 \backsim G\left(\alpha_{\lambda_2}, \beta_{\lambda_2}\right), \qquad \nu \backsim G\left(\alpha_\nu, \beta_\nu\right),$$
$$\omega \backsim B\left(\alpha_\omega, \beta_\omega\right), \qquad \xi^L \backsim G\left(\alpha_{\xi^L}, \beta_{\xi^L}\right),$$
$$\alpha^+ \backsim N\left(\mu_{\alpha^+}, \sigma^2_{\alpha^+}\right), \qquad \xi^R \backsim B\left(\alpha_{\xi^R}, \beta_{\xi^R}\right),$$

for GJR-GARCH(1,1)-ICAPM-Mixture process, as

$$\lambda_1 \backsim N\left(\mu_{\lambda_1}, \sigma^2_{\lambda_1}\right), \qquad \gamma \backsim G\left(\alpha_\gamma, \beta_\gamma\right),$$
$$\lambda_2 \backsim G\left(\alpha_{\lambda_2}, \beta_{\lambda_2}\right), \qquad \nu \backsim G\left(\alpha_\nu, \beta_\nu\right),$$
$$\omega \backsim B\left(\alpha_\omega, \beta_\omega\right), \qquad \xi^L \backsim G\left(\alpha_{\xi^L}, \beta_{\xi^L}\right),$$
$$\alpha \backsim B\left(\alpha_\alpha, \beta_\alpha\right), \qquad \xi^R \backsim B\left(\alpha_{\xi^R}, \beta_{\xi^R}\right),$$

and for the EGARCH(1,1,1)-ICAPM-GED process, as

$$\lambda_1 \backsim B\left(\alpha_{\lambda_1}, \beta_{\lambda_1}\right),$$
$$\lambda_2 \backsim N\left(\mu_{\lambda_2}, \sigma^2_{\lambda_2}\right), \qquad \gamma \backsim N\left(\mu_\gamma, \sigma^2_\gamma\right),$$
$$\omega \backsim N\left(\mu_\omega, \sigma^2_\omega\right), \qquad \delta \backsim G\left(\alpha_\delta, \beta_\delta\right),$$
$$\nu \backsim G\left(\alpha_\nu, \beta_\nu\right), \qquad \xi \backsim G\left(\alpha_\xi, \beta_\xi\right),$$

where $B(\cdot,\cdot)$ denotes beta distribution, $Unif(\cdot,\cdot)$ denotes uniform distribution, and $G(\cdot,\cdot)$ denotes gamma distribution. These priors and their hyperparameter values are chosen based on their forecasting performance.

MCMC Algorithm

With the priors given above, the posterior densities become intractable. Hence, to facilitate the parameter estimation, the study employs the GG sampling to obtain the posterior statistics for the model parameters that have different supports. Under the GG, each model parameter is sampled one-by-one directly from the univariate conditional posterior density. This algorithm is similar to Gibbs sampling, which accepts all samples, but it is approximated at a finite number of grid points. See, among others, Wichitaksorn and Choy (2015) and Ritter and Tanner (1992) for more details on the GG sampling.

Though the GG algorithm fits well with the intractable posteriors for parameters with different constraints, it requires a lot of computational time as it needs to evaluate all grid points; the higher number of grid points, the more accurate the model parameter is. Hence, the study follows Wichitaksorn and Choy (2015) to parallelise the grid points and evaluate them at the same time. This parallel computing technique can, more or less, fasten the GG estimation procedure. Note that the parallel GG may not be reversible but it is simpler to implement, less computationally expensive, and has significantly smaller autocorrelation coefficients of the Markov chains generation than other variant methods. See more discussion on the convergence of Griddy Gibbs sampling and other perturbed Markov chains in Dinh, Rundell and Buzzard (2017). The generic algorithm below shows the parallel-GG estimation steps that can be applied to all models.

Parallel-Griddy-Gibbs Sampling Algorithm:

Let $\Theta = \{\theta_1, \dots, \theta_K\}$ and $s = 1, 2, \dots, S$, respectively, denote the parameter space and the iterations.

1. Take an initial vector of the estimates either from a random vector or the QMLE method, $\hat{\theta}_k^{(0)}, k = 1, 2, \dots, K$.

2. Repeat for $s = 1, 2, \dots, S$, and for $k = 1, 2, \dots, K$.

(a) Draw candidate $\hat{\theta}_k^{(s)}$ where $\hat{\theta}_k^{(s)}$ is divided into G grid points as $\hat{\theta}_k^{(s,1)}, \ldots, \hat{\theta}_k^{(s,G)}$.

(b) Calculate full conditional posterior density for each grid point $p(\hat{\theta}_k^{(s,g)} | \hat{\theta}_1^{(s,g)}, \cdots, \hat{\theta}_{k-1}^{(s,g)}, \hat{\theta}_{k+1}^{(s-1,g)}, \cdots, \hat{\theta}_K^{(s-1,g)})$ and let $p(\hat{\theta}_k^{(s,g)} | \cdot) = w_k^g$ where $g = 1, \cdots, G$.

(c) Normalise w_k^g as $\widetilde{w}_k^g = \frac{w_k^g}{\sum_{g=1}^G w_k^g}$ and cumulative sum $\widetilde{w}_k^g$ implies the empirical cdf $F\left(\hat{\theta}_k^{(s,g)}\right)$ where $F\left(\hat{\theta}_k^{(s,G)}\right) = \widetilde{w}_k^G = 1$

(d) Draw $u \sim Unif\,(0, 1)$ and set $\hat{\theta}_k^{(s)} = \hat{\theta}_k^{(s,g)}$ if $F(\widehat{\theta}_k^{(s,g-1)}) < u \leq F(\widehat{\theta}_k^{(s,g)})$

3. Repeat Step 2 for $\hat{\theta}_1^{(s)}, \hat{\theta}_2^{(s)}, \ldots, \hat{\theta}_K^{(s)}$.

It is essential to discard an initial set of the posterior draws to remove the effect of the initial conditions. Moreover, the initial range is an important factor to obtain the robustness of model parameters and the initial range can be obtained through some estimation methods. This is known as adaptive GG sampling. In this paper, the study uses the QMLE method by Hafner and Kyriakopoulou (2021) to obtain the initial range of the values. Note in Step (d) that u will always fall into an interval, which is between two grid points, e.g., $g - 1$ and g. For example, if u is less than or equal $F(\hat{\theta}_k^{(s,g-1)})$ but greater than $F(\hat{\theta}_k^{(s,g-1)})$, $\hat{\theta}_k^{(s)}$ will be set as $\hat{\theta}_k^{(s,g-1)}$. However, if u is above $F(\hat{\theta}_k^{(s,g)})$ but less than or equal $F(\hat{\theta}_k^{(s,g+1)})$, $\hat{\theta}_k^{(s)}$ will be set as $\hat{\theta}_k^{(s,g+1)}$. Since the study parallelises the grid-point evaluations, the estimation method is the Parallelisation-adaptive-GG sampling method. Another important component in the GG sampling is the choice of the number of grid points where the study needs to trade-off between the accuracy and the computational time Dinh et al. (2017). The number of grid points need not be the same for all iterations. It can be adjusted to be adaptive. In the end, the study collects the posterior samples to produce relevant statistics and densities for inference.

3.4 Simulation Study

In the simulation study, the study aims to assess the performance of the proposed GARCH models including the GARCH(1,1)-ICAPM-Mixture, EGARCH(1,1,1)-ICAPM-Mixture, LogGARCH(1,1)-ICAPM-Mixture, GJR-GARCH(1,1)-ICAPM-Mixture and EGARCH(1,1,1)-ICAPM-GED, and the parallel-GG estimation method. In all cases, the study compares the estimation results between the parallel-GG with those of the full one-step QMLE using 100 datasets. Each dataset generates 1000 simulated observations for the proposed models. The vector of initial value $\hat{\theta}_k^{(0)}$ is a vector of random numbers where the study found there is a convergence issue for the full one-step QMLE method while the proposed parallel-GG method converges well.

Note that all true parameter values in the simulation study were chosen based on the empirical data and prior works. The likelihood function in Equations 3.3-3.4 is essential for model calibration, either in terms of the full one-step QMLE or the GG method. The QMLE is performed by maximizing the log-likelihood function and this optimization method can be computed using the *fmincon* function in MATLAB.

According to the GG setting, the study ran 3000 MCMC iterations with 300 grid points including 500 burn-ins, and then, kept every fifth draw, which yielded 500 posterior samples. We implemented the experiments in MATLAB and performed them on a desktop computer with an Intel Core i7-9700 CPU, 3.00 GHz, and 32 GB RAM. The prior distribution of all proposed models used in the simulation study can be found in Table A1 in the supplementary material. To compare across models, the study calculated the simulation inefficiency factor (SIF); see, e.g., Siddhartha and Edward (1996), the standard deviation (SD), the mean absolute deviation (MAD), and the computational time. Also, AIC, BIC, and deviance information criterion (DIC) (except for the QMLE method) were used to assess the in-sample fit.

The results in Table 3.1 show that the proposed models using the parallel-GG method are more favorable than those of the QMLE. Our parameter estimates are close to their true values and have reasonable SIFs. Precisely, the MADs are lower for the MCMC estimates. Furthermore, according to the average AICs and BICs, the MCMC algorithm does better, except for the EGARCH(1,1,1)-ICAPM-GED. The one-step QMLE could have a convergence issue as the SDs are significantly variable and of high value, which the study chose not to report in Table 3.1.

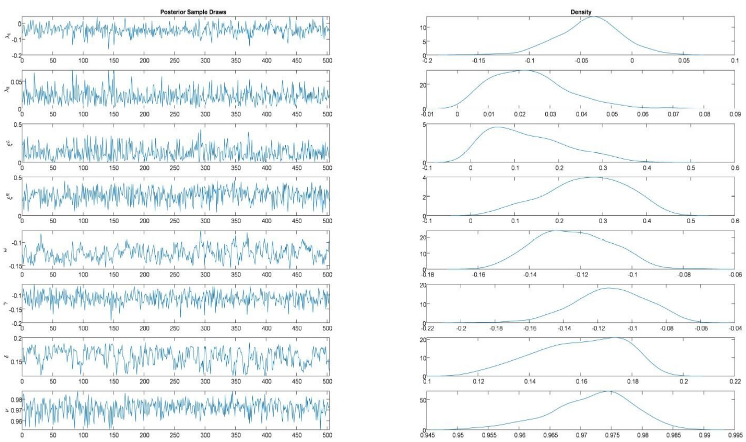
Table 3.1: Estimation Results of Various Models in Simulation Study

Models	Parameters	True Value	Parallel-GG				One-step QMLE	
			Mean	SD	SIF	MAD	Mean	MAD
GARCH-ICAPM-Mixture	λ_1	0.0000	0.0000	0.0012	2.4774	**0.0000**	0.0017	0.0017
	λ_2	0.0426	0.0454	0.0061	2.1778	**0.0028**	0.0909	0.0483
	ξ^L	0.0654	0.0656	0.0258	2.2808	**0.0002**	0.0991	0.0337
	ξ^R	0.3141	0.3080	0.0945	2.2174	**0.0061**	0.2985	0.0156
	ω	0.0080	0.0087	0.0026	2.1067	**0.0007**	0.0099	0.0019
	α	0.0880	0.0861	0.0033	2.1073	**0.0019**	0.0180	0.0700
	ν	0.8980	0.8987	0.0642	2.1594	**0.0007**	0.9787	0.0807
	AIC		**1596.40**				2548.27	
	BIC		**1630.76**				2582.63	
	DIC		2103.64				n.a.	
	Time		442.51				0.01	
EGARCH-ICAPM-Mixture	λ_1	0.0000	-0.0448	0.0356	2.3378	0.0448	0.0195	**0.0195**
	λ_2	0.0200	0.0466	0.0202	2.2520	**0.0266**	0.0599	0.0399
	ξ^L	0.1500	0.1722	0.0860	2.3734	**0.0222**	0.1246	0.0254
	ξ^R	0.3500	0.2940	0.0838	2.5761	0.0560	0.3624	**0.0124**
	ω	-0.1000	-0.1011	0.0181	5.6055	**0.0011**	-0.0866	0.0134
	γ	-0.1200	-0.1313	0.0206	2.0793	0.0113	-0.1222	**0.0022**
	δ	0.1300	0.1375	0.0223	6.3039	**0.0075**	0.2074	0.0774
	ν	0.9800	0.9603	0.0104	3.3496	**0.0197**	0.8327	0.1473
	AIC		**3091.61**				3141.54	
	BIC		**3130.88**				3180.80	
	DIC		3143.56				n.a.	
	Time		1582.34				1.04	
LogGARCH-ICAPM-Mixture	λ_1	0.0000	0.0000	0.0001	2.2997	**0.0000**	0.0001	0.0001
	λ_2	0.0450	0.0434	0.0042	2.0423	**0.0016**	0.0330	0.0120
	ξ^L	0.0700	0.0705	0.0084	2.2974	**0.0005**	0.1122	0.0422
	ξ^R	0.2800	0.3243	0.0550	2.1204	0.0443	0.2586	**0.0214**
	ω	0.0320	0.0381	0.0020	2.5199	0.0061	0.0354	**0.0034**
	α^+	0.0030	0.0028	0.0018	2.3736	**0.0002**	0.0066	0.0036
	α^-	0.0530	0.0557	0.0040	2.6059	**0.0027**	0.0563	0.0033
	ν	0.9600	0.9622	0.0028	2.5192	**0.0022**	0.9232	0.0368
	AIC		**2562.25**				2569.77	
	BIC		**2601.51**				2609.03	
	DIC		2547.88				n.a.	
	Time		2012.44				1.33	
GJRGARCH-ICAPM-Mixture	λ_1	0.0000	0.0000	0.0001	2.5272	**0.0000**	0.0001	0.0001
	λ_2	0.0350	0.0349	0.0035	2.0742	**0.0001**	0.0515	0.0165
	ξ^L	0.0700	0.0703	0.0084	2.4086	**0.0003**	0.1217	0.0517
	ξ^R	0.2800	0.2311	0.0462	2.1888	0.0489	0.2516	**0.0284**
	ω	0.0080	0.0087	0.0006	2.1908	**0.0007**	0.0095	0.0015
	α	0.0310	0.0312	0.0032	2.1718	**0.0002**	0.0294	0.0016
	γ	0.0070	0.0072	0.0021	2.2443	**0.0002**	0.0132	0.0062
	ν	0.9500	0.9494	0.0026	2.2629	**0.0006**	0.9478	0.0022
	AIC		**2307.76**				2340.01	
	BIC		**2347.02**				2379.27	
	DIC		2294.11				n.a.	
	Time		1452.25				0.68	
EGARCH-ICAPM-GED	λ_1	0.0000	0.0006	0.0002	2.3395	**0.0006**	0.0072	0.0072
	λ_2	0.0540	0.0612	0.0078	2.1796	**0.0072**	0.0702	0.0162
	ξ	1.2500	1.3004	0.0839	3.5385	**0.0504**	1.0075	0.2425
	ω	-0.1020	-0.0976	0.0035	2.7592	**0.0044**	-0.1701	0.0681
	γ	-0.1780	-0.1084	0.0097	2.2352	**0.0696**	-0.1014	0.0767
	δ	0.0550	0.0452	0.0028	2.1275	**0.0098**	0.0258	0.0292
	ν	0.9540	0.9745	0.0029	2.7787	0.0205	0.9488	**0.0052**
	AIC		-1712.20				**-1879.07**	
	BIC		-1677.85				**-1844.72**	
	DIC		-1709.20				n.a.	
	Time		403.52				0.56	

Notes that Time = computational time (seconds). n.a. = not available. Bold font indicates better results.

The computational time is higher for the asymmetric-GARCH-ICAPM with mixture innovation than that of the GED as there are more parameters in the model. As an example, the convergence draws in Figure 3.1 show that the GG-MCMC algorithm for the EGARCH(1,1,1)-ICAPM with mixture innovation process converges well. The convergence draws of other models can be found in the supplementary material.

Figure 3.1: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH-ICAPM-Mixture Process



To assess the abilities in capturing the extreme events of the proposed models, the study also estimates, as an example, the EGARCH-ICAPM-Mixture model in extreme quantiles including 0.01, 0.05, and 0.10. The results in Table 3.2 show that for lower quantiles QMLE seems to return lower MAD for some parameters. However, for the overall model, based on the AIC and BIC the parallel-GG method performs better.

Table 3.2: Estimation Results of EGARCH-ICAPM-Mixture Model at Lower Quantiles in Simulation Study

		Parallel-Computing-GG				1-step QMLE	
0.01quantile							
		Mean	SD	SIF	MAD	Mean	MAD
λ_1	0.0000	-0.0320	0.0359	2.2372	0.0320	0.0006	**0.0006**
λ_2	0.0200	0.0504	0.0220	2.0991	**0.0304**	0.0587	0.0387
ξ^L	0.1500	0.2508	0.1392	2.1993	0.1008	0.1014	**0.0486**
ξ^U	0.3500	0.2606	0.1379	2.1551	0.0894	0.3392	**0.0108**
ω	-0.1000	-0.1060	0.0180	3.3646	0.0060	-0.0948	**0.0052**
γ	-0.1200	-0.1265	0.0191	2.3725	0.0065	-0.1223	**0.0023**
δ	0.1300	0.1387	0.0220	3.7167	0.0087	0.1289	**0.0011**
ν	0.9800	0.9648	0.0091	2.3915	**0.0152**	0.9616	0.0184
AIC		**2959.52**				2970.62	
BIC		**2998.78**				3009.88	
DIC		2956.62				n.a.	
0.05quantile							
λ_1	0.0000	-0.0476	0.0380	3.0571	0.0476	0.0040	**0.0040**
λ_2	0.0200	0.0507	0.0221	2.3280	**0.0307**	0.0612	0.0412
ξ^L	0.1500	0.1995	0.1108	2.2283	0.0495	0.1046	**0.0454**
ξ^U	0.3500	0.2623	0.1117	2.4280	0.0877	0.3324	**0.0176**
ω	-0.1000	-0.1048	0.0180	4.4462	**0.0048**	-0.0949	0.0051
γ	-0.1200	-0.1283	0.0206	3.2226	0.0083	-0.1154	**0.0046**
δ	0.1300	0.1411	0.0218	3.7457	**0.0111**	0.1670	0.0370
ν	0.9800	0.9615	0.0102	3.0885	**0.0185**	0.9065	0.0735
AIC		**3041.28**				3066.29	
BIC		**3080.54**				3105.56	
DIC		3055.58				n.a.	
0.10quantile							
λ_1	0.0000	-0.0448	0.0356	2.3378	0.0448	0.0195	**0.0195**
λ_2	0.0200	0.0466	0.0202	2.2520	**0.0266**	0.0599	0.0399
ξ^L	0.1500	0.1722	0.0860	2.3734	**0.0222**	0.1246	0.0254
ξ^U	0.3500	0.2940	0.0838	2.5761	0.0560	0.3624	**0.0124**
ω	-0.1000	-0.1011	0.0181	5.6055	**0.0011**	-0.0866	0.0134
γ	-0.1200	-0.1313	0.0206	2.0793	0.0113	-0.1222	**0.0022**
δ	0.1300	0.1375	0.0223	6.3039	**0.0075**	0.2074	0.0774
ν	0.9800	0.9603	0.0104	3.3496	**0.0197**	0.8327	0.1473
AIC		**3091.61**				3141.54	
BIC		**3130.88**				3180.80	
DIC		3143.56				n.a.	

Note that n.a. = not available. Bold font indicates better results.

In summary, the proposed models with the parallel-GG estimation are superior in terms of accuracy for parameter values and model performance. In practice, the QMLE estimates may be used as the starting value for the MCMC to achieve faster convergence and less computational time. However, this is absolutely not necessary as the QMLE sometimes does not converge well. Though the asymmetric-GARCH-ICAPM with mixture innovation requires more computational time than that of the GED, however, their property in capturing the extreme event cannot be ignored and should be investigated further in the empirical study.

3.5 Empirical Study

Based on the simulation study results, the study then implemented the proposed GARCH models with the ICAPM structure and mixture innovation for the daily stock returns from various markets around the world between January 2009 and April 2020. The data were obtained from Yahoo Finance and other relevant sources for the US (S&P500), Hong Kong (HSI), Australia (ASX200), New Zealand (NZX50), UK (FTSE100), Japan (Nikkei225), and Thailand (SET). We chose the start date at the beginning of 2009 to avoid the effect of the global financial crisis in 2008. For the out-of-sample forecast evaluation purpose, the study split the dataset into the training set that starts from January 2009 to January 2020 and the test set is from February 2020 to April 2020. The test set represents the main aim to assess the forecasting performance of the exponential-type GARCH models during the COVID-19 pandemic, especially in the early period when stock markets around the world tumbled severely. Note that, for each market, the returns in the study are the daily demeaned log returns, $r_t = (ln\,(S_t/S_{t-1}) - \bar{r}_t) * 100$, where S_t is the stock market index at time t.

Table 3.3 shows summary statistics for the seven markets from both training and test sets. As expected and based on the Jarque-Bera test results, all stock returns are non-normally distributed except the HSI returns in the out-of-sample period. Most of them are the right-skewed distribution, which indicates the mean-adjusted returns are still positive, except the out-of-sample returns from the NZX50 and Nikkei225. This might be due to the prolonged effect of COVID-19 that made them negative returns. The high kurtosis reveals the fat-tailed property of the returns. All these properties can be confirmed by the time series and density plots in Figure 3.2 and 3.3. Hence, modelling these returns using the proposed GARCH families with a mixture innovation is justified.

Table 3.3: Descriptive Statistics of the Daily Returns from Seven Stock Markets

	S&P500	**HSI**	**ASX200**	**NZX50**	**FTSE100**	**Nikkei225**	**SET**
In-sample data: 01/01/2009-30/01/2020							
Min.	-6.94	-6.04	-4.39	-3.76	-6.22	-10.61	-5.86
Max.	6.79	7.13	3.56	2.69	5.01	7.39	5.71
Mean	-0.0011	-0.0017	0.0001	-0.0004	-0.001	-0.0007	-0.0023
Std.	1.02	1.25	0.91	0.59	0.99	1.34	1.02
Skewness	-0.35	-0.10	-0.31	-0.51	-0.25	-0.46	-0.31
Kurtosis	8.10	5.38	4.86	5.71	6.23	7.43	6.88
JB Test	3075***	649***	458***	1001***	1243***	2338***	1744***
Obs.	2788	2739	2851	2863	2799	2737	2705
Out-of-sample data: 01/02/2020-30/04/2020							
Min.	-12.60	-4.88	-9.82	-6.21	-11.17	-6.04	-11.18
Max.	9.13	5.03	7.15	13.43	9.01	7.96	7.90
Mean	-0.0145	-0.0046	0.0158	0.0264	-0.0146	0.0132	0.0157
Std.	3.91	1.98	3.21	2.47	3.02	2.61	3.20
Skewness	-0.36	-0.18	-0.54	1.89	-0.75	0.44	-1.15
Kurtosis	4.39	3.54	4.01	13.69	5.77	4.15	6.10
JB Test	6.36*	1.08	5.64*	396.57***	25.58***	5.25**	38.37***
Obs.	62	61	62	74	62	60	62

Note that JB Test is the test statistic from Jarque-Bera's normality test. *,**,*** indicate the statistical significance at 10%, 5% and 1%, respectively. Obs. = number of observations.

Figure 3.2: Time Series and Density Plots of the Daily Returns from Seven Stock Markets between 01/01/2009 and 31/01/2020 (In-sample Data Set)

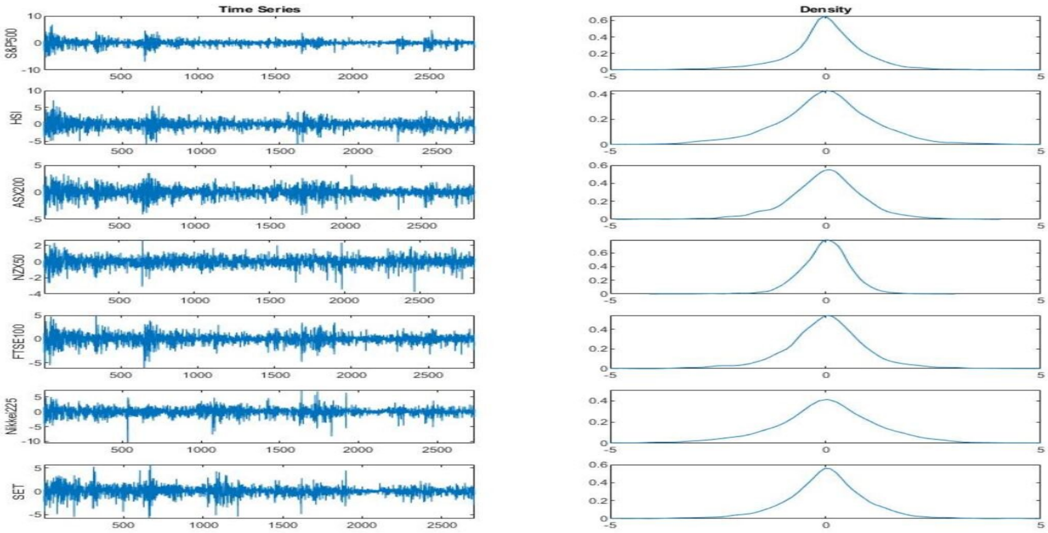
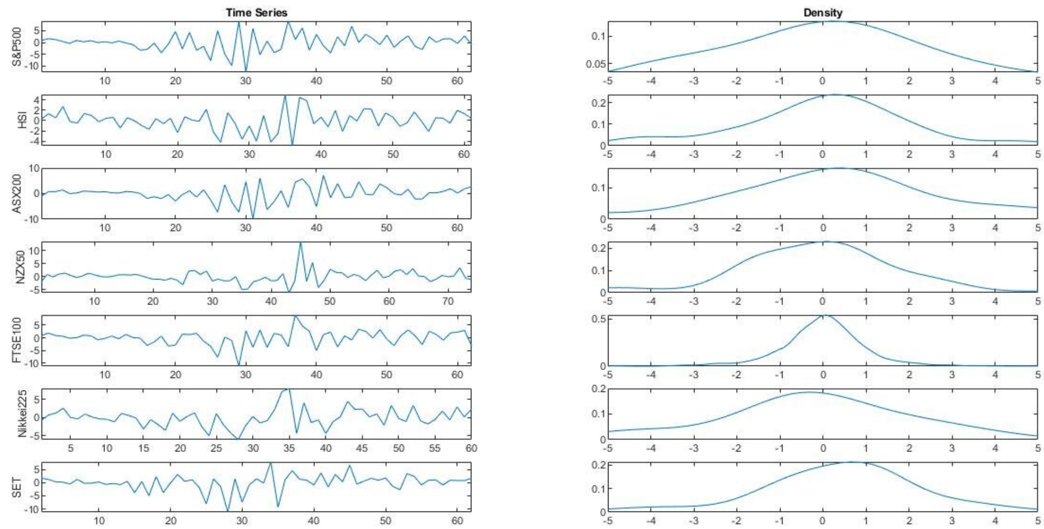


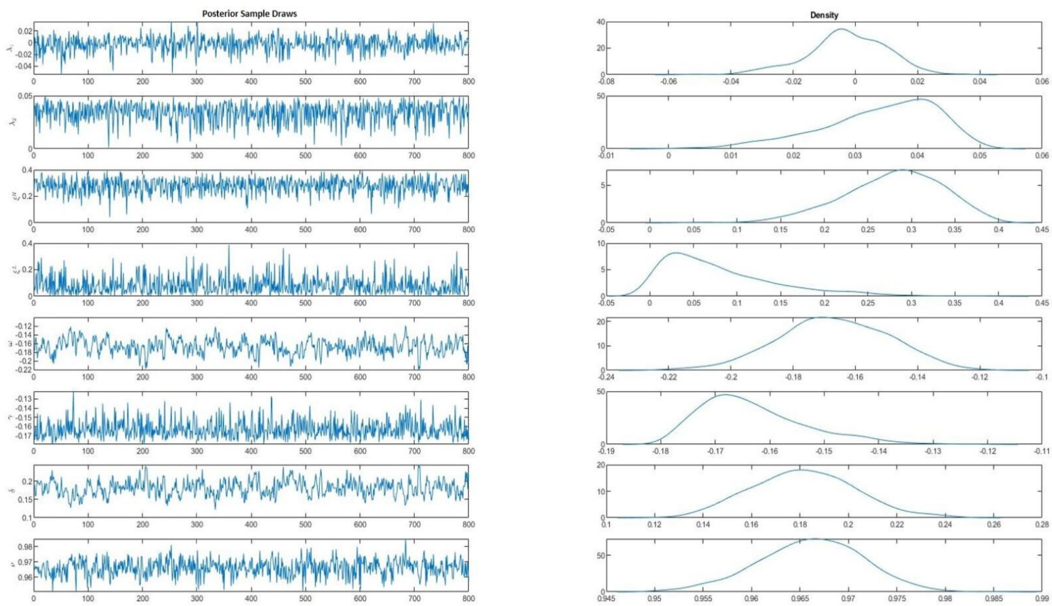
Figure 3.3: Time Series and Density Plots of the Daily Returns from Seven Stock Markets between 01/02/2020 and 30/04/2020 (Out-of-sample Data Set)



Regarding the estimation, there are slight modifications for the MCMC settings from the simulation study. In all cases, the study ran 10000 MCMC iterations, using 500 grid points, including 2000 burn-ins, and kept every tenth draw that yielded 800 posterior samples. Note that the prior distribution of all proposed models for each stock return can be found in Table A.2 and Table A.3 in Appendix.

Figure 3.4, as an example, shows the convergence draws and the density plots of posterior samples of the EGARCH(1,1,1)-ICAPM with mixture innovation process for the S&P500 returns. See additional empirical study results in the supplementary material. Overall, the convergence draws show that the parameter estimates reach a steady state while the density plots look reasonable.

Figure 3.4: Convergence Draws and Density Plots of Posterior Samples for EGARCH-ICAPM-Mixture Process from the S&P500 Returns



It can be clearly seen from Table 3.4 that most of the exponential-type-GARCH-ICAPM models, except EGARCH(1,1,1)-ICAPM with GED innovation, return the hypothetical value of zero for $\hat{\lambda}_1$, that confirms the assumption of risk-free rate under the ICAPM. Though the $\hat{\lambda}_1$ from EGARCH(1,1,1)-ICAPM with GED innovation model is significantly different from zero, its value is very tiny and its effect can be negligible. As expected, the parameter estimates of $\hat{\lambda}_2$, which is the risk premium, from all markets are positive and statistically significant. However, there are disagreements on $\hat{\lambda}_2$ across markets. The returns for the S&P500 and NZX50 each have a higher risk premium from two different models including the EGARCH(1,1,1)-ICAPM with mixture innovation and the GJR-GARCH(1,1)-ICAPM with mixture innovation for the former, and the GARCH(1,1)-ICAPM with mixture innovation and the GJR-GARCH(1,1)-ICAPM with mixture innovation for the latter while the FTSE100 and N225 have the higher risk premium from the GARCH(1,1)-ICAPM with mixture innovation and the GJR-GARCH(1,1)-ICAPM with mixture innovation, respectively. Since the higher risk premium indicates a higher risk-based return, that means the S&P500 and NZX50 fare better than the other markets between January 2009 and January 2020.

Table 3.4: In-sample Parameter Estimates Across Models and Markets using Parallel-Griddy-Gibbs Method

Model	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
S&P500																										
	λ_1		λ_2		ξ^L		ξ^R		ξ		ω		γ_1		δ_1		α_1^+		α_1^-		α_1		γ_1		ν_1	
EGARCH(1,1,1)-ICAPM-Mixture	-0.003	0.0125	0.0346***	0.0089	0.0807	0.0686	0.2786***	0.0579	-	-	-0.1668***	0.0176	-0.1632***	0.0095	0.1813***	0.0213	-	-	-	-	-	-	-	-	0.9662***	0.0053
LogGARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0366***	0.0020	0.0675***	0.0080	0.3030***	0.0370	-	-	0.0383***	0.0018	-	-	-	-	0.0049**	0.0020	0.0597***	0.0013	-	-	-	-	0.9589***	0.0004
GJR-GARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0734***	0.0057	0.0705***	0.0084	0.2847***	0.0380	-	-	0.0080***	0.0007	-	-	-	-	-	-	-	-	0.0106***	0.0026	0.0314***	0.0025	0.9467***	0.0005
GARCH(1,1)-ICAPM-Mixture	-0.0051	0.0087	0.0229*	0.0128	0.2998***	0.0831	0.3560***	0.0353	-	-	0.0181***	0.0050	-	-	-	-	-	-	-	-	0.1475***	0.0197	-	-	0.8450***	0.0175
EGARCH(1,1,1)-ICAPM-GED	0.0002*	0.0001	0.0198***	0.0019	-	-	-	-	1.7341***	0.0524	-0.1684***	0.0004	-	-	0.1805***	0.0001	-	-	-	-	-	-	-0.1222***	0.0104	0.9608***	0.0003
HSI																										
EGARCH(1,1,1)-ICAPM-Mixture	0.005	0.0271	0.0248**	0.0116	0.0867	0.0688	0.2552***	0.0697	-	-	-0.0898***	0.0124	-0.0520***	0.011	0.1152***	0.0161	-	-	-	-	-	-	-	-	0.9825***	0.0045
LogGARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0299***	0.0017	0.0682***	0.0081	0.3537***	0.0284	-	-	0.0402***	0.0019	-	-	-	-	0.0042*	0.0022	0.0568***	0.0030	-	-	-	-	0.9609***	0.0018
GJR-GARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0369***	0.0032	0.0693***	0.0081	0.2606***	0.0379	-	-	0.0088***	0.0006	-	-	-	-	-	-	-	-	0.0087***	0.0022	0.0373***	0.0026	0.9488***	0.0020
GARCH(1,1)-ICAPM-Mixture	0.0009	0.0092	0.0285**	0.0136	0.0986	0.0711	0.3069***	0.0449	-	-	0.0171***	0.0059	-	-	-	-	-	-	-	-	0.0594***	0.0084	-	-	0.9270***	0.0103
EGARCH(1,1,1)-ICAPM-GED	0.0002*	0.0001	0.0197***	0.0020	-	-	-	-	1.2093***	0.0418	-0.0514***	0.0008	-	-	-0.0242	0.0155	-	-	-	-	-	-	0.1082***	0.0001	0.9320***	0.0003
ASX200																										
EGARCH(1,1,1)-ICAPM-Mixture	0.0025	0.0151	0.0306***	0.0105	0.0458	0.0382	0.2495***	0.0595	-	-	-0.1062***	0.0136	-0.0901***	0.0124	0.1177***	0.0159	-	-	-	-	-	-	-	-	0.9812***	0.0044
LogGARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0260***	0.0027	0.0693***	0.0083	0.3684***	0.0256	-	-	0.0349***	0.0018	-	-	-	-	0.0018	0.0011	0.0584***	0.0027	-	-	-	-	0.9608***	0.0019
GJR-GARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0548***	0.0038	0.0692***	0.0076	0.2579***	0.0381	-	-	0.0083***	0.0007	-	-	-	-	-	-	-	-	0.0088***	0.0022	0.0341***	0.0023	0.9473***	0.0009
GARCH(1,1)-ICAPM-Mixture	-0.0032	0.0088	0.0523**	0.0262	0.0433	0.0377	0.2879***	0.0456	-	-	0.0108***	0.0029	-	-	-	-	-	-	-	-	0.0758***	0.0116	-	-	0.9080***	0.0128
EGARCH(1,1,1)-ICAPM-GED	0.0002**	0.0001	0.0225***	0.0021	-	-	-	-	1.5872***	0.0618	-0.1030***	0.0002	-	-	-0.0408***	0.0102	-	-	-	-	-	-	0.1235***	0.0001	0.9749***	0.0029
FTSE100																										
EGARCH(1,1,1)-ICAPM-Mixture	-0.0067	0.0183	0.0346***	0.0091	0.0471	0.0404	0.2665***	0.0609	-	-	-0.1454***	0.0176	-0.1402***	0.0148	0.1617***	0.0205	-	-	-	-	-	-	-	-	0.9689***	0.0065
LogGARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0175***	0.0067	0.0683***	0.0082	0.3350***	0.0307	-	-	0.0288***	0.0017	-	-	-	-	0.0018	0.0011	0.0487***	0.0024	-	-	-	-	0.9614***	0.0023
GJR-GARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0313***	0.0038	0.0697***	0.0081	0.2611***	0.0383	-	-	0.0085***	0.0007	-	-	-	-	-	-	-	-	0.0104***	0.0027	0.0342***	0.0024	0.9471***	0.0008
GARCH(1,1)-ICAPM-Mixture	-0.0056	0.0100	0.0586**	0.0250	0.0542	0.0471	0.3051***	0.0495	-	-	0.0196***	0.0060	-	-	-	-	-	-	-	-	0.1096***	0.0160	-	-	0.8666***	0.0189
EGARCH(1,1,1)-ICAPM-GED	0.0002*	0.0001	0.0135***	0.0015	-	-	-	-	1.4982***	0.0551	-0.1204***	0.0002	-	-	-0.0705	0.0105	-	-	-	-	-	-	0.1498***	0.0001	0.9749***	0.0029
N225																										
EGARCH(1,1,1)-ICAPM-Mixture	-0.0139	0.0223	0.0315***	0.0103	0.1229*	0.0748	0.2946***	0.0589	-	-	-0.1521***	0.0189	-0.1376***	0.0177	0.2200***	0.0252	-	-	-	-	-	-	-	-	0.9301***	0.0122
LogGARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0230***	0.0047	0.0688***	0.0082	0.3542***	0.0292	-	-	0.0437***	0.0016	-	-	-	-	0.0045**	0.0021	0.0557***	0.0025	-	-	-	-	0.9597***	0.0011
GJR-GARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0316***	0.0031	0.0707***	0.0093	0.2533***	0.0453	-	-	0.0097***	0.0007	-	-	-	-	-	-	-	-	0.0087***	0.0017	0.0373***	0.0021	0.9474***	0.0010
GARCH(1,1)-ICAPM-Mixture	-0.0045	0.0089	0.0287**	0.0126	0.1195	0.0786	0.3349***	0.0424	-	-	0.0644***	0.0147	-	-	-	-	-	-	-	-	0.1283***	0.0180	-	-	0.8340***	0.0209
EGARCH(1,1,1)-ICAPM-GED	0.0002*	0.0001	0.0254***	0.0021	-	-	-	-	1.5815***	0.0523	-0.0975***	0.0023	-	-	-0.0778***	0.0117	-	-	-	-	-	-	0.1499***	0.0001	0.9320***	0.0004
SET																										
EGARCH(1,1,1)-ICAPM-Mixture	-0.0073	0.0172	0.0292***	0.0113	0.0649	0.0563	0.2463***	0.0638	-	-	-0.1654***	0.0182	-0.0565***	0.0131	0.1959***	0.0221	-	-	-	-	-	-	-	-	0.9762***	0.0057
LogGARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0461***	0.0024	0.0683***	0.0081	0.2955***	0.0395	-	-	0.0373***	0.0018	-	-	-	-	0.0044**	0.0020	0.0574***	0.0030	-	-	-	-	0.9591***	0.0007
GJR-GARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0352***	0.0033	0.0698***	0.0085	0.2783***	0.0340	-	-	0.0081***	0.0008	-	-	-	-	-	-	-	-	0.0084***	0.0021	0.0342***	0.0022	0.9469***	0.0007
GARCH(1,1)-ICAPM-Mixture	-0.0032	0.0099	0.0429**	0.0198	0.0678	0.0562	0.3149***	0.0389	-	-	0.0090***	0.0025	-	-	-	-	-	-	-	-	0.0904***	0.0122	-	-	0.8967***	0.0116
EGARCH(1,1,1)-ICAPM-GED	0.0002**	0.0001	0.0195***	0.0019	-	-	-	-	1.7098***	0.0416	-0.1685***	0.0003	-	-	-0.0417***	0.0099	-	-	-	-	-	-	0.2036***	0.0001	0.9749***	0.0030
NZX50																										
EGARCH(1,1,1)-ICAPM-Mixture	0.0091	0.01	0.0279**	0.0114	0.0433	0.0352	0.2367***	0.0629	-	-	-0.1486***	0.0337	-0.0327***	0.0112	0.1233***	0.0245	-	-	-	-	-	-	-	-	0.9598***	0.0143
LogGARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0496***	0.0038	0.0700***	0.0084	0.3370***	0.0327	-	-	0.0264***	0.0019	-	-	-	-	0.0018	0.0011	0.0561***	0.0024	-	-	-	-	0.9602***	0.0015
GJR-GARCH(1,1)-ICAPM-Mixture	0.0000	0.0001	0.0536***	0.0043	0.0691***	0.0079	0.2489***	0.0377	-	-	0.0080***	0.0008	-	-	-	-	-	-	-	-	0.0072***	0.0021	0.0239***	0.0025	0.9469***	0.0007
GARCH(1,1)-ICAPM-Mixture	-0.0033	0.0093	0.0957**	0.0464	0.0473	0.0357	0.2627***	0.0509	-	-	0.0194**	0.0078	-	-	-	-	-	-	-	-	0.0734***	0.0158	-	-	0.8614***	0.0375
EGARCH(1,1,1)-ICAPM-GED	0.0002*	0.0001	0.0158***	0.0018	-	-	-	-	1.5939***	0.0598	-0.1204***	0.0002	-	-	-0.0178*	0.0093	-	-	-	-	-	-	0.1235***	0.0001	0.9751***	0.0029

Note that *,**,*** indicate the statistical significance at 10%, 5% and 1%, respectively.

All GARCH-ICAPM models from all markets return the very high volatility persistence, assessed by $\hat{\nu}$ for the GARCH and EGARCH, $\hat{\nu} + (\hat{\alpha}^+ + \hat{\alpha}^-)/2$ for the LogGARCH, and $\hat{\alpha} + \hat{\nu} + \hat{\gamma}/2$ for the GJR-GARCH. This indicates the longer-lasting effect of the volatility. That means if the volatility happens, it is likely to continue for a couple of days. Based on the $|\hat{\gamma}| > 0$ for the GJR-GARCH, $\hat{\gamma} < 0$ for the EGARCH, $\alpha^+ < \alpha^-$ for the LogGARCH, and $\hat{\xi}^L < \hat{\xi}^R$ from the GPD, there is a strong leverage effect in all stock markets. This implies the scale of the negative news/shocks is larger than the positive news/shocks, while the estimated constant term in the GARCH family, $\hat{\omega}$, is positive for all series except the EGARCH. This is probably due to the fact that the constant term represents the stability level of unconditional variance that differs in each model.

In summary, the study finds that all stock markets have a very high volatility persistence effect. Moreover, the (asymmetric-)GARCH-ICAPM family clearly shows a higher persistence effect than the classical GARCH-ICAPM model. There is also strong statistical evidence that the returns from all markets are asymmetrically and non-normally distributed, which is a confirmed fact of financial markets. Precisely, financial stock returns are volatility clustering, heavy-tailed, and asymmetrically distributed, see more details in Caporale and Zekokh (2019), Engle (2004), Fan and Patton (2014), and Patton (2012). This justifies the use of the exponential-type-GARCH-ICAPM with a mixture innovation process here in this study.

To compare across models, the study calculates the marginal (log-)likelihood given in Chib (1995), the root mean squared error (RMSE), and the mean absolute percentage error (MAPE) for the in-sample fit, while the study uses the RMSE, MAPE, and the two-sample Kolmogorov-Smirnov (KS) test for the out-of-sample forecasting performance. Note that the KS test is used to assess whether the forecast distribution is close to the empirical one. The higher the p-value, the closer the forecast distribution is.

The results from Table 3.5 show that the LogGARCH(1,1)-ICAPM-Mixture is the best performing model in terms of marginal log-likelihood for the training period in all markets while the EGARCH(1,1,1) with either the mixture or the GED innovation is the best in terms of the forecast errors (RMSE and MAPE) under the same situation, except the HSI and FTSE100. For the COVID-19 forecasting period, the EGARCH(1,1,1)-ICAPM is still the one that returns the best out-of-sample forecasts either in terms of forecast errors or forecast distribution. Precisely, the EGARCH(1,1,1)-ICAPM with the GED innovation is slightly better as measured by the RMSE and MAPE while that with the mixture innovation is marginally better in terms of the KS test. Hence, the study can conclude that the EGARCH-ICAPM model is the main driving model in the early period of COVID-19. This is probably caused by the news impact function, which is the main feature of the model that can capture asymmetric shocks. These findings also confirm the superiority of including the ICAPM structure to the proposed model in this study.

As in the simulation study, the study also assesses the proposed models across markets and quantiles. For example, Table 3.6 shows the results from the in-sample parameter estimation of EGARCH(1,1,1)-ICAPM with mixture innovation in all markets. In terms of the marginal log-likelihood, the study finds the models with the lowest mixing probability or quantile of 0.01 are the best in capturing the extreme returns, and hence the volatility.

Table 3.5: Model Comparison and Performance Across Markets

	Mrgnl. Lglik.	In-Sample		Out-of-Sample		KSTEST
		RMSE	MAPE	RMSE	MAPE	P value
S&P500						
EGARCH-ICAPM-Mixture	22460	1.43	6.64	4.13	18.17	**0.08*****
LogGARCH-ICAPM-Mixture	**23938**	1.41	6.77	4.00	10.47	0.05***
GJR-GARCH-ICAPM-Mixture	23451	1.35	6.45	3.99	10.44	0.04***
GARCH-ICAPM-Mixture	19682	1.64	7.78	4.04	14.18	0.06***
EGARCH-ICAPM-GED	4500	**1.29**	**5.81**	3.97	10.06	0.02***
EGARCH-GED	2943	1.30	5.98	**3.96**	**9.68**	0.02***
EGARCH-Mixture	16066	1.48	7.25	4.12	17.23	0.03***
HSI						
EGARCH-ICAPM-Mixture	27874	1.77	7.53	**2.09**	**1.65**	0.34***
LogGARCH-ICAPM-Mixture	**28321**	**1.73**	**6.83**	2.24	1.99	0.21***
GJR-GARCH-ICAPM-Mixture	28139	2.06	8.58	2.43	2.42	0.19***
GARCH-ICAPM-Mixture	24016	2.05	8.66	2.54	2.74	0.22***
EGARCH-ICAPM-GED	16531	1.77	7.65	2.30	2.31	**0.54*****
EGARCH-GED	10982	1.78	7.61	2.34	2.27	0.47***
EGARCH-Mixture	19920	1.81	7.56	2.14	1.74	0.18***
ASX200						
EGARCH-ICAPM-Mixture	22657	1.29	15.89	**3.19**	**2.44**	0.27***
LogGARCH-ICAPM-Mixture	**23419**	1.32	14.60	3.30	5.06	0.10***
GJR-GARCH-ICAPM-Mixture	23167	1.35	15.21	3.34	5.44	0.23***
GARCH-ICAPM-Mixture	19680	1.49	15.82	3.40	5.96	0.18***
EGARCH-ICAPM-GED	5804	**1.28**	**14.12**	3.30	5.43	**0.30*****
EGARCH-GED	3808	1.30	16.85	3.31	5.15	0.19***
EGARCH-Mixture	16197	1.35	16.66	3.20	2.54	0.16***
FTSE100						
EGARCH-ICAPM-Mixture	23594	1.43	12.72	3.21	1.96	**0.34*****
LogGARCH-ICAPM-Mixture	**24669**	**1.38**	**12.56**	3.15	1.67	0.18***
GJR-GARCH-ICAPM-Mixture	24253	1.44	13.57	3.18	1.75	0.22***
GARCH-ICAPM-Mixture	20607	1.60	15.27	3.26	1.89	0.26***
EGARCH-ICAPM-GED	7433	1.39	13.71	**3.14**	**1.70**	0.33***
EGARCH-GED	4905	1.41	13.05	3.16	1.73	0.26***
EGARCH-Mixture	16872	1.49	14.40	3.22	1.92	0.18***
N225						
EGARCH-ICAPM-Mixture	28909	1.92	7.26	2.87	2.071	0.25***
LogGARCH-ICAPM-Mixture	**29597**	1.95	7.65	2.92	2.37	0.18***
GJR-GARCH-ICAPM-Mixture	29497	1.92	7.24	2.91	2.30	0.19***
GARCH-ICAPM-Mixture	25064	2.11	7.92	2.92	2.20	0.21***
EGARCH-ICAPM-GED	16785	1.79	6.80	**2.82**	**2.067**	**0.31*****
EGARCH-GED	11032	**1.78**	**6.75**	2.87	2.10	0.23***
EGARCH-Mixture	20661	1.98	7.66	2.93	2.18	0.10***
SET						
EGARCH-ICAPM-Mixture	22895	1.46	7.38	**3.27**	**1.42**	**0.28*****
LogGARCH-ICAPM-Mixture	**23422**	1.44	7.16	3.35	1.57	0.27***
GJR-GARCH-ICAPM-Mixture	23341	1.41	6.91	3.33	1.50	0.21***
GARCH-ICAPM-Mixture	19726	1.55	7.59	4.07	2.03	0.22***
EGARCH-ICAPM-GED	6494	**1.37**	**6.85**	3.29	1.48	0.24***
EGARCH-GED	4305	1.38	6.87	3.32	1.46	0.16***
EGARCH-Mixture	16358	1.52	7.66	3.38	1.48	0.15***
NZX50						
EGARCH-ICAPM-Mixture	15259	0.844	8.74	2.52	5.21	**0.32*****
LogGARCH-ICAPM-Mixture	**15878**	0.91	9.31	2.54	5.92	0.04***
GJR-GARCH-ICAPM-Mixture	15565	0.88	9.18	2.55	6.03	0.10***
GARCH-ICAPM-Mixture	13177	0.86	8.90	2.54	5.78	0.30***
EGARCH-ICAPM-GED	-6150	0.86	8.94	2.54	5.86	0.09***
EGARCH-GED	-4073	0.87	9.03	2.54	5.89	0.08***
EGARCH-Mixture	10918	**0.843**	**8.58**	**2.51**	**5.04**	0.09***

Note that Mrgnl. Lglik. is Marginal Log-Likelihood. *** indicates the statistical significance at 1%. Bold font indicates better results.

Table 3.6: In-sample Parameter Estimates of EGARCH-ICAPM-Mixture Model at Lower Quantiles in All Markets

Quantile		λ_1	λ_2	ξ^L	ξ^R	ω	γ	δ	ν	**Mrgnl. Lglik.**
					S&P500					
0.01	Mean	-0.0155	0.0330	0.0790	0.3735	-0.1680	-0.1625	0.1932	0.9635	**22788**
	Std.	0.0167	0.0111	0.0835	0.1186	0.0183	0.0119	0.0230	0.0050	
0.05	Mean	-0.0047	0.0337	0.0626	0.3596	-0.1644	-0.1627	0.1821	0.9659	22568
	Std.	0.0125	0.0094	0.0583	0.0657	0.0159	0.0095	0.0192	0.0049	
0.10	Mean	-0.0030	0.0346	0.0807	0.2786	-0.1668	-0.1632	0.1813	0.9662	22460
	Std.	0.0125	0.0089	0.0686	0.0579	0.0176	0.0095	0.0213	0.0053	
					HSI					
0.01	Mean	-0.0274	0.0262	0.1389	0.2036	-0.0902	-0.0546	0.1210	0.9797	**28042**
	Std.	0.0241	0.0113	0.1104	0.1237	0.0124	0.0104	0.0167	0.0047	
0.05	Mean	-0.0069	0.0250	0.1085	0.2968	-0.0901	-0.0524	0.1183	0.9816	27923
	Std.	0.0275	0.0114	0.0921	0.0930	0.0131	0.0105	0.0176	0.0046	
0.10	Mean	0.0050	0.0248	0.0867	0.2552	-0.0898	-0.0520	0.1152	0.9825	27874
	Std.	0.0271	0.0116	0.0688	0.0697	0.0124	0.0110	0.0161	0.0045	
					ASX200					
0.01	Mean	-0.0188	0.0293	0.1257	0.3141	-0.1011	-0.0941	0.1168	0.9804	**22710**
	Std.	0.0158	0.0110	0.0986	0.1245	0.0129	0.0113	0.0152	0.0044	
0.05	Mean	-0.0044	0.0294	0.0722	0.3188	-0.1038	-0.0913	0.1176	0.9811	22650
	Std.	0.0142	0.0105	0.0625	0.0739	0.0133	0.0119	0.0158	0.0042	
0.10	Mean	0.0025	0.0306	0.0458	0.2495	-0.1062	-0.0901	0.1177	0.9812	22657
	Std.	0.0151	0.0105	0.0382	0.0595	0.0136	0.0124	0.0159	0.0044	
					FTSE100					
0.01	Mean	-0.0358	0.0339	0.1108	0.3110	-0.1455	-0.1406	0.1716	0.9644	**23724**
	SD	0.0166	0.0092	0.0975	0.1260	0.0176	0.0139	0.0211	0.0064	
0.05	Mean	-0.0177	0.0338	0.0609	0.3216	-0.1395	-0.1387	0.1594	0.9689	23623
	SD	0.0176	0.0094	0.0547	0.0715	0.0169	0.0141	0.0200	0.0065	
0.10	Mean	-0.0067	0.0346	0.0471	0.2665	-0.1454	-0.1402	0.1617	0.9689	23594
	SD	0.0183	0.0091	0.0404	0.0609	0.0176	0.0148	0.0205	0.0065	
					N225					
0.01	Mean	-0.0283	0.0282	0.1283	0.3681	-0.1462	-0.1365	0.2193	0.9293	**29092**
	Std.	0.0251	0.0111	0.0890	0.1251	0.0167	0.0159	0.0231	0.0109	
0.05	Mean	-0.0185	0.0305	0.1302	0.3778	-0.1466	-0.1362	0.2167	0.9302	28955
	Std.	0.0228	0.0110	0.0900	0.0633	0.0179	0.0173	0.0243	0.0117	
0.10	Mean	-0.0139	0.0315	0.1229	0.2946	-0.1521	-0.1376	0.2200	0.9301	28909
	Std.	0.0223	0.0103	0.0748	0.0589	0.0189	0.0177	0.0252	0.0122	
					SET					
0.01	Mean	-0.0254	0.0268	0.0952	0.2176	-0.1623	-0.0585	0.1995	0.9769	**23089**
	Std.	0.0167	0.0111	0.0835	0.1186	0.0183	0.0119	0.0230	0.0050	
0.05	Mean	-0.0154	0.0287	0.0743	0.3008	-0.1625	-0.0569	0.1961	0.9763	22949
	Std.	0.0168	0.0112	0.0693	0.0766	0.0182	0.0125	0.0223	0.0054	
0.10	Mean	-0.0073	0.0292	0.0649	0.2463	-0.1654	-0.0565	0.1959	0.9762	22895
	Std.	0.0172	0.0113	0.0563	0.0638	0.0182	0.0131	0.0221	0.0057	
					NZ50					
0.01	Mean	-0.0045	0.0268	0.0838	0.3015	-0.1492	-0.0370	0.1282	0.9583	**15397**
	Std.	0.0115	0.0118	0.0690	0.1310	0.0298	0.0104	0.0218	0.0135	
0.05	Mean	0.0043	0.0272	0.0547	0.2955	-0.1504	-0.0348	0.1257	0.9577	15286
	Std.	0.0099	0.0116	0.0458	0.0842	0.0338	0.0114	0.0234	0.0157	
0.10	Mean	0.0091	0.0279	0.0433	0.2367	-0.1486	-0.0327	0.1233	0.9598	15259
	Std.	0.0100	0.0114	0.0352	0.0629	0.0337	0.0112	0.0245	0.0143	

Note that Mrgnl. Lglik. is Marginal Log-Likelihood and the bold font indicates better results.

3.6 Conclusions

During the early period of COVID-19 pandemic (the first quarter of 2020), the plummets of stock markets around the world indicated the unexpected risk from the unforeseeable future. In this study, the study analyses the returns from seven stock markets including the US (S&P500), Hong Kong (HSI), Australia (ASX200), UK (FTSE100), Japan (N225), Thailand (SET), and New Zealand (NZX50). The model the study used in the analysis is the ICAPM together with the asymmetric-GARCH processes, including standard GARCH, exponential GARCH, log GARCH, and GJR-GARCH, where their innovation is either the mixture of GPD and Gaussian or the GED. Due to the complexity in estimating the parameters of the GPD and the GED, the study uses the parallel computing for GG algorithm, which is an MCMC method, to facilitate the estimation. Our simulation study shows that the parallel-GG outperforms the one-step QMLE in terms of the mean absolute deviation. We then proceed to the empirical study using the parallel GG to estimate all (asymmetric-)GARCH-ICAPM models. We found in the training (in-sample) period of January 2009 to January 2020 that the LogGARCH(1,1)-ICAPM with mixture innovation and the EGARCH(1,1,1)-ICAPM with either mixture or GED innovation are the best performing models across seven stock markets. For the forecasting (early COVID-19) performance, the study found that both of EGARCH(1,1,1)-ICAPM with mixtures and GED innovations are the models that return the lowest forecast errors and the closest forecast distribution (to the real data). The outstanding forecasting performance of EGARCH models is possibly caused by the news impact function that can capture the asymmetric shocks existing during the early period of the COVID-19 pandemic.

Chapter 4

Variational Bayes with Latent Variables and Data Augmentation for High Dimensional Mixed Regular Vine Dependence Structure and Mixture Innovation Model in Financial Time Series

4.1 Abstract

This paper introduces Bayesian inference and machine learning to multivariate regular vine copula dependence models, R-vines, in graph theory for high dimensional financial time series. In high dimensional data, it has been accepted that a variable pair belongs to an individual characteristic and also has a unique path dependence. Research in the past has mainly focused on standard multivariate copula as well as two specific sub-classes of vine copula (canonical vine copula and drawable vine copula) and cannot address the fact situation stated above. R-vines may be able to solve this problem with a massive number of tree structure possibilities; however, such work is rare in the literature and requires highly intensive computation. In this study, a regular vine pair structure is used to consider a range of bivariate copula functions, with a total of thirteen candidates evaluated. We also propose mixture distribution incorporated with Gaussian distribution and two generalised Pareto distributions in Extreme Value Theory (EVT) in the R-vine dependence model. The exponential-type generalised autoregressive conditional heteroskedasticity model (EGARCH) and the intertemporal capital asset pricing model (ICAPM) are chosen for the univariate marginal model. Therefore, a variational Bayes method (with data augmentation) and regular vine copula with EGARCH-ICAPM and a mixture distribution model have been proposed, the so-called VBDA-R-vine-EGARCH-ICAPM-Mixture model.

Simulation experiments prove that the proposed model is justified, and it outperforms various competing models. We then proceed to empirical experiments with high dimensional data and up to 100 dimensions. These confirm that the proposed model may be one of the best. Finally, the paper includes full R-vine specifications simply presented in matrix representations. Furthermore, this paper provides the full algorithm of the proposed model including the parallel R-vine with intractable margins model, data generation and bivariate copula selection in R-vine.

Keywords: *High Dimensional Data, Regular Vine Copulae, Bayesian Inference and Machine Learning, Exponential-Type Generalised Autoregressive Conditional Heteroskedasticity Model, Intertemporal Capital Asset Pricing Model, Mixture Distribution.*

4.2 Introduction

In the past few decades, since Sklar (1959) showed that a copula function can describe non-linear comovement between variables, the study of copula dependence models has been rapidly growing. A prominent and useful feature of the copula, as a multivariate distribution, is its ability to join different marginal distributions. Hence, copulae have been used widely in many application areas, in recent times: in hydrology (Tao, Wang, Wang, Ni & Wu, 2021); energy (Schinke-Nendza et al., 2021); agricultural and forest meteorology (Chatrabgoun et al., 2020); oceanology (Aghatise, Khan & Ahmed, 2021); computer science (Sharma, Hazra & Sekharan, 2021); insurance (R. Zhou & Ji, 2021); and more widely in economics, econometrics, and finance (Bladt & McNeil, 2022).

Vast literature shows copula models are well-known for modeling dependence in quantitative risk management because of the tail asymmetry property. Modeling of tails is vital because it is a most common risk measure such that VaR (Value-at-Risk) and CVaR (Conditional Value-at-Risk/Expected shortfall). See the risk measure definition in McNeil et al. (2015). The standard multivariate copula is one of the most popular dependence models. It consists of two copula families: one, a family of elliptical copula - for a recent study, see Wichitaksorn et al. (2019); and, two, a family of Archimedean copula - for a recent study, see Górecki, Hofert and Okhrin (2021). However, a standard multivariate copula has some restrictions. In particular, it has a flexible restriction of only single parameter to control all pairs of tail comovement and single parameter is even worse especially once the number of dimensions is high. This restriction was noticed by Aas, Czado, Frigessi and Bakken (2009). They proposed a new class of copula the so-called R-vine (Regular vine copula). Additionally, there are two special sub-classes of vine copula including the C-vine (Canonical vine copula) and the D-vine (Drawable vine copula) (Kurowicka & Cooke, 2005). A further discussion of vine copulae is presented in Section 4.3, but see also Joe (2015, Chapter 3). In the literature, studies of vine copulae are still limited, especially R-vines. One of the important reasons for this is the complexity of the vine copula and the fact it requires very intensive computation. In particular, the R-vine has a much larger of number of possible tree structures to choose from than the C-vine and the D-vine, especially, again, when the number of dimensions is high. However, R-vine is a very flexible dependence model as, again for the same reason noted above, there is a massive number of tree structure possibilities. Moreover, it relaxes the parameter restriction of standard multivariate copula which would be beneficial for financial tail behavior analysis in high dimensions.

There are very few studies in the financial literature on R-vines with high dimensions. For instance, Müller and Czado (2018) studied up to 398 dimensions on S&P500 constituents. Again, Müller and Czado (2019) studied 2,131 dimensions, while other R-vine studies examine the lower dimensions (less than 25). See, among others, the study of R-vines in lower dimensions in Brechmann, Czado and Aas (2012), Maya, Gomez-Gonzalez and Velandia (2015) and H. Li, Liu and Wang (2022) . For a recent survey on copula models, see, among others, Fan and Patton (2014), Patton (2012), Czado (2010) and Joe et al. (2010). Indeed, the study of R-vine in high dimensions is still scarce and it needs more investigation. This observation is the starting point of the present paper, which is a study of R-vines. Note that the closest model to the paper is in the study of Dißmann, Brechmann, Czado and Kurowicka (2013).

Dißmann et al. (2013) study R-vines in application to 16-dimensional financial indices using the quasi maximum likelihood estimation method (QMLE). Given standardised residuals, vine copula models are applied among five different classes including mixed R-vines, mixed C-vines, mixed D-vines, all student-t R-vines and standard multivariate Gaussian copula. There are seven bivariate copula candidates in the study of Dißmann et al. (2013). Given the Vuong (1989) test and likelihood value, the best vine copulae are mixed R-vines. With this, this thesis follows with study of the best vine copulae by an extension from 16 dimensions to up to high 100 dimensional financial data using S&P500 constituents for the first contribution. The second contribution of this is to extend seven bivariate copulae to 13 bivariate copulae including 2-parameter BB copula families and more Archimedean copula families in order to cover more possible tail asymmetry behaviors of financial data. Further details are discussed in Section 4.5. The third contribution is in the marginal model. This paper proposes using ICAPM[1]-Asymmetric-GARCH with a mixture innovation for better volatility cluster filtration and more insightful risk premium information on individual financial securities. The mixture innovation is the distribution of a Gaussian and two heavy-tailed distributions while Dißmann et al. (2013) used the ARMA-GARCH model with only one Student-t innovation.

Another important contribution of this paper is an estimation method. In the literature, QMLE is widely used in copulae studies; see, among others, Sun, Fu, Liao and Xu (2020), Nagler, Bumann and Czado (2019) and B. Chang and Joe (2019). However, in general, the QMLE estimator relies on dependence initial parameters and has a convergence issue (Mantel & Myers, 1971). This leads to the fourth contribution. This paper hence proposes efficient Bayesian and machine learning estimation methods by a novel general R-vine dependence model where these algorithms are also compatible with two special sub-classes of vine copulae. Bayesian inference and machine learning need more computational time and complexity; to mitigate this, this thesis adopts a parallel computing technique for the proposed algorithms and the QMLE estimator to the Bayesian inference and machine learning approach as the initial condition in order to obtain a faster convergence outcome. Besides, two main Bayesian Markov chain Monte Carlo (MCMC) and machine learning algorithms are explored to the novel R-vine dependence model. Firstly, a random walk chain Metropolis-Hastings (rw-MH) sampling algorithm in Bayesian MCMC estimation method of Koop et al. (2007). Secondly, in Bayesian inference and machine learning, a variational Bayes with (and without) data augmentation and multivariate distribution (VBDA) of Loaiza-Maya and Smith (2019). Furthermore, this paper also investigates and then proposes an alternative efficient variational approximation so as to return an efficient VB estimator and minimising

[1]ICAPM is an intertemporal capital asset pricing model which was originated by Merton (1973).

computational time, which can be seen in simulation study in the Appendix B.1.2. See, among others, the relevant surveys of estimation methods including MLE and Bayesian inference and machine learning estimation method in Chang et al. (2018), Oh and Patton (2016), Pareek and Thakkar (2015), and Kolm et al. (2014).

The rest of this paper is organised as follows. Section 4.3 presents graph theory in high dimensional regular vine copula, which is a major contribution of this paper, including two special sub-classes in the term of the regular vine and also presents the marginal distribution and the mixture distribution concept. Followed by, all Bayesian inference and machine learning algorithms for general R-vine dependence models are set out in Section 4.4. Empirical experiments in high dimensional study present in Section 4.5. This chapter ends with conclusion and discussion in Section 4.6. Simulation experiments and the data generation algorithm can be found in Appendix B.1.

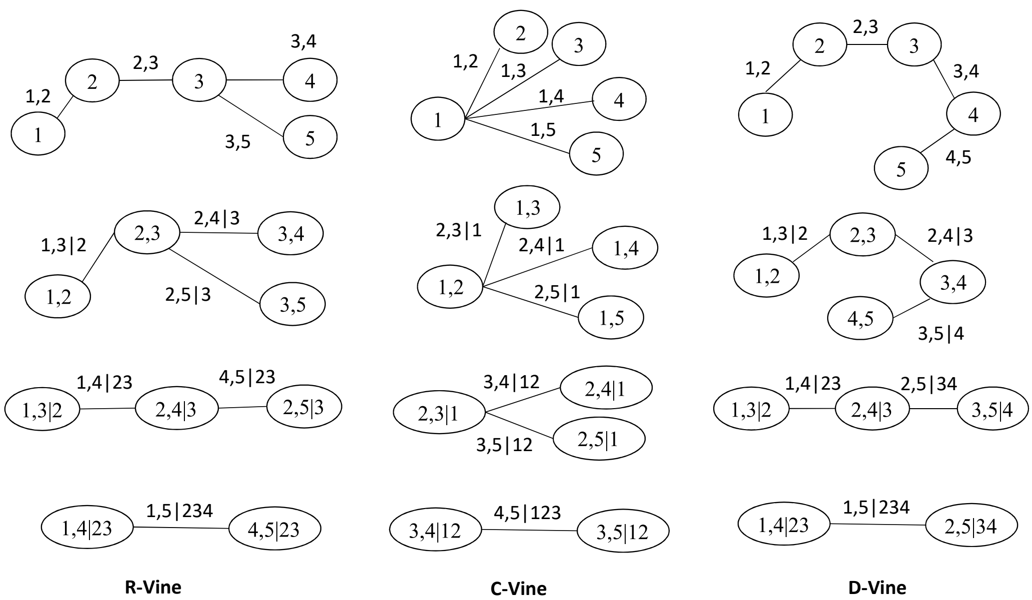
4.3 Graph Theory

4.3.1 Regular Vines in N-by-N Matrix Representation

This section briefly introduces with the necessary theoretical background of a regular vine (R-vine) and is also compatible with two special sub-classes, the canonical vine (C-vine) and drawable vine (D-vine) copulae in n-dimensions in the graphical model, where n is number of dimension. In graph theory, an R-vine is presented in a tree structure which consists of a set of nodes, N, connected by edges, E. A regular vine ($\mathcal{V}$) of n-dimensions is a nested set of $n-1$ trees. The first tree $T=1$ will have n nodes and $n-1$ pairs/edges. After that, the pair/edge in $T=1$ becomes the node on the second tree $T=2$ and $T=2$ has $n-1$ nodes and $n-2$ pairs/edges with the proximity condition that these nodes must share a common node in tree $T=1$. For the third tree $T=3$, there are $n-2$ nodes and $n-3$ pairs/edges with the proximity condition that nodes in $T=3$ must share a common node in tree $T=2$. Proceeding in this way until the last tree, T_{n-1} has two nodes and one edge with the proximity condition that they must share all remaining nodes. For more definitions and examples, such as a complete union, conditioning and conditional sets of an edge, see Dißmann et al. (2013). More theoretical details and properties of vine copulae can be found in Bedford and Cooke (2001), Bedford and Cooke (2002), Kurowicka and Joe (2010), and Joe (2015).

To develop an algorithm in a vine graphical model, the convenient way is that containing the trees in an n-by-n matrix. Dißmann et al. (2013) constructed a lower triangular matrix $\mathcal{M}$ to store vines as a tree structure in n dimensions, which is demonstrated in Figure 4.1.

Figure 4.1: Vine Copulae in Five Dimensions



Note that the graphical model demonstrates R-vine, C-vine and D-dine by column representation. By row, the figure begins with the first tree T_1 up to the fourth tree T_4.

Definition 4.3.1 (Matrix Constraint Set) *Let $\mathcal{M}$ be $(m_{i,k})$, where $i, k = 1, ..., n$, be a lower triangular n-by-n matrix. The i^{th} constraint set for $\mathcal{M}$ is*

$$\mathcal{CM}(i) = \{(\{m_{k,k}, m_{i,k}\}, \mathcal{D})|i = k + 1, ..., n, \mathcal{D} = \{m_{i+1,k}, ..., m_{n,k}\}\} \tag{4.3.1}$$

for $i = n - 1, ..., 1$. Otherwise, $\mathcal{D} = \varnothing$ (if $i = n$). Hence, $\mathcal{D}$ is the conditioning set and

$\{m_{k,k}, m_{i,k}\}$ *is the conditioned set.*

In other words, each pair variable within the constraint set is built up from a diagonal element $m_{k,k}$ plus another element in the same column $m_{i,k}$ conditioning on set of $\mathcal{D} = \{m_{i+1,k}, ..., m_{n,k}\}$. Otherwise, $\mathcal{D} = \varnothing$ when i equal to n. To express definition 4.3.1, the 5-by-5 matrix $\mathcal{M}$ of Figure 4.1 can be constructed as

$$\mathcal{M}^R = \begin{pmatrix} 1 & & & & \\ 5 & 5 & & & \\ 4 & 4 & 2 & & \\ 3 & 2 & 4 & 3 & \\ 2 & 3 & 3 & 4 & 4 \end{pmatrix}, \quad \mathcal{M}^C = \begin{pmatrix} 5 & & & & \\ 4 & 4 & & & \\ 3 & 3 & 3 & & \\ 2 & 2 & 2 & 2 & \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \mathcal{M}^D = \begin{pmatrix} 5 & & & & \\ 1 & 4 & & & \\ 2 & 1 & 3 & & \\ 3 & 2 & 1 & 2 & \\ 4 & 3 & 2 & 1 & 1 \end{pmatrix}$$

Once an n-by-n matrix, when $n = 5$, $\mathcal{M}$ can be constructed, it would be easy to construct a general matrix $\mathcal{M}$ in n-dimensions. For different upper triangle matrix construction of vines in graphical theory, see Joe (2015, Chapter 3).

4.3.2 Multivariate Regular Vines Density

This section uses matrix $\mathcal{M}$ in a numerical method for multivariate regular vine density including multivariate canonical vine density and multivariate drawable vine density in n-dimensions. Previously, and n-by-n matrix $\mathcal{M}$ was appropriately generated with a proximity condition as discussed in the previous section. Now, the n-dimensional multivariate vine copulae density can be formulated with an additional two n-by-n matrices specification such that bivariate copulae types $\mathcal{T} = (t_{i,k})$ and its parameters $\mathcal{P} = (\mathfrak{p}_{i,k})$ where $i = n, ..., 2$ and $k = n-1, ..., 1$. First, the study introduces the density of vine copulae in term of a product of copulae indexing. A copula of Sklar is a multivariate distribution function, C, and the marginals are $F_1(x_1), ..., F_n(x_n)$ (Sklar, 1959). Therefore, the multivariate copula distribution is

$$F(x_1, ..., x_n) = C\{F_1(x_1), ..., F_n(x_n)\} \tag{4.3.2}$$

for n-dimensional data. After that, vine copulae density in n-dimensions can be expressed in graph theory. Mathematically, the C-vine density is

$$\prod_{j=1}^{n-1}\prod_{i=1}^{n-j} c_{j,j+i|1,...,j-1}\Big\{F(x_j|x_1, ..., x_{j-1}), F(x_{j+i}|x_1, ..., x_{j-1})\Big\}. \tag{4.3.3}$$

D-vine density is

$$\prod_{j=1}^{n-1}\prod_{i=1}^{n-j} c_{i,i+j|i+1,...,i+j-1}\Big\{F(x_i|x_{i+1}, ..., x_{i+j-1}), F(x_{i+j}|x_{x+1}, ..., x_{i+j-1})\Big\}. \tag{4.3.4}$$

R-vine density is

$$\prod_{i=1}^{n-1}\prod_{e\in E_i} c_{C_{e,a},C_{e,b}|D_e}(F_{C_{e,a}|D_e}(X_{C_{e,a}}|\mathbf{X}_{D_e}), F_{C_{e,b}|D_e}(X_{C_{e,b}}|\mathbf{X}_{D_e})). \tag{4.3.5}$$

One node of a D-vine in a particular tree T_i can only have one edge while in a C-vine a particular node in a particular tree T_i must have $n - j$ edges. The D-vine notation presents more independence than the C-vines notation. However, a C-vine may have an advantage when the study knows that there is one particular variable that influences the data set. The R-vine is a general case and has far more flexibility.

Here is an example of five dimensional regular vine copula density in Figure 4.1. The full R-vine copula density in Figure 4.1 can simply be formulated as follow

$$\begin{aligned} f(x_1, x_2, x_3, x_4, x_5) = {} & c_{1,2} \cdot c_{2,3} \cdot c_{3,4} \cdot c_{3,5} \text{ (unconditional pairs)} \\ & \times c_{1,3|2} \cdot c_{2,4|3} \cdot c_{2,5|3} \cdot c_{1,4|23} \cdot c_{4,5|23} \cdot c_{1,5|234} \text{ (conditional pairs)} \end{aligned}$$

where $c_{u,v|k}$ stand for $c_{u,v|k}(F_{u|k}(x_u|x_k), F_{v|k}(x_v|x_k))$. All definitions and further details can see, among others, Aas et al. (2009) for C-vines and D-vines, and in Dißmann et al.

(2013) for R-vines. Next, R-vine density is presented in graph theory using matrices $\mathcal{M}^*$, $\mathcal{T}^*$ and $\mathcal{P}^*$ in Figure 4.2 which is also usable for its special cases.

Figure 4.2: Graphical Model: Full Vine Copulae Specification in N-by-N Matrix Representation including T^*, P^* and M^* with the Proximity Condition

$$\mathcal{M}^* = \begin{pmatrix} m_{1,1} & & & \\ m_{2,1} & m_{2,2} & & \\ \vdots & \vdots & \ddots & \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n} \end{pmatrix},$$

$$\mathcal{T}^* = \begin{pmatrix} & & & \\ t_{2,1} & & & \\ t_{3,1} & t_{3,2} & & \\ \vdots & \vdots & \ddots & \\ t_{n,1} & t_{n,2} & \cdots & t_{n,n-1} & \end{pmatrix}, \mathcal{P}^* = \begin{pmatrix} & & & \\ \mathfrak{p}_{2,1} & & & \\ \mathfrak{p}_{3,1} & \mathfrak{p}_{3,2} & & \\ \vdots & \vdots & \ddots & \\ \mathfrak{p}_{n,1} & \mathfrak{p}_{n,2} & \cdots & \mathfrak{p}_{n,n-1} & \end{pmatrix}$$

Note that parameters in N dimensions are at least $N(N-1)/2$.

For the red label in Figure 4.2, the conditioned set is $\{m_{1,1}, m_{2,1}\} \in \mathcal{M}^*$ and the conditioning set $\mathcal{D}$ is all $m_{i,1},\ i = 3, .., n$ at tree $n-1$ (the last tree) with bivariate copula type and parameter are $t_{2,1}$ and $\mathfrak{p}_{2,1}$, respectively. The blue label is tree one, T_1. The conditioned set is $\{m_{2,2}, m_{n,2}\} \in \mathcal{M}^*$. The conditioning set $\mathcal{D}$ is $\varnothing$ with bivariate copula type and parameter are $t_{n,2}$ and $\mathfrak{p}_{n,2}$, respectively. In graph theory, the regular vine density is therefore

$$f_{1,\dots,n} = \prod_{k=1}^{n} f_k \prod_{k=n-1}^{1} \prod_{i=n}^{k+1} c_{m_{k,k},m_{i,k}|m^*}(F_{m_{k,k}|m^*}, F_{m_{i,k}|m^*}), \tag{4.3.6}$$

where m^* is $m_{i+1,k},\dots,m_{n,k} \in \mathcal{M}^*$ with bivariate copula types $\mathcal{T}^*$, bivariate copula parameters $\mathcal{P}^*$ and corresponding to the vines density in Equations 4.3.3, 4.3.4 and 4.3.5. For the R-vine density algorithm, see Dißmann et al. (2013)[Algorithm 2.1].

4.3.3 Marginal Distribution

This section presents the proposed marginal model, where a hybrid version of ICAPM and the asymmetric-GARCH model with mixture distribution to capture some dynamics of financial returns' characteristics such as volatility clustering and time varying heteroskedastic volatility. Besides, the advantages of the proposed marginal model are risk premium information and the capturing of a shock asymmetry and asymmetric fat tails properties of financial data. The study which comes closest to the proposed marginal model is by Hafner and Kyriakopoulou (2021). However, they used Gaussian innovation while the present work uses a more complex mixture innovation. Also, this is an extension of the proposed marginal model to a general dependence structure study in regular vine copulae.

Marginal Model

We first introduce ICAPM of Merton (1973) which is a linear-in-variance function of financial returns, y_t and is given by

$$y_t = \lambda_1 + \lambda_2 \mathbf{h}_t \ + \varepsilon_t, \tag{4.3.7}$$

where $t = \{1, ..., T\}$ is a time index, λ_1 is a risk-free rate, λ_2 is a risk premium, $\mathbf{h}_t$ is a conditional volatility, $\varepsilon_t = \sqrt{\mathbf{h}_t} z_t$ is an innovation with mean zero and the conditional variance (or volatility) $\mathbf{h}_t$, and z_t is the independent and identically distributed (iid) random variable with zero mean and unit variance. This paper makes an exponential-type GARCH model which is the Exponential-GARCH(p,q,r) (EGARCH(p,q,r)) model given by

$$\log \mathbf{h}_t = \omega + \sum_{i=1}^{p} \gamma_i z_{t-i} + \sum_{j=1}^{q} \delta_j |z|_{t-j} + \sum_{k=1}^{r} \nu_k \log \mathbf{h}_{t-k}. \tag{4.3.8}$$

For the details of the EGARCH model see, among others, Nelson (1991). For surveys of asymmetric GARCH including EGARCH, see, among others, Hentschel (1995) and Jose Rodriguez and Ruiz (2012).

Marginal Innovation

To innovate the marginal model, this paper employs a mixture distribution. While Sahamkhadam et al. (2018) used the ARMA-GARCH-Standard-Copulae model, this study utilises the ICAPM-EGARCH-Regular-Vine-Copulae model, making a novel contribution to the literature. In reference to mixtures, there are three mixture components, a Gaussian distribution and two generalised Pareto distributions (GPDs) in EVT. The two GPDs function are used to capture the asymmetric-fat-tails, which is one of the most important properties in financial data Engle (2004).

In statistics, mixture distribution has flexible properties. It can be represented by either a finite set of distribution functions (cdf), $P_k(x_t)$, or density functions (pdf), $p_k(x_t)$, $k = 1, \ldots, K$. The mixture distribution and density are a sum such that $F(x_t) = \sum_{k=1}^{K} w_k P_k(x_t)$ and $f(x_t) = \sum_{k=1}^{K} w_k p_k(x_t)$, respectively. $w > 0$ is a weight and $\sum w = 1$. If the mixing weight w is pre-specified, it can also represent the quantile level of the mixture distribution. Given the mixture components, the mixture pdf is

$$p(x_t) = \begin{cases} \frac{1}{\beta^L}\left(1 + \xi^L \frac{(x_t - \mu^L)}{\beta^L}\right)^{-\frac{1}{\xi^L}-1} & , x_t \le \Phi^{-1}(\alpha^L) \\ \phi(x_t) & , \Phi^{-1}(\alpha^L) < x_t < \Phi^{-1}(\alpha^R) \\ \frac{1}{\beta^R}\left(1 + \xi^R \frac{(x_t - \mu^R)}{\beta^R}\right)^{-\frac{1}{\xi^R}-1} & , x_t \ge \Phi^{-1}(\alpha^R), \end{cases} \tag{4.3.9}$$

where α^L and α^R are a pre-specified quantile, $\mu = (-\infty, \infty)$ is a location parameter, $\beta = (0, \infty)$ is a scale parameter, $\xi = (-\infty, \infty)$ is a shape parameter, and L and R denote the left and right tails, respectively. The support of $x_t \sim GPD(\mu, \beta, \xi)$ is $x_t \ge 0$ when

$\xi = [0, \infty)$ and $x_t = [\mu, \mu - \beta/\xi]$ when $\xi = (-\infty, 0)$, and $\xi = [0, 1/2]$ represents a fat tail and exists at least up to the second moment. More details for the moment property see Hosking and Wallis (1987) and Singh and Guo (1995).

The GPD, in the sense of a generalisation, consists of an ordinary Pareto distribution, an exponential distribution and a short-tailed, Pareto type II distribution. Further details of a popular GPD in the peak-over-threshold method in EVT are found in McNeil et al. (2015, Chapter 5). Note that $\phi(\cdot)$ and $\Phi^{-1}(\cdot)$ are the standard Gaussian pdf and inverse cdf, respectively.

4.4 Bayesian Inference and Machine Learning in the Graphical Model and N-Dimensions

This section describes the methods of the model parameter estimator in Bayesian inference and machine learning approach. This paper proposes two Bayesian approaches for modeling financial time series data: the Bayesian Markov chain Monte Carlo (MCMC) method and the variational Bayes method. These two Bayesian approaches are computationally intensive and mathematically complex. This paper adopts the rw-MH MCMC algorithm of Koop et al. (2007) and the variational Bayes with (without) data augmentation of Loaiza-Maya and Smith (2019) to a novel problem specific to the general R-vine dependence structure study. Importantly, these two methods can be particularly effective when the posterior density of the marginal model becomes intractable, as is the case with the proposed model. Moreover, this paper expands upon existing algorithms by utilising plugged-in parallel computing and the QMLE method. By applying these techniques to Bayesian inference and machine learning algorithms, the computational time is significantly reduced through iteration reduction, without sacrificing estimation accuracy.

4.4.1 Full Likelihood Function

It is important to discuss a likelihood function in Bayesian inference and machine learning since this is a connection between Bayesian MCMC and Bayesian machine leaning. A likelihood function is a joint distribution of the observed data given by a chosen mathematical model. Given the observed n-dimensional data and data standardisation, the full log likelihood function of the proposes model is

$$\begin{aligned}\Lambda(\theta|y_{k,t}) &= \sum_{k=1}^{n} log(f_k) \sum_{k=n-1}^{1} \sum_{i=n}^{k+1} log(c_{m_{k,k},m_{i,k}|m^*}(F_{m_{k,k}|m^*}, F_{m_{i,k}|m^*})) \\ &= \sum_{k=1}^{n} \sum_{t=1}^{T} log[q_k^L p_L^{GPD}(x_{k,t}) + (1 - q_k^L - q_k^R) p^G(x_{t,k}) + q_k^R p_R^{GPD}(x_{t,k})] \\ &\quad \sum_{k=n-1}^{1} \sum_{i=n}^{k+1} \sum_{t=1}^{T} log(c_{t_{m_{k,k},m_{i,k}|m^*}}(F_{t_{m_{k,k}|m^*}}, F_{t_{m_{i,k}|m^*}})),\end{aligned} \tag{4.4.1}$$

where

$$\begin{aligned}p_L^{GPD}(x_t^k) &= \frac{1}{\sqrt{\mathbf{h}_t^k}(1-\xi^{L,k})(1-2\xi^{L,k})^{1/2}} \left(1 + \frac{\xi^{L,k}(y_t^k - \lambda_1^k - \lambda_2^k \mathbf{h}_t^k)}{\sqrt{\mathbf{h}_t^k}(1-\xi^{L,k})(1-2\xi^{L,k})^{1/2}}\right)^{-\frac{1}{\xi^{L,k}}-1}, \\ p^G(x_t^k) &= \frac{1}{(2\pi\mathbf{h}_t^k)^{1/2}} exp\left[\frac{-(y_t^k - \lambda_1^k - \lambda_2^k \mathbf{h}_t^k)^2}{2\mathbf{h}_t^k}\right], \\ p_R^{GPD}(x_t^k) &= \frac{1}{\sqrt{\mathbf{h}_t^k}(1-\xi^{R,k})(1-2\xi^{R,k})^{1/2}} \left(1 + \frac{\xi^{R,k}(y_t^k - \lambda_1^k - \lambda_2^k \mathbf{h}_t^k)}{\sqrt{\mathbf{h}_t^k}(1-\xi^{R,k})(1-2\xi^{R,k})^{1/2}}\right)^{-\frac{1}{\xi^{R,k}}-1},\end{aligned}$$

which is compatible with equation 4.3.6. c_t is a unique bivariate copula pdf. F_t is a mixture cdf which is defined in Section 4.3.3. x_t is a standardised residual given by equation 4.3.7. The parameter θ is $\{\lambda_1^k, \lambda_2^k, \xi^{L,k}, \xi^{R,k}, \omega^k, \gamma^k, \delta^k, \nu^k, \mathcal{P}^*\}$. In n dimensions, there are at least $\frac{n(n+15)}{2}$ parameters that need to be estimated. It is a quadratic function in the similarity form of covariance matrix. The number of model parameter that need to be estimated is massive when high dimension. For example, Table 4.4 and Table 4.5 in an appendix B demonstrated the empirical estimated parameter of R-vine copula-based model and at least 606 parameters and 6,094 parameters in 30 dimensions and 100 dimensions, respectively. Considerably, the trade off between flexibility and parsimony may be taken into account. This is lead to the research area of regularisation or shrinkage method. The most common shrinkage methods are, for instance, lasso and ridge method Scherr and Zhou (2020); Nezamdoust and Eskandari (2022); Liang, Xu, Chen and Li (2022). For further research, regularised copula-based models could be the one interesting topic.

4.4.2 Bayesian MCMC Estimator

This section describes an alternative parallel algorithm of rw-MH sampling incorporated with QMLE in a novel model of regular vine copulae in n-dimensions. The main algorithm is presented in Algorithm 1. The step of this algorithm is referred to as the Inference for the Margins (IFM) method by Al Janabi et al. (2017). Algorithm 2 is performed within Algorithm 1 and returns the ultimate parameter estimator(s).

In summary, Algorithm 1 shows how to obtain the parameter estimates $\hat{\mathcal{P}}^*$ and $\hat{\theta}$ of R-vines and the margins, respectively. In the first step, the maximum matrix $\mathbb{M}$ is produced by $\mathcal{M}^*$. Element $\mathbf{m}_{i,k}$ in $\mathbb{M}$ is the maximum of all entries in the k^{th} column with all rows between *i* and *n*. The **first loop** computes the data standardisation in parallel. **In the second loop**, the algorithm also parallelly computes the independent pairs of the R-vine copula model in tree one T_1, while the **third loop** computes the dependent pairs of the R-vine copula model, which is the rest of the tree $T_i,\ i = 2, ..., n-1$. In general, the algorithm obtains QMLE estimators as an initial condition for the rw-MH sampling estimation method in Algorithm 2. $c(u, v; t, \mathfrak{p})$ is a specific-bivariate copula function with copula type t and copula parameter $\mathfrak{p}$. $v_{i,k}^{direct}$ and $v_{i,k}^{indirect}$ are recursive functions of the conditional bivariate cdf copula or $h(u, v|t, \mathfrak{p})$;

$$\begin{aligned} v_{i,k}^{direct} &= C(v_{m_{k,k}}|u_{m_{i+1,k}}, u_{m_{i+2,k}}, ..., u_{m_{n,k}}), \\ v_{i,k}^{indirect} &= C(v_{m_{i+1,k}}|u_{m_{k,k}}, u_{m_{i+2,k}}, ..., u_{m_{n,k}}), \end{aligned} \tag{4.4.2}$$

for all $n \geq i > 1$. Denote that $h(u, v|t, \mathfrak{p}) = C_{2|1}(v|u; \theta, \delta)$. $C(u, v; \theta, \delta) = G(x, y; \theta, \delta)$ where $G(x, y; \theta, \delta)$ is a bivariate survival function and x, y are monotonically decreasing transforms. Therefore, the conditional cdf is $C_{2|1}(v|u; \theta, \delta) = \frac{\partial G}{\partial x} \cdot \frac{\partial x}{\partial u}$. See Dißmann et al.

Algorithm 1 Parallel algorithm of regular vine copulae model using random walk chain Metropolis-Hastings sampling MCMC and maximum likelihood estimation.

Require: y_t

Calculate $\mathbb{M}_{i,k} = \mathbf{m}_{i,k} | i = n, k = 1, ..., n$. $\mathbf{m}_{n,k} = \mathit{max}\{\mathrm{m}_{i,k}, ..., m_{n,k}\}$ for all $k = n, ..., 1$. m is spanning tree maximisation.
parfor $\mathrm{i}_g \leftarrow 1, n$ **do** ▷ estimate marginal parameters from equation 4.3.7, 4.3.8 and 4.3.9, do parallel
    Perform $\hat{\theta}_{i_g}^{QMLE,Margins} \leftarrow \arg\max_{\theta\epsilon\Theta} log(f_{i_g}(\theta \mid y_{i_g}))$.
    Perform $\hat{\theta}_{i_g}^{rmMH,Margins} | (\hat{\theta}_{i_g}^{QMLE,Margins},\ y_{i_g})$ via call **Algorithm 2**
    Calculate z_{t,i_g} via standardise $y_{t,i_g} | \hat{\theta}^{rmMH}_{Margins,i_g}$.
    Calculate $u_{i_g} = F_{i_g}(z_{t,i_g}) \sim Unif(0,1)$.
end parfor
Allocate $V^{direct} = (v_{i,k}^{direct})$ where $i = n$ and $k = 1, ..., n$.
Allocate $(v_{n,1}^{direct}, ..., v_{n,n}^{direct}) = (u_1, ..., u_n)$.
parfor k $\leftarrow 1, n-1$ **do** ▷ first tree, do parallel
    Perform $\hat{\mathfrak{p}}_{n,k}^{QMLE}, BIC^{QMLE} \leftarrow \arg\max_{\theta\epsilon\Theta} log(c(z_{n,k}^{(2)}, z_{n,k}^{(1)}; t_{n,k}))$ using equation 4.4.1 for copula candidate.
    Find $\hat{\mathfrak{p}}_{n,k}^{QMLE}, t_{n,k}$ by $\mathit{min}\{BIC^{QMLE}\}$
    Set $z_{n,k}^{(1)} = v_{n,k}^{direct}$ and $z_{n,k}^{(2)} = v_{i,(n-\mathbf{m}_{n,k}+1)}^{direct}$.
    Perform $\hat{\mathfrak{p}}_{n,k}^{rmMH} | (z_{n,k}^{(1)},\ z_{n,k}^{(2)},\ \hat{\mathfrak{p}}_{n,k}^{QMLE})$ using equation 4.4.1 via call **Algorithm 2**
    Calculate $v_{n-1,k}^{direct} = h(z_{n,k}^{(1)}, z_{n,k}^{(2)} | t_{n,k}, \hat{\mathfrak{p}}_{n,k}^{rmMH})$
    Calculate $v_{n-1,k}^{indirect} = h(z_{n,k}^{(2)}, z_{n,k}^{(1)} | t_{n,k}, \hat{\mathfrak{p}}_{n,k}^{rmMH})$
end parfor
for k $\leftarrow n-2, 1$ **do** ▷ column iteration, iteration for tree $n-1$ to tree 2
    for i $\leftarrow n-1, k+1$ **do** ▷ row iteration
        Calculate $\mathbb{M}_{i,k} = \mathbf{m}_{i,k}$. $\mathbf{m}_{i,k} = \mathit{max}\{\mathrm{m}_{i,k}, ..., m_{n,k}\}$.
        Set $z_{i,k}^{(1)} = v_{i,k}^{direct}$.
        if $\mathbf{m}_{i,k} = m_{i,k}$ **then**
            $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{direct}$
        else
            $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{indirect}$
        end if
        Perform $\hat{\mathfrak{p}}_{i,k}^{QMLE}, BIC^{QMLE} \leftarrow \arg\max_{\theta\epsilon\Theta} log(c(z_{i,k}^{(2)}, z_{i,k}^{(1)}; t_{i,k}))$ using equation 4.4.1 for copula candidate.
        Find $\hat{\mathfrak{p}}_{i,k}^{QMLE}, t_{i,k}$ by $\mathit{min}\{BIC^{QMLE}\}$
        Perform $\hat{\mathfrak{p}}_{i,k}^{rmMH} | (z_{i,k}^{(2)}, z_{i,k}^{(1)}, t_{i,k}, \hat{\mathfrak{p}}_{i,k}^{QMLE})$ using equation 4.4.1 via call **Algorithm 2**
        Calculate $v_{i-1,k}^{direct} = h(z_{i,k}^{(1)}, z_{i,k}^{(2)} | t_{i,k}, \hat{\mathfrak{p}}_{i,k}^{rmMH})$
        Calculate $v_{i-1,k}^{indirect} = h(z_{i,k}^{(2)}, z_{i,k}^{(1)} | t_{i,k}, \hat{\mathfrak{p}}_{i,k}^{rmMH})$
    end for
end for
return $\hat{\mathcal{P}}^*$, $\hat{\mathcal{T}}^*$, $\hat{\mathcal{M}}^*$, $\hat{\theta}$

(2013) for further discussion.

Algorithm 2 shows the rw-MH sampling MCMC method in Algorithm 1. In general, Metropolis-Hastings algorithms depend on initial conditions (θ, Ω). The first step of rw-MH algorithms is discarding θ; the more discarding of θ there is, the more computational time is needed for rw-MH algorithms. This step depends on the choice of initial conditions. This paper uses the initial conditions from QMLE to facilitate the issue so as to obtain a reasonable acceptance rate faster, which is about 50%. z is called the increment random variable, and is a Gaussian increment, $z \sim N(0, c\Omega)$. c is a constant. It is also used for the sake of the appropriate acceptance rate. For further details of MH algorithms, see Koop et al. (2007, Exercise 17.5 and Exercise 11.18).

Algorithm 2 Random walk chain Metropolis-Hastings sampling MCMC algorithm.

Require: $\Lambda(\theta|y), \hat{\theta}^{QMLE}, \hat{\Sigma}^{QMLE}, c, nburn, nrep.$

Calculate $\Sigma^* = c \times \hat{\Sigma}^{QMLE}$.
Set $\theta^1 = \hat{\theta}^{QMLE}$.

for $i \leftarrow 2, nburn + nrep$ **do**
  Draw a candidate $\theta^* = \theta^{i-1} + z$
  Draw an acceptance probability

$$\alpha(\theta^{i-1}, \theta^*) = min\left[\frac{\Lambda(\theta^*|y)}{\Lambda(\theta^{i-1}|y)}, 1\right]$$

  Set $\theta^i = \theta^*$ with probability $\alpha(\theta^{i-1}, \theta^*)$. Otherwise, set $\theta^i = \theta^{i-1}$.
end for

Calculate posterior means $\hat{\theta}^{rwMH}$ from the draws θ and whatever posterior inference of interest are needed.
return $\hat{\theta}^{rwMH}$

4.4.3 Variational Bayes Estimator

This section presents an alternative parallel algorithm for applying the variational Bayes method to Bayesian inference and machine learning in n-dimensional general regular vines. This paper follows the use of variational Bayes method with (and without) data augmentation (VBDA) estimation in the specific drawable vine model of Loaiza-Maya and Smith (2019). We extend the VBDA method into the general dependence model of regular vines with an additional stopping criterion in order to reduce the iterative learning step and hence spend less computational time in the VBDA algorithm. In particular, this paper explores three variational Bayes estimation methods, two types of variational Bayes with latent variables and data augmentation (VBDA1 and VBDA2) and one variational Bayes with latent variables without data augmentation (VBDA0). In this study, these algorithms are also plugged-in QMLE estimators and a parallel computing technique in order to mitigate the computational time and convergence issue.

The augmented posterior density is

$$p(\theta, \mathbf{u}|y) = \frac{f(y, \mathbf{u}, \theta)}{f(y)} = \big(c(\mathbf{u}|\theta)p(\theta)\prod_{t=1}^{T}\mathcal{I}(a_t \le u_t < b_t)\big)/f(y).$$

$p(\theta)$ is the prior density. $f(y)$ is the marginal likelihood function. $c(\mathbf{u}|\theta)$ is the copula density. $\mathcal{I}$ is the indicator function. VB estimation is approximated $p(\theta, \mathbf{u}|y)$ by a tractable density $q_\lambda(\theta, \mathbf{u})$ in a so-called variational approximation. This estimation corresponds to minimising the Kullback-Leibler (*KL*) divergence

$$KL(q_\lambda(\theta, \mathbf{u})||p(\theta, \mathbf{u}|y)) = \int log\left(\frac{q_\lambda(\theta, \mathbf{u})}{p(\theta, \mathbf{u}|y)}\right)q_\lambda(\theta, \mathbf{u})\, d\theta\, du$$

or maximising the lower bound of the logarithm of the marginal likelihood (or the evidence lower bound (ELBO)), given by

$$\mathcal{L}(\lambda) = \int log\left(\frac{p(\theta)f(y, \mathbf{u}|\theta)}{q_\lambda(\theta, \mathbf{u})}\right)q_\lambda(\theta, \mathbf{u})\, d\theta\, du,$$

where θ and $\mathbf{u}$ are independent. For further details and examples of variational Bayes methods, see, among others, Gunawan, Tran, Suzuki, Dick and Kohn (2019) and Tran, Nguyen and Dao (2021). For recent reviews of variational Bayesian inference in machine learning, see, among others, Blei, Kucukelbir and McAuliffe (2017) and M. S. Smith, Loaiza-Maya and Nott (2020).

Variational Approximation

The density approximation of q_λ can be written as $q_\lambda(\theta, \mathbf{u}) = q_{\lambda^a}(\theta)q_{\lambda^b}(\mathbf{u})$. The specification of q_{λ^a} and q_{λ^b} is a crucial key. q_{λ^a} has parameter λ^a. q_{λ^b} has parameter λ^b with support on $[a_1, b_1) \times ... \times [a_T, b_T)$. This paper selects the choice of **u** approximation as follows:

$$DA0 : q_{\lambda^b}(u) = 0,$$
$$DA1 : q_{\lambda^b}(u) = \prod_{t=1}^{T} \frac{1}{b_t - a_t} \mathcal{I}(a_t \leq u_t < b_t),$$
$$DA2 : q_{\lambda^b}(u) = \prod_{t=1}^{T} \frac{\phi(x_t; \mu_t, \Omega_t)}{(b_t - a_t)\phi(x_t)}.$$

$DA2$ nests $DA1$, which covers a wide range of data dependency with an effective mean field approximation. $DA1$ is expected to be more accurate for data with low dependence. $\lambda^b = \{\mu_1, ..., \mu_T, log\ \omega_1, ..., , log\ \omega_T\}$, $\Omega_t = diag(\omega_1, ..., \omega_T)$, ϕ is the standard Gaussian density.

For the approximation of θ, this paper uses the popular choice for q_{λ^a}, which is a multivariate Gaussian distribution, $N(M, \Sigma)$. In the R-vines application, a huge of number of parameter estimates must be obtained. Since the closed form solution of multivariate Gaussian distribution is provided, it is quick to calculate the gradient of $log\ q_{\lambda^a}(\theta)$. Further, Σ improves the accuracy of the gradient estimate. The gradient of factor representation of $\Sigma = BB' + D^2$ has been derived and proved to be fast to compute (Ong, Nott & Smith, 2018).

Algorithm 3 demonstrates the VBDA algorithm with stochastic gradient ascent optimisation. Note that, in Algorithm 1, Algorithm 2 is replaced by Algorithm 3 in order to perform VBDA method in R-vine model. This approach is tractable posterior distribution. A vector of control variates ς is used to reduce the variance of an unbiased estimator of the gradient ∇. There are three proposed VB estimators of regular dependence models, which are specified in **Line 4** where $q_{(u)}$ can be $\{DA0, DA1, DA2\}$. To implement the

Algorithm 3 Algorithm of variational Bayes data augmentation with control variates and ADADELTA learning rate.

Require: $\Lambda(\theta|y), DA, \lambda^{(0)}, \mu^{QMLE}, S, nVB, c, t_w, P, \rho, \epsilon.$

Set $k = 1$.
Initialise $\lambda|\mu^{QMLE}$.
Generate $(\theta_s^{(k)}, \mathbf{u}_s^{(k)}) \sim q_{\lambda^k}(\theta, u)$ for $s = 1, ..., S$ where
$q_{\lambda^b}(\mathbf{u})$ could be *i.* $DA0$ *ii.* $DA1$ or *iii.* $DA2$ and $q_{\lambda^a}(\theta) \sim N(M_\theta, \Sigma_\theta)$.
Calculate $\varsigma_i^{(k)} = (\varsigma_1^{(k)}, ..., \varsigma_m^{(k)})$ with

$$\varsigma_i^{(k)} = \frac{cov\Big(\big[log\ h(\theta,\mathbf{u}) - log\ q_\lambda(\theta,\mathbf{u})\big] \times \nabla_{\lambda_i} log\ q_\lambda(\theta,\mathbf{u}), \nabla_{\lambda_i} log\ q_\lambda(\theta,\mathbf{u})\Big)}{var(\nabla_{\lambda_i} log\ q_\lambda(\theta,\mathbf{u}))}$$

Set $k = k + 1$.
while *stopping rule* is not satisfied **do**
    Generate $(\theta_s^{(k)}, \mathbf{u}_s^{(k)}) \sim q_{\lambda^k}(\theta, u)$ for $s = 1, ..., S$.
    Calculate $\nabla_\lambda \left(\widehat{\mathcal{L}(\lambda^{(k)})}\right) = \left(g_{\lambda_1}^{(k)}, \ldots, g_{\lambda_m}^{(k)}\right)'$

$$g_{\lambda_i}^{(k)} = \frac{1}{S}\sum_{s=1}^{S}\left(\log h(\theta_s^{(k)}, \mathbf{u}_s^{(k)}) - \log q_\lambda(\theta_s^{(k)}, \mathbf{u}_s^{(k)}) - \varsigma_i^{(k-1)}\right) \times \nabla_{\lambda_i} \log q_\lambda(\theta_s^{(k)}, \mathbf{u}_s^{(k)})$$

    Calculate $\varsigma_i^{(k)} = (\varsigma_1^{(k)}, ..., \varsigma_m^{(k)})$.
    Perform $\Delta\lambda^{(k)} \leftarrow$ the ADADELTA method.
    Set $\lambda^{(k+1)} = \lambda^{(k)} + \Delta\lambda^{(k)}$
    Calculate $\widehat{LB}(\lambda^{(k+1)}) = \frac{1}{S}\sum_{s=1}^{S}\left(\log h(\theta_s^{(k+1)}, \mathbf{u}_s^{(k+1)}) - \log q_\lambda(\theta_s^{(k+1)}, \mathbf{u}_s^{(k+1)})\right)$
    if $k > c$ **then**
        Calculate $\overline{LB}_{k-t_w} = 1/t_w \sum_{i=1}^{t_w} \widehat{LB}(\lambda^{(k-i+1)})$
        if $\overline{LB}(\lambda^{(k-t_w)}) > max(\overline{LB})$ **then**
            $patience = 0$.
            Collect $\lambda^{(*)} \leftarrow \lambda^{(k+1)}$.
        else
            $patience = patience + 1$.
        end if
        if $patience > P$ **then**
            stop.
        end if
    end if
    Set $k = k + 1$.
end while
return $\lambda^{(*)}$

algorithm, the initial $\lambda^{(0)}|\mu^{QMLE}$ from QMLE is required. The estimated LB is then sequentially computed in **Line 12** by $\lambda^{(k+1)} = \lambda^{(k)} + \rho^{(k)}\nabla_\lambda\widehat{\mathcal{L}(\lambda^{(k)})}$, where $\rho^{(k)}$ is the learning rate, and $\nabla_\lambda\widehat{\mathcal{L}(\lambda^{(k)})}$ is the unbiased estimate of LB's gradient $\nabla_\lambda\mathcal{L}(\lambda^{(k)})$ in the ADADELTA method in **Line 11**. For more details of an adaptive rate method, see Zeiler (2012). $\nabla_\lambda\mathcal{L}(\lambda)$ is the expectation of $\nabla_\lambda log\ q_\lambda(\theta,\mathbf{u})$ with respect to $q_\lambda(\theta,\mathbf{u})$ such that $\nabla_\lambda\mathcal{L}(\lambda) = E_q(\nabla_\lambda log\ q_\lambda(\theta,\mathbf{u})\{log\ h(\theta,\mathbf{u}) - log\ q_\lambda(\theta,\mathbf{u})\})$ where $p(\theta,\mathbf{u}|y) \propto f(y,\mathbf{u}|\theta)p(\theta) = h(\theta,\mathbf{u})$. Hence, the unbiased estimate of LB's gradient is $\nabla_\lambda\widehat{\mathcal{L}(\lambda^{(k)})} = (g_{\lambda_1},...,g_{\lambda_m})'$ where $g_\lambda = \frac{1}{S}\sum_{s=1}^{S}\big(log\ h(\theta_s,\mathbf{u}_s) - log\ q_\lambda(\theta_s,\mathbf{u}_s)\big)\nabla_\lambda log\ q_\lambda(\theta_s,\mathbf{u}_s)$. Note that m is the number of elements of λ. For further discussion of the algorithm, see Dißmann et al. (2013). The *stopping rule* is to terminate the whole procedure if

(i.) there is a fixed number of stochastic gradient ascent steps, nVB, or

(ii.) the moving average $\widehat{LB}$ is not improved, $\overline{LB}_{k-t_w} = 1/t_w\sum_{i=1}^{t_w}\widehat{LB}(\lambda^{(k-i+1)})$, which is referred to as the $patience$ parameter and $patience = patience + 1$ in **Line 20** and the whole procedure stops when $patience >$ maximum patience P in **Line 22-23**, and

(iii.) all above conditions will be checked when the k^{th} iteration is greater than a constant, c.

Finally, $\lambda^{(*)}$ is collected when $\overline{LB}_{k-t_w}$ is greater than $max(\overline{LB})$ (Tran et al., 2021). Note that the simulation study in Bayesian inference and machine learning including the rw-MH algorithm and the VBDA algorithm is presented in Appendix B.1.

4.5 Empirical Study in High Dimensions

This section presents empirical study results of the novel proposed Bayesian inference and machine learning in general R-vine model in high dimensions. The 13 bivariate copula families are adopted for selection in a regular vine path dependence structure. The mixture distribution is introduced into the univariate marginal ICAPM-EGARCH model, which is intractable margins, to improve the capture of financial time series characteristics. For the estimation method, this study explores the traditional QMLE method as a benchmark and the proposed alternative estimation method in Bayesian inference and machine learning including the rw-MH algorithm and variational Bayes with (and without) data augmentation (VBDA) algorithms in the R-vine model and up to 100 dimensions. Hence, there are up to 4,950 bivariate copula functions and 6,138 parameters that were estimated in the R-vine model. The empirical results showed that the proposed variational Bayes with data augmentation type 1, VBDA1, with mixed R-vine and mixture innovation in a marginal model is an outstanding model regarding a marginal likelihood value in high dimensions. Besides, the MAD value for three months also shows that VBDA1 is the best forecasting method. Interestingly, variational Bayes inference appears to offer a significant improvement in forecast performance as the number of dimensions increases, making it a potential alternative to traditional method. More precisely, in variational Bayes inference, the greater the number of dimensions and the longer the forecast period, the better the measure of variability. By contrast, the QMLE method seems to deteriorate in high dimensions.

Multivariate copula models can be fitted to n-dimensional time series data through a sequential method in two stages, see Dißmann et al. (2013) for more details.

In the first stage, a univariate marginal model is individually selected and fitted to the data in order to obtain standardised residuals and then transformed to marginally uniform data. This study uses the proposed marginal model as described in Section 4.3.3. The competing marginal model is the popular marginal ARMA(1,1)-GARCH(1,1)-Student-*t* model.

In the second stage, a dependence structure among these normalised residuals is formulated using the multivariate copula model. In the literature, it is broadly accepted that different pairs of variables possess different asymmetric and tail dependencies. Consequently, the standard multivariate copula model cannot capture these facts but vine copulae can. While D-vine and C-vine copulae do a good job in capturing dependence structure, there is some restriction, for example, in its particular path structures. This paper, hence, studies a novel flexible R-vine copula as a general vine dependence structure rather than a sub-class of D-vine and C-vine copulae among efficient Bayesian and machine learning estimators. The R-vines are very flexible and present a far more complicated model than the standard copula and its two special sub-classes of vine because of a massive of choice of path structures and pair-copula families. Precisely, in high dimensional data, the possible path structure number of an R-vine grows massively with $\frac{n!}{2}2^{\binom{n-2}{2}}$, see Morales Napoles, Cooke and Kurowicka (2010) for more details.

This paper studies high dimensional time series data in financial markets of up to 100 dimensions. For the training set, the study selected the daily adjusted price of the stock index of the top 100 market capitalisation stocks of the Nasdaq stock market between January 02, 2009 and September 20, 2021, with the stock index having to be listed before 2009 (in total 3,201 daily observations). The test set is between September 21, 2021, and December 21, 2021 (in total, 65 daily observations). All data from Yahoo Finance were retrieved via MATLAB data feed function. After data cleaning was done, the study obtained 100 stock indices for the empirical experiment study. For the list of the 100 stock indices, see Appendix B.5. Note that the proposed models in Section 4.3 were experimented with for fit by using two sets of stock indices up to 100 time series and 30 time series (the last 30 stock indices analysed). The aim of two empirical experiments was to explore the proposed model's performance when the number of dimension increases to high levels.

The table 4.1 shows descriptive statistics of overall stock returns on top five business sectors and can be summarised as follows. Top five business sectors of the analysis are in technology (22 firms), health care (18 firms), finance (11 firms), consumer services (11 firms) and capital goods (8 firms), and market capitalisation is US$30,814 billion and represents 46% of the total market capitalisation in the Nasdaq market. For all stocks, mean of return is positive and the return range is [-39.02%, 52.29%]. The range of standard deviation is [1.06%, 3.67%]. The range of skewness is [-1.04%, 1.56%]. The range of kurtosis is [5.95%, 47.26%].

Table 4.1: Top Five Business Sectors: Descriptive Statistics of Overall Stock Returns

Business Sector	**Number of Firm**
Technology	22
Health care	18
Finance	11
Consumer services	11
Capital goods	8
Mean	[-39.02, 52.29]*
Standard deviation	[1.06, 3.67]*
Skewness	[-1.04, 1.56]*
Kurtosis	[5.95, 47.26]*

Note that * represents the maximum and minimum statistical values of stock returns (%) in the top five business sectors of this study.

Therefore, all returns identify an asymmetrical and heavy-tailed distribution. Additional, the Jarque-Bera test statistic value of all returns confirms that all returns reject the null hypothesis that the return comes from a Gaussian distribution at the 1% significance level.

4.5.1 Regular Vine Distribution Selection in N-Dimensions

The following tasks are the sequential steps of fitting an R-vine copula with marginal specification:

(a) Standardisation of the residuals of the returns using the univariate marginal model.

(b) Selection of the R-vine path structure of the tree, $\mathcal{M} = (N, E)$ given by maximise empirical Kendall's τ via

b.1 Formulation of empirical Kendall's τ of all possible pairs variable, $\hat{\tau}_{i,k},\ 1 \leq i < k \leq n$,

b.2 Performing of spanning tree maximisation (the sum of absolute empirical

Kendall's τ)

$$max \sum_{e=\{i,k\}\ in\ spanning\ tree} |\hat{\tau}_{i,k}|,$$

b.3 For each edge, selection of bivariate copula family $t_{i,k}$ and its parameter(s) $\mathfrak{p}_{i,k}$ from all possible bivariate copula candidates via Bayesian information criterion (BIC) value

(c) Iteration of the step (b) from the first tree T_1 (unconditional tree, $\hat{\tau}_{i,k}$) to the last tree T_{n-1} (conditional tree, $\hat{\tau}_{i,k|D}$ where D is the proximity condition).

The above methodology returns all full specifications of a regular vine including $\mathcal{M}, \mathcal{T}, \mathcal{P}$, and the QMLE was used in step *b.3* for the estimation of all possible parameter(s). Also, this full specification of a regular vine copula is used in the Bayesian inference and machine learning. It is important to mention that, for the first tree, parallel computation is possible because of the independent path structure in R-vines. Hence, the particular steps of this method are as follows. The first step of the first tree, parallel computation technique is performed in order to obtain $\hat{\tau}_{1,k}$ where $k = 1, ..., n-1$. The second step is the **FOR** loop in order to obtain $\hat{\tau}_{i,k}$ where $i < k, i = n-1, ..., 2,\ k = 1, ..., n-2$. For further discussion of a maximum spanning tree algorithm, see Dißmann et al. (2013).

This paper uses various bivariate copula functions to capture all possible (a)symmetrical dependence properties of financial time series, while, four main copula families including Gaussian, Student-*t*, Gumbel and Frank copulae were considered, reflecting the research paper by Dißmann et al. (2013). This study hence considered 13 copula families and all candidates of the bivariate copula families used in this study are as follows.

- Gaussian copula presents symmetric and no tail dependence.
- Student-*t* copula presents symmetric and upper and lower tail dependence.
- Frank copula presents symmetric and tail independence.
- Clayton copula presents asymmetric and lower tail dependence.

- B5/Joe, Gumbel and Galambos copulae present asymmetric and different upper and lower tail dependence
- BB1, BB3, BB4, BB5, BB7 and BB10 copulae present asymmetric and different upper and lower tail dependence.

Bivariate copula distribution and some important dependence structure properties are summarised in Appendix B.3 such as the cumulative distribution function, lower tail and upper tail orders and rank-based measurements of dependence. See further discussion about copula properties in, among others, Joe (2015, Chapter 4) and Nikoloulopoulos, Joe and Li (2012, Appendix). We performed Ljung-Box tests for the standardised residuals of all the proposed models and these tests indicated independence of the standardised residuals. For the details, see in Appendix B.4.

Figure 4.3: Empirical Results: Normalised Mixed R-Vine Copula Contour Plot in Lower Triangular Matrix in 30 Dimensions

Note that, according to the limitations of presentation, 10 representatives out of 30 pairwise mixed R-vine dependence measurements are demonstrated. The upper triangular matrix presents the Kendall's $\hat{\tau}$ and Spearman $\hat{\rho_S}$ of the R-Vine-ICAPM-EGARCH-Mixture model using the variational Bayes (VBDA1) method, respectively.

Figure 4.3 presents an example of the empirical results of a normalised R-vine copula contour plot and a rank-based dependence measurement in the univariate ICAPM-EGARCH-Mixture marginal model with the VBDA2 estimation method in a 30-dimensional analysis. According to the limitations of presentation in this paper, the example shows a normalised 10-dimensional R-vine contour plot of the stock indices (stock name is located in the diagonal matrix.) in a lower triangular matrix where the row represents the i^{th} tree ($i = 1, ..., 9$) out of 30 stock indices. The upper triangular matrix presents empirical Kendall's $\hat{\tau}^k$ and $\hat{\rho_s}^k$, respectively, where the k^{th} column ($k = 10, ..., 2$) represents tree $\hat{\mathcal{T}_i}$ where $i = 1, ..., 9)$. Figure 4.4 is similar to Figure 4.3 but it demonstrates the empirical specimen of a normalised 10-dimensional R-vine contour plot and Kendall's $\hat{\tau}$ and Spearman $\hat{\rho}_s$ of the stock indices in a high 100-dimensional experiment. See the full empirical result including empirical Kendall's $\hat{\tau},\ \hat{\rho}_s,\ \hat{\mathcal{M}}$ and $\hat{\mathcal{T}}$ of Figure 4.3 and Figure 4.4 in Appendix B.2.

Figure 4.4: Empirical Results: Normalised Mixed R-Vine Copula Contour Plot in Lower Triangular Matrix in 100 Dimensions

Note that, according to the limitations of presentation, 10 representatives out of 100 pairwise mixed R-vine dependence measurements are demonstrated. The upper triangular matrix presents the Kendall's $\hat{\tau}$ and Spearman $\hat{\rho_S}$ of the R-Vine-ICAPM-EGARCH-Mixture model using the variational Bayes (VBDA1) method, respectively.

Table 4.2 summarises the proposed model's performance in 30 dimensions using the training data set and test data set, as follows.

For the training set, the proposed VBDA1-R-vine-ICAPM-EGARCH-Mixture model has the best performance thanks to the marginal likelihood value. The second and the third best performance due to marginal likelihood are still the variational Bayes with data augmentation type 2 and without data augmentation, respectively. As for the most complicated estimation method of VBDA2, there is no argument as to why the computation of the R-vine copula took the greatest computational time. VBDA1 is the second most complex estimation method, and the computational time is therefore comparatively less than VBDA2; it is, however, less than the less-complicated estimation methods such as VBDA0 because of the virtue of convergence. See the example of an evidence lower bound (ELBO) convergence of the stock index pairwise (DE,CAT) of all VBDA estimation methods in Figure 4.5 and the example of the first graphical tree path dependence of the VBDA2-R-vine-ICAPM-EGARCH-Mixture model from the spanning tree maximisation algorithm in Figure 4.6, which is explained at the beginning of this section. In terms of Akaike information criterion (AIC), BIC and likelihood value, the competing QMLE-R-vine-ARMA-GARCH-Student-*t* model performs the best. The second best performance according to AIC, BIC and likelihood value is rw-MH and the R-vine-ICAPM-EGARCH-Mixture model with an appropriate acceptance rate.

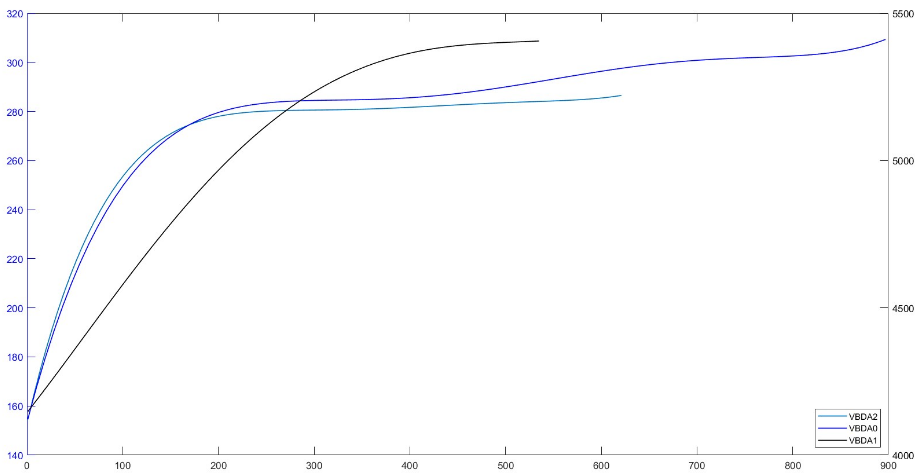
For the test set, forecasting performance was calculated for one-month and three-month periods using MAD and RMSE. For forecast error value, there is a similar predictive performance for both QMLE and rw-MH methods and these are the best forecasting models. Interestingly, in variational Bayes inference, when the forecasting period of time increases, VBDA1 seems to perform better compared to the other VBDA, according to MAD value. Further evidence can see in the experiment in high dimensions.

Table 4.2: Empirical Results: Performance Comparison in Models and Estimation Methods in 30 Dimensions for Both Training Set and Test Set

Model	Est. Metd.	Training Data						Test Data			
								1 month		3 months	
		Mrgnl. Lglik.	AIC	BIC	Lglik	Last nVB Step	Time	MAD	RMSE	MAD	RMSE
MD1	QMLE	n.a.	302,512	302,955	-150,347	n.a.	23	**1.22**	**1.63**	1.43	**1.94**
MD1	rw-MH	1,381,159	300,006	302,968	-149,107	n.a.	1,484	**1.22**	**1.63**	1.42	2.03
MD1	VBDA0	1,395,092	320,038	323,005	-159,122	944	3,586	1.50	2.57	1.54	2.26
MD1	VBDA1	**1,402,133**	328,476	331,339	-163,360	489	1,465	1.45	2.31	1.48	2.21
MD1	VBDA2	1,395,652	318,248	321,177	-158,234	728	10,530	1.35	2.01	1.54	2.19
MD2	QMLE	n.a.	**298,389**	**298,677**	**-148,359**	n.a.	28	**1.22**	**1.63**	**1.41**	2.04

Note that MD1 = R-vine-ICAPM-EGARCH-Mixture model. MD2 = R-vine-ARMA-GARCH-Student-t model. The acceptance rate of rw-MH method is 49 %. Time is presented in minute. Est. Metd. = Estimation Method. n.a. = not available. Bold font indicates better results. Mrgnl. Lglik. = Marginal log likelihood. Accpt. Rate = acceptance rate with 49%. MAD = mean absolute deviation. RMSE = root mean square error. Accpt. Rate and Last nVB Step are an average of all iterations. Time = time computation only in the copula dependence model in minutes. The last test data section in this table is an average error measure obtained between September 21, 2021 and December 21, 2021 in up to 100 simulations. A note on this computational time: the study implemented the experiments in MATLAB and performed them on a laptop computer with the 11^{th} Gen Intel(R) Core(TM) i7-1185G7, 3.00GHz, and 32 GB RAM.

Figure 4.5: Empirical Results: The Specimen of Evidence Lower Bound (ELBO) Convergence Representative for all VBDAs in 30 Dimensions



Note that this is a demonstration of ELBO convergence of a stock pairwise (DE,CAT) with the selected Student-t copula function and the stopping rule in Algorithm 3 of all VBDA methods including VBDA0, VBDA1 and VBDA2. The ELBO values of VBDA0 are plotted against the right y-axis. The ELBO values of VBDA1 and VBDA2 are plotted against the left y-axis.

Figure 4.6: Empirical Results: Graphical Tree 1 in 30 Dimensions for an R-vine from Algorithm Selection using VBDA2 in the R-vine-ICAPM-EGARCH-Mixture Model

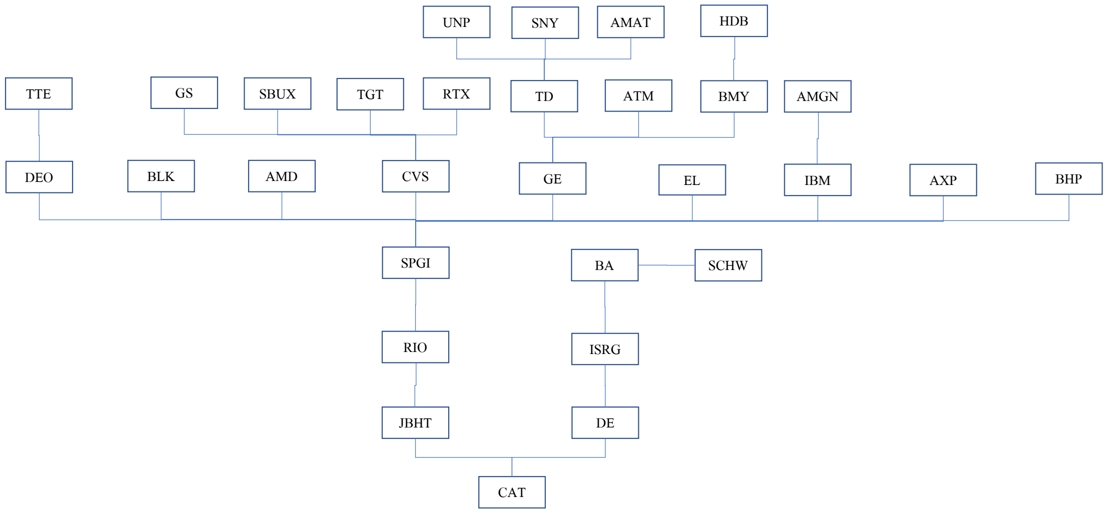


Table 4.3 summarises the proposed models performance in 100 dimensions using the training data set and test data set as follows.

Overall, once the number of dimensions increases up to a high dimensions, the variational Bayes model tends to improve the performance. In general, Bayesian inference and machine learning tend to be an outstanding method rather than traditional QMLE method, especially variational Bayes. It implies that Bayesian inference and machine learning in particular variational Bayes with the data augmentation type 1 algorithm may perform well in big data analysis. According to the training and test data, this study found that variational Bayes with data augmentation type 1 (VBDA1) with the R-vine-ICAPM-EGARCH-Mixture model can beat all other models according to the marginal likelihood value and the three months' forecast error of the MAD value[2].

Given the other model comparisons, the study could argue that the rw-MH with the R-vine-ICAPM-EGARCH-Mixture model is the most outstanding model, again with a reasonable acceptance rate. Variational Bayes inference is, however, one of

[2]The VBDA1-R-vine-ICAPM-EGARCH-Mixture model has the best performance according to three-months-MAD at three decimal digits.

the best outstanding model. The second most preferable model is the VBDA0-R-vine-ICAPM-EGARCH-Mixture model according to AIC value and BIC value. Obviously, the traditional QMLE method seems to decrease in performance in high dimensions in training data; however, in test data, QMLE method could be an option.

Table 4.3: Empirical Results: Performance Comparison in Models and Estimation Methods in 100 Dimensions for Both Training Set and Test Set

		Training Data					Test Data			
Model	Est. Metd.						1 month		3 months	
		Mrgnl. Lglik.	AIC	BIC	Lglik	Last nVB Step	MAD	RMSE	MAD	RMSE
MD1	QMLE	n.a.	969,425	1,015,504	-477,779	n.a.	1.18	1.59	1.35	**1.85**
MD1	rw-MH	4,372,594	**906,846**	**948,133**	**-446,490**	n.a.	**1.17**	**1.58**	**1.34**	**1.85**
MD1	VBDA0	4,483,822	915,862	956,847	-451,037	1,116	1.19	1.60	1.37	1.90
MD1	VBDA1	**4,485,991**	963,895	1,005,219	-475,009	567	1.21	1.63	**1.34**	1.86
MD1	VBDA2	4,482,233	917,936	959,083	-452,053	1,086	1.19	1.61	1.37	1.89
MD2	QMLE	n.a.	964,942	1,010,414	-475,638	n.a.	1.20	1.61	**1.34**	1.88

Note that MD1 = R-vine-ICAPM-EGARCH-Mixture model. MD2 = R-vine-ARMA-GARCH-Student-*t* model. Est. Metd. = Estimation Method. Mrgnl. Lglik. = Marginal log likelihood. n.a. = not available. The acceptance rate of rw-MH method is 48%. Bold font indicates better results. MAD = mean absolute deviation. RMSE = root mean square error. Acceptance rate and Last nVB Step are an average of all iterations. The last test data section in this table is an average error measure obtained between September 21, 2021 and December 21, 2021, in up to 100 simulations.

Table 4.4 and Table 4.5 summarise the algorithm selection of bivariate copula function from Algorithm 1 for all experiments in 30 dimensions and 100 dimensions in the regular vine model. There are 435 pair copula functions in 30 dimensions and 4,950 pair copula function in 100 dimensions. In 30 dimensions, the most outstanding model with VBDA1 has 638 parameters to be estimated; this is, comparatively, the lowest parameter number to estimate. In 100 dimensions, there are 6,138 estimated parameters in VBDA1 which is the highest number of parameters, while VBDA0 turned out the lowest number of parameter estimates in the algorithm selection. Generally, the elliptical copula family is the family best fitted to empirical data. Three main bivariate copula functions including the Student-*t* copula, Gaussian copula and Frank copula were chosen by algorithm selection which covered at least 68% of the total number of copula functions in the analysis. It is implied that at least 32% of empirical data has asymmetric and tail dependence properties. In 30 dimensions, the top two bivariate copula functions of algorithm selection are the Student-*t* copula and Frank copula, while, in 100 dimensions, the elliptical copula family (Gaussian and Student-*t* copulae) provide the top two copulae that fitted to the real data. Furthermore, the BB6 copula and BB10 copula have no selection from the algorithm in 30 dimensions and 100 dimensions, respectively.

Table 4.4: Empirical Results: Algorithm Selection of 30 Dimensions of Copula Family in Models and Estimation Methods

Model	**R-vine-ICAPM-EGARCH-Mixture**					**Competing Model**
Estimation Method	**VBDA0**	**VBDA1**	**VBDA2**	**rw-MH**	**QMLE**	**QMLE**
No. of Estimated Parameters	657	638	650	656	682	606
No. of Copula Families	435					
Gaussian	53	73	54	37	39	44
Student-*t*	156	119	151	156	143	118
Frank	111	111	114	124	121	178
Clayton	19	23	24	29	31	24
Gumbel	17	12	12	14	16	6
B5/Joe	8	4	8	3	7	6
Galambos	5	9	8	7	3	6
BB1	6	10	9	4	7	3
BB3	1	1	3	1	1	0
BB4	1	5	1	1	0	0
BB6	0	0	0	0	0	0
BB7	12	15	14	12	12	1
BB10	46	53	37	47	55	49

Note that the competing model is R-vine-ARMA-GARCH-Student-*t* model.

Table 4.5: Empirical Results: Algorithm Selection of 100 Dimensions of Copula Family in Models and Estimation Methods

Model	**R-vine-ICAPM-EGARCH-Mixture**					**Competing Model**
Estimation Method	**VBDA0**	**VBDA1**	**VBDA2**	**rw-MH**	**QMLE**	**QMLE**
No. of Estimated Parameters	6094	6138	6115	6133	6133	6133
No. of Copula Families			4950			
Gaussian	1537	1659	1535	1527	1527	1527
Student-*t*	1226	1526	1215	1207	1207	1207
Frank	617	777	625	633	633	633
Clayton	528	254	526	526	526	526
Gumbel	440	228	446	452	452	452
B5/Joe	196	70	193	193	193	193
Galambos	178	185	174	173	173	173
BB1	141	157	142	141	141	141
BB3	51	40	49	43	43	43
BB4	24	47	29	34	34	34
BB6	6	3	10	11	11	11
BB7	6	4	6	10	10	10
BB10	0	0	0	0	0	0

Note that the competing model is R-vine-ARMA-GARCH-Student-*t* model. There is a slightly different in experimental result for 100 dimensions compared to 30 dimensions of bivariate copula selection in Algorithm 1. Namely, the full selection of bivariate copulae was used in 30 dimensions experiment as explained in line 12 and line 28 of Algorithm 1; in 100 dimensions, the study used a one-step selection algorithm according to QMLE method to mitigate the time consumed in high dimensions in Bayesian inference and machine learning.

4.6 Conclusion and discussion

This paper provides a significant contribution by first introducing a variational Bayes without (with) data augmentation type 1 and type 2 in Bayesian inference and machine learning with rw-MH Bayesian MCMC in Bayesian inference and competing QMLE to a regular vine copula model in graph theory in n-dimensions. While standard multivariate copulae have some limitations regarding a single control parameter and this is even worse when the analysis is in high dimensions. Sub-class C-vine and D-vine copulae are proposed with different choices of bivariate copula functions. This gives in flexibility by tree construction of pairwise copula; however, there is a restriction on a tree path structure. Hence, this study uses a general R-vine copula to eliminate the tree path structure restriction with the benefit of the relationship explanation of financial data. Various 13 bivariate copula functions are candidate for the R-vine copula model in this study but it is not limited to these because others can easily be added into the proposed algorithm. Besides, the proposed model is also compatible with the two special sub-classes of copulae (C-vine and D-vine) and can conducted with n-dimensional data analysis.

Further, this study first introduces mixture distribution to the R-vine model which gave the model even more flexibility. Mixture distribution consists of a Gaussian distribution and two generalised Pareto distributions in a univariate marginal intertemporal capital asset pricing and asymmetric GARCH model which also revealed more risk premium characteristics of a specific financial security. In addition, this study also provides a parallel R-vine with margins algorithm to a Bayesian inference and machine learning, bivariate copula selection in R-vine algorithm and an R-vine with margins data simulation algorithm. Note that the proposed algorithm can calculate the joint distribution for arbitrary R-vine copulae as well.

To conclude, the outcome of this paper generates an R-vine tree path dependence, pair-copula types and parameter estimate in n-dimensions. The univariate marginal model gives an individual stock risk premium and risk-free rate. Our proposed model can outperform the competing model. The outstanding proposed model is VBDA1-R-Vine-ICAPM-EGARCH-Mixture model. Besides, the study analytically found that traditional QMLE estimation method has significantly deteriorated in high dimensions. For further study, there are many possibilities depending on research interests. For example, the model can be investigated to a higher data frequency or with other copulae, e.g., a factor copula or a rotated copula. The copula selection condition could be changed rather than using $Kendall's\ \tau$, i.e., $Spearman\ \rho_s$ or other beneficial point of view in particular financial area. This research can be extended to other areas, e.g., the high portfolio optimisation problem. From a programming point of view, researchers may develop the algorithm in a super computer in order to mitigate the computational time in Bayesian inference and machine learning.

Chapter 5

Regular Vine Copula Model for Investment Portfolio Optimisation in Turbulent Financial Markets using Computational Intelligence

5.1 Abstract

This paper introduces the application of the artificial immune system and genetic algorithms in computational intelligence to investment portfolio strategy incorporated with sentiment analysis. The proposed forecasting model is a multivariate regular vine copula-based model with various candidates of 13 parametric bivariate copula functions. The proposed univariate marginal model is a hybridisation of intertemporal capital asset pricing and the asymmetric exponential generalised autoregressive conditional heteroscedasticity model, while the innovation part is a mixture of Gaussian distribution and two generalised Pareto distributions in extreme value theory, the so-called R-vine-ICAPM-EGARCH-Mixture model. The empirical experiments decompose into two main scenarios to address the different model performance measures on stock portfolios. Scenario 1 is the scenario of the new normal economic analysis according to the impact of the COVID-19 crisis and others, and is built for the risk-reward portfolio performance measure. Scenario 2 is built for the performance measure of diversity and convergence using data of the pre-COVID-19 pandemic and the investigation of the variation of dependence structure in the financial markets. Furthermore, this study also determines the portfolios with a class of cryptocurrency together with the traditional stock portfolios. The performance measure empirically proves that the proposed model outperforms the benchmark models in terms of risk-rewards and diversity indicators. This paper also applies the parallel computing technique to the proposed evolutionary algorithms to accelerate the solution convergence. The backtesting study in various portfolio strategies justifies the use of the proposed evolutionary algorithms and portfolio models with sentimental analysis. To be precise, stock selection in portfolio management is recommended for investment efficiency. Besides, paper discovers that there is a significant change in the financial market dependence structure according to the impact of the COVID-19 crisis and others, or, it is claimed that the financial market has been transformed and is in the state of the new normal.

Keywords: *Computational Intelligence, Optimal Portfolio Strategy, Sentiment Analysis, Multivariate Vine Copula-based Forecasting Model, Mixture Distribution, Cryptocurrency, COVID-19 New Normal Economic.*

5.2 Introduction

The problem of investment asset selection in portfolio is of importance to both corporate and individual investor. The main aim is to maximise profit at terminal time of investment based upon minimum investment and associated risk. Under various choice of financial products, investors want to know which securities and what proportion of the security should be put into the portfolio to obtain maximum wealth at the end of investment. That leads to the modern portfolio theory of Markowitz (1952) where Markowitz invented mean-variance portfolio optimisation (MVPO) and, since then, this pioneering work has rapidly developed and, is still in an active research area in many research perspectives. For instance, in recent research, see Puelz (2002), Chellathurai and Draviam (2005), Topaloglou, Vladimirou and Zenios (2008), Serban, Stefanescu and Ferrara (2013), Jeon and Park (2021), J. Du (2022), D. Xu et al. (2022) and J.-H. Du, Guo and Wang (2022).

Mathematically, the objective function in MVPO is to either minimise variance-covariance matrix, Σ_p, or maximise expected portfolio return, $E(r_p)$, subject to some realistic constraints, i.e., no short selling. In general, an efficient portfolio frontier can easily be obtained by partial derivative to the Σ_p with respect to the i^{th} asset proportion (W_i). A rational investor prefers a portfolio that returns the highest expected return given all associated risks, the so-called efficient mean-variance portfolio set, and a point that returns the lowest risk level is the so-called global minimum variance (GMV) of a portfolio. An optimal portfolio of Markowitz may, however, not hold if a rational investor wants to consider more extreme downside risks such as Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR). Hence, an alternative optimal portfolio problem has been proposed to improve Markowitz's MVPO, for example, the mean-VaR portfolio problem and the mean-CVaR portfolio problem (Consigli, 2002; B. Zhang, Wei, Yu, Lai & Peng, 2014; Q. Xu, Zhou, Jiang, Yu & Niu, 2016). Furthermore, in literature, there is another well-known optimisation problem (OP) which is a certainty equivalence tangency (CET) portfolio problem or Sharpe ratio (SR) optimisation problem which are more reasonable in practice since the CET optimisation problem considers the portfolio that gives the highest expected portfolio returns per unit of portfolio risk (Sharpe, 1963, 1994). Further mathematical discussion of these portfolio optimisation (POs) is provided in the next section. For the survey in POs problems, see, among others, Kolm et al. (2014); Pareek and Thakkar (2015); Y. Zhang, Li and Guo (2018); Kalayci et al. (2019); and Geboers, Depaire and Annaert (2022).

MVPO problems can be solved by exact solution techniques such as quadratic programming (Qin, 2015), nonlinear programming (Kırış & Ustun, 2012) or Lagrangian methods (Shaw, Liu & Kopman, 2008). However, when the problem is more complicated, for example, when the number of problem dimensions is increased, the exact solution technique may not be applicable since it is yielding the well-known non-deterministic polynomial-time hard (NP-hard problem) or non-convex optimisation problem, namely, it could potentially have many local minima. This therefore forces many researchers to propose an inexact solution technique, which is a huge and intensive computational based research area in PO problems, also known as the computational intelligence based meta-heuristic technique or machine learning algorithm. Computational intelligence is an approximation algorithm that was inspired from a metaphor of natural selection, for example, the genetic algorithm (Jalota & Thakur, 2018), particle swarm algorithm (Deng, Lin & Lo, 2012), Hopfield network (Hopfield, 1984), simulated annealing (T. J. Chang, Meade, Beasley & Sharaiha, 2000) and artificial immune system (Golmakani & Alishah, 2008).

In computational intelligence literature, one of the most well-known natural inspired algorithms are the class of genetic algorithm (GA) in evolutionary algorithms (EA). A GA is an intuitive algorithm, in short, there are five main steps including initial population, fitness function, selection, crossover, and mutation (R. E. Smith, Forrest & Perelson, 1993; Mitchell, 1998). Another well-known class of machine learning algorithms in computational intelligence literature is the class of the artificial immune system algorithm

(AIS), which is inspired by immunology. The clonal selection algorithm (CSA) is one of the most popular algorithms in AIS (De Castro & Von Zuben, 2002). The CSA concept is to solve the PO problems that use shape-space formalism and immunological terminology to describe cellular evolution and antigen-antibody (Ag-Ab) interaction. Note that, in programming, the encoding scheme of both GA and CSA is identical. Further discussion of both algorithms is demonstrated in the next section, where this study introduces CSA and GA algorithms to the proposed vine copula-based model. For surveys of exact and inexact solution techniques including the AIS and GA see Kalayci et al. (2019, Section 6), and the survey of the AIS including the CSA see Timmis et al. (2008).

In PO problems, forecasting asset returns is a crucial part since there are the main primary input to the PO problem. In the past decade, a class of multivariate copula models has been an active and popular model in the literature of forecasting returns in multivariate analysis since Sklar (1959) proposed copula theory. Copula properties can fulfil statistical properties of dependence structure of a pair of data through joint multivariate distribution in terms of univariate marginal distribution, and can also be extended to, in a general, high dimensional comovement structure. This is very important in the analysis of PO when there is a high number of assets to consider in investment.

In copula literature, the research focuses on the modeling of standard bivariate copulas, as well as on the family of elliptical and Archimedean copulas for multivariate copula modeling. There is a magnitude of works of standard copula in the literature, and for a recent study see Wichitaksorn et al. (2019) for elliptical copula and Górecki et al. (2021) for Archimedean copula. However, there are some limitations in standard copula, i.e., a restricted number of copula parameters. Therefore, this leads to another flexible, new class of copula models of Aas et al. (2009), also known as a vine copula model in multivariate analysis. A vine copula consists of a general regular vine (R-vine) and two special sub-classes including the canonical vine (C-vine) and drawable vine (D-vine). Again, in vine copula literature, research so far has concentrated solely on the two special sub-classes of vine copula. There are few studies in R-vine model due to the model's complexity and demand of the highly intensive computation technique (Dißmann et al., 2013; Müller & Czado, 2018, 2019).

To get up-to-date information on copula models, several studies including Fan and Patton (2014), Patton (2012), and Czado (2010) can be consulted. Notably, there have been few investigations on the use of the R-vine copula model in PO problems and computational intelligence. As a result, this study aims to fill this research gap by proposing a forecasting model based on the R-vine copula for PO and computational intelligence. In general, the financial time series exhibit volatility clustering, time varying heteroskedastic volatility and asymmetric-fat-tail behaviours (Engle, 2004). The family of the (generalised) autoregressive conditional heteroscedasticity ((G)ARCH) model appears to be one of the most popular models to capture these characteristics since it was introduced by Engle (1982) and Bollerslev (1986); and, since then, there have been many intensive research extensions. The asymmetric-GARCH family is one of the most well-known sub-families in this research area, such as LogGARCH (Geweke, 1986; Milhøj, 1987), EGARCH (Kolm et al., 2014), TGARCH (Zakoian, 1994), Power-GARCH (Bildirici & Ersin, 2009), GJR-GARCH (Glosten et al., 1993) etc. Hence, this study follows the well-known univariate EGARCH model to the R-vine copula model incorporated with the well-known intertemporal capital asset pricing model (ICAPM) of Merton (1973). While the innovation is a mixture distribution (Sahamkhadam et al., 2018; Caporale & Zekokh, 2019), it proves that it is an outperforming model (Khanthaporn & Wichitaksorn, 2022). The closest forecasting model in the optimisation problem to this study is in the study of Sahamkhadam et al. (2018) using GARCH-EVT with the standard copula model, while this study uses ICAPM-EGARCH-Mixture with R-vine and 13 candidate of bivariate copula functions. Hence, this is the second contribution that this study offers to the R-vine-ICAPM-EGARCH-Mixture model with 13 candidates of bivariate copulae to the PO problems and machine learning optimisation. No prior works have been found on the application of CSA algorithm to the R-vine copula-based with 13 candidates of bivariate copulae and the mentioning of a marginal model in n-dimensional investment portfolio management and review of its performance.

This leads to the research interests and contribution in the area of PO and machine learning using forecasting of the regular vine dependence model. This study also explores investment portfolio performance of the proposed model with the attractive asset class, a cryptocurrency that may (or may not) improve portfolio strategy among recent economic circumstances of the COVID-19 pandemic.

In addition, to measure convergence and diversity portfolio performance, this paper performs various empirical experiments to test the proposed models using the data of the pre-COVID-19 pandemic. Note that there are two main scenarios of the backtesting study where this paper uses the data in a different period such that the first scenario uses whole data availability and the second scenario uses data from only the pre-COVID-19 period. With these two different data uses in their scenarios, the purpose of the investigation is the variation of dependence structure in the financial markets according to the COVID-19 pandemic. Further discussion and literature of the proposed forecasting model and optimisation algorithm in computational intelligence can be seen in the next section.

The rest of this study is organised as follows. Section 5.3 introduces the portfolio selection problem and discusses the development of an R-vine copula-based model, as well as covering multivariate copula concepts. Section 5.4 presents the clonal selection algorithm and genetic algorithm in computational intelligence and their application to the portfolio problem and R-vine copula-based model. Section 5.5 analyses the empirical performance of the proposed dynamic portfolio strategies, and the conclusion and discussion are in Section 5.6.

5.3 Model

This section cover two main topics. First, Subsection 5.3.1 presents the development of PO problems and a copula-based model with a mixture distribution of financial returns, which is crucial for capturing the asymmetric-fat tail behaviour of financial time series data. Second, Subsection 5.3.2 provides an introduction to multivariate dependence copula models including standard copulae and vine copulae in graph theory.

5.3.1 Portfolio Optimisation Model

The modern portfolio theory of Markowitz provides the two important characteristics in MVPO which are the expected portfolio return, $E(r_p) = W'E(r_i)$, and the portfolio standard deviation $\sigma_p = \sqrt{W'\Omega W}$ (Markowitz, 1952), where the i is the i^{th} asset in the portfolio, W is a vector of a weight of asset and Ω is the variance-covariance matrix of asset returns in the portfolio. Markowitz initiates an efficient frontier of MVPO by differentiating Σ_p with respect to W_i. Markowitz also shows the portfolio with minimising portfolio variance (GMV portfolio) and the tangency portfolio, which is the most efficient portfolio. One important extension of MVPO is the Sharpe ratio (SR) PO, which is a benchmark PO of this paper because of the similarity of MVPO but the objective function considers the maximisation of $E(r_p)$ over a unit of σ_p, also known as the certainty equivalence tangency (CET) PO problem (Sharpe, 1963, 1994). To denote that n is the number of assets in a portfolio with expected asset return $\bar{r}_t = [\bar{r}_{t,1}, \bar{r}_{t,2}, ..., \bar{r}_{t,n}], i = 1, ..., n$, asset proportions $W_t = [w_{t,1}, w_{t,2}, ..., w_{t,n}]$ and Ω is an $n \times n$ matrix. The CET optimisation model can be formulated as:

$$
\begin{aligned}
\underset{W_t}{\text{maximise}} \quad & \frac{W_t'\bar{r}_t}{\sqrt{W_t'\Omega W_t}} && \text{Sharpe ratio} \\
\text{w.r.t.} \quad & \sum W_t' = 1, && \text{full investment} \\
& \forall i \in 1, 2, ..., n : \\
& w_{t,i} \geq 0, && \text{no short selling.}
\end{aligned} \tag{5.3.1}
$$

In the Quantitative Risk Management (QRM) literature, the two most popular downside risks are VaR and CVaR. VaR can be defined as the α^{th} percentile of loss distribution

L of portfolio allocation, where $\alpha \in (0,1)$ is the confidence level and $F(\cdot)$ is the cumulative distribution function of the random variable or

$$VaR_\alpha(L_{t,i}) \coloneqq q_\alpha(L_{t,i}) = \inf\{l \in \mathbb{R} : F_{t,i}(l) \geq \alpha\}.$$

While CVaR is the expectation of the tail of the loss distribution and above the α^{th} level of confidence, or

$$CVaR_\alpha(L_{t,i}) \coloneqq \frac{1}{1-\alpha}\int_\alpha^1 VaR_u(L_{t,i})\,du = E[L_{t,i}|L_{t,i} \geq VaR_\alpha(L_{t,i})].$$

For further concepts and definitions of VaR and CVaR in QRM see McNeil et al. (2015, chapter 2). The other two PO models in this paper consider these two downside portfolio risks which are similar to the minimise-VaR PO model (Consigli, 2002), the Omega-VaR ratio PO model (Sehgal, Sharma & Mansini, 2023) or the minimise-CVaR PO model (Rockafellar & Uryasev, 2000). Therefore, the considered PO models are as follows:

$$\begin{aligned}
&\underset{W_t,\alpha}{\text{maximise}} && f_\alpha(W_t) = \frac{W_t'\bar{r}_t}{W_t'VaR_\alpha(\bar{L}_t)} && \text{expected return/VaR} \\
&\text{w.r.t.} && \sum W_t' = 1, && \text{full investment} \\
& && \forall i \in 1,2,...,n: && \\
& && w_{t,i} \geq 0, && \text{no short selling;}
\end{aligned} \tag{5.3.2}$$

and

$$\begin{aligned}
&\underset{W_t,\alpha}{\text{maximise}} && f_\alpha(W_t) = \frac{W_t'\bar{r}_t}{W_t'CVaR_\alpha(\bar{L}_t)} && \text{expected return/CVaR} \\
&\text{w.r.t.} && \sum W_t' = 1, && \text{full investment} \\
& && \forall i \in 1,2,...,n: && \\
& && w_{t,i} \geq 0, && \text{no short selling.}
\end{aligned} \tag{5.3.3}$$

Once all inputs are set up for PO problems, the optimal portfolio strategy of optimisation model 5.3.2 and optimisation model 5.3.3 is chosen by the highest risk-adjusted portfolio return, which are similar ideas with the CET portfolio in model 5.3.1 or global minimum VaR portfolio and global minimum CVaR portfolio (Bodnar, Lindholm, Niklasson & Thorsén, 2022). More precisely, all PO models in the current paper are a class of risk and reward optimisation models where PO model 5.3.1 is determining a traditional deviation risk, PO model 5.3.2 is determining more conservative VaR risk, and PO model 5.3.3 is determining the most extreme CVaR risk.

5.3.2 R-vine Copula-Based Forecasting Model

This subsection presents the forecasting model of this study. The proposed forecasting model is used in PO models 5.3.1, 5.3.2 and 5.3.3, as discussed in Section 5.3.1. Additionally, the CSA and GA machine learning algorithms are presented in Section 5.4. The proposed forecasting model is a hybrid model that combines a well-known ICAPM model with asymmetric EGARCH model that uses flexible mixture innovation. The mixture innovation consists of three components, namely, two GPDs distribution in EVT for the important asymmetric-tail behaviour property of financial time series data (Engle, 2004) and Gaussian distribution for the central normal circumstances. See further details in Section 4.3.3. Then, it is extended to the multivariate case in portfolio problems by using an R-vine copula model.

Multivariate Copula Model

These days the copula dependence model of Sklar (1959) has been widely accepted and used in many research area such as climate and weather research (X. Li & Babovic, 2019), civil engineering (Lanzafame, Timmermans, Orlin, Valls & Nápoles, 2021), medicine (Popović, Genç & Domma, 2022), insurance (Deresa, Van Keilegom & Antonio, 2022) and, in particular, QRM (Bladt & McNeil, 2022; Neumeyer et al., 2019; Czado, 2013; Oh & Patton, 2018). A copula of Sklar is a multivariate distribution function, $C \sim Unif(0,1)$, where the marginals are $F_1(x_1), ..., F_n(x_n)$. The multivariate copula can be written as

$$F(x_1, ..., x_n) = C\{F_1(x_1), ..., F_n(x_n)\}, \tag{5.3.4}$$

for *n*-dimensional data. Note that $Unif$ is a uniform distribution. One of the most popular copula models in literature is a standard multivariate copula model which is in an elliptical copula family. The elliptical copula consists of the Gaussian copula and Student-*t* copula, respectively, as follows:

$$\begin{aligned} C_{\Sigma}^{Ga}(u) &= \Phi_{\Sigma}(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_n)), \\ C_{v,\Sigma}^{t}(u) &= t_{v,\Sigma}\left(t_v^{-1}(u_1), \ldots, t_v^{-1}(u_n)\right), \end{aligned} \tag{5.3.5}$$

where $\Phi^{-1}(\cdot)$ is an inverse Gaussian cdf. Σ is a $n \times n$ correlation matrix and Σ is equal to ρ, and when $d = 2$, it is bivariate copula. t^{-1} is an inverse Student-*t* cdf. $t_{v,\Sigma}$ is the cdf of multivariate Student-*t* distribution where $v > 2$ degrees of freedom.

Another popular standard copula in the QRM literature is a class of Archimedean copula model. There are one-parameter and two-parameter copulae in the class of Archimedean copula which represent various (a)symmetric tail behaviour. The one-parameter Archimedean copula models are, for example, Clayton, Frank and Gumbel copulae. The two-parameter Archimedean copula models are BB1-BB10 copulae. For all bivariate copulae in details, i.e., (a)symmetric tail behaviour, (conditional) cdf and pdf function, copula-based dependence measure and copula survival function, see Joe (2015, Chapter 4). Nevertheless, a standard copula model in multivariate analysis has a flexibility limitation (Aas et al., 2009). Particularly, the restriction is that there is only one single parameter controlling all pairs of comovement. It is even more inapplicable, especially, when the PO's problem is in high dimensions. This leads to a more complex and intensive computational family of copula model, the proposed regular vine copula model.

Regular Vine Copula Model

Since standard multivariate copula may be the lack of some flexibility. Essentially, when the number of data dimensions increases, the pair seem to require their own parameter(s) to represent the pair relationship, particularly, in financial data, where the pair relationship has its own (non)-linear co-movement.

Hence, the multivariate vine copula model has been invented to fill this gap by assigning one parametric copula function to each data pair see Joe (2015, Chapter 3). To be precise, the number of parameters in R-vine copula on n dimensions/nodes comparatively are massive and grows exponentially, at least $\frac{n(n+15)}{2}$ parameters including the model parameters of the proposed marginal model, or, the only R-vine copula parameter, the number of the parameter being $\frac{n!}{2}2^{\binom{n-2}{2}}$ (Morales Napoles et al., 2010). That makes an expensive cost of parameter estimation and far more model complexity than the standard multivariate copula. For the sub-class D-vine and C-vine copulae, the copula parameter is $n!/2$ number of parameters. Some discussion in copula parameters includes the example of a graphical tree of three-, four- and five-dimensions - see Aas et al. (2009).

In graph theory, vine copulae can be represented in a tree diagram. Mathematically, it is in the form of an *n*-by-*n* matrix. To construct the vine copula in matrix representation, in general, three matrices are required which are the matrices of copula types $\mathbf{T}$, parameter $\mathbf{P}$ and specification $\mathbf{M}$. Given a copula pdf function, $c(\cdot)$, and a marginal pdf function, $f(\cdot)$, a general R-vine copula can be formulated (Brechmann et al., 2012) and the R-vine pdf is discussed in Section 4.3.2.

For a different implementation, for example an upper triangular matrix - see Joe (2015, Chapter 3). In the literature of the PO problem, the work in copula has been in the central standard and two sub-classes of the C-vine and D-vine copulae with the traditional optimisation method. More precisely, an application of machine learning optimisation in the PO problem using an R-vine copula-based model has very few. Hence, this study initiates the forecasting of the R-vine copula-based model to the application of computational intelligence with various 13-candidate parametric bivariate copulae in order to obtain the most possible fitting copula function to the model and better capturing of the asymmetric tail behaviour of multivariate financial returns in the PO problem. The 13 copula functions used in the current study are Gaussian, Student-*t*, Frank, Clayton, Joe, Gumbel, Galambos, BB1, BB3, BB4, BB5, BB7 and BB10 copulae.

5.4 Algorithm

This section presents optimisation algorithms in computational intelligence and implementation steps for this study, including Subsection 5.4.1, which provides a brief explanation of the CSA and GA algorithms, and Subsection 5.4.2, which outlines the procedure of optimal portfolio selection.

5.4.1 Computational Intelligence

Clonal Selection Algorithm

This section summarises the CSA algorithm, which is a class of immune algorithms in machine learning, based on the original paper of De Castro and Von Zuben (2002). The CSA algorithm is inspired by the clonal selection theory in the immune system (Burnet, 1959). The clonal selection theory is when the body is infected from a bacterium or invader, an antigen (Ag). An immune mechanism of defence will automatically activate by producing antibodies (Ab) from some subpopulation of bone marrow derived cells (B lymphocytes) to go against an Ag. However, it appears that partial Ab plays an active role in the immune mechanism due to the production of highly diverse populations in natural variation. In fact, by the clonal selection principle and the affinity maturation process, the only Ab that successfully bind with the Ag will be (i) maintenance as memory cells; (ii) selection and cloning of the most stimulated individuals (otherwise death); (iii) affinity and maturation and re-selection of the higher affinity clones; (iv) generation and maintenance of diversity; and (v) hypermutation proportional to the cell's affinity. For further discussion of the clonal selection principle and the affinity maturation process see De Castro and Timmis (2002). In the machine learning procedure, the CSA algorithm is as follows:

(a) Generate a candidate set of an initial population (P), where the component of the remaining population is P_r and memory cells are M;

(b) Determine an affinity maturation process by selecting the n best population P_n;

(c) Reproduce (or Clone) these P_n populations, by creating a temporary population (T) where the closest size is an increasing function of the affinity measure of the Ag;

(d) Perform a hypermutation scheme to the current population, T, where the hypermutation is a proportion of the Ab affinity maturation to generate a hyper-maturated Ab population (T^*);

(e) Re-select individually the improved T^* to the memory population set (P);

(f) Maintain diversity by replacing d low affinity of the Ab.

Note that specification and further discussion of the CSA algorithm to solve the PO problems are discussed in Section 5.5.3. For additional discussion of the CSA algorithm see Bin Shalan and Ykhlef (2015), and for its flow chart see García-Mejía et al. (2021). The recent survey of an algorithm class of AIS including the CSA algorithm see, among others, L. Li, Lin and Ming (2022). The latest development of AIS can be found in the ICARIS conference (the International Conference on Artificial Immune Systems) and it is worth mentioning that the CSA algorithm in the PO problems is still scarce in literature, especially, the CSA application to the forecasting model of the R-vine and conditional volatility with mixture innovation in the PO problems.

Genetic Algorithm

The GA algorithm is inspired by Charles Darwin's theory of natural evolution. The theory of natural evolution is, in brief, to a better understanding of the GA algorithm as follows. In nature, animals compete with each other in order to obtain virtual resources, for instance, food and so on. In the same species of population, individuals compete to attract others to survive. While the best performing individual will be selected, or vice versa, the least performing individuals will not be selected and are likely to have a lower chance of survival. This is also called a reproduction process (or selection process), which means that the most fit individuals produce a relatively large number of offspring. Due to the reproduction process, a recombination of the good characteristics of each ancestor can be generated as the best fit offspring whose fitness may be better than the parent. After a few generation, species evolve spontaneously and adapt to their environment. For further biological background of the GA algorithm - see Chapter 2 in Sivanandam and Deepa (2008).

The GA algorithm consists of four major components such as repertoire, diversity, variation, and natural selection, where each step plays a crucial mechanism in the solution of the PO problems. In the literature, genetic algorithms (GAs) are one of a huge class of optimisation algorithms in evolutionary computation, which were first proposed by Holland (1995). Then, Holland's theory rapidly evolved and nowadays the GAs stand up as a powerful algorithm, in particular, the PO problems. See specimens of the development of the GA algorithm in the PO problems literature, among others, in Forrest and Mitchell (1993), Harik, Lobo and Goldberg (1999), Deb, Pratap, Agarwal and Meyarivan (2002), Q. Zhang and Li (2007), Malossini, Blanzieri and Calarco (2008), T.-J. Chang, Yang and Chang (2009), Zhu, Wang, Wang and Chen (2011), Cao, Recknagel and Orr (2014), Liu and Zhang (2015), Corus and Oliveto (2018), Ding, Chen and Zhong (2019) and Butt et al. (2021). In general, the GA procedure is as follows:

(a) Initialisation: This process begins with the simulation of a population. Each individual in a population is a specific solution to the PO's problem. To be precise, a set of parameters (genes) is randomly generated. The set of genes is therefore a set of binary string (0 or 1) also known as the set of solution or chromosome.

(b) Fitness function: Fitness score for each chromosome is computed according to the objective function. The top fitting chromosomes at some thresholds are then selected for subsequent operations, for instance, maximising or minimising the objective function.

(c) Reproduction (of Selection): This process is a reproduction of the new chromosomes (the next generation). On the one hand, the higher the fitness score, the more chance to be selected with higher probability, P^{Top}. On the other hand, the lower the fitness score, the less chance to be selected with lower probability, P^{Tail}, this is still necessary selection for the sake of maintenance of diversity.

(d) Crossover: This step is the most significant step, where each pair of parent chromosomes is randomly exchanged at the crossover probability, $P^{Crossover}$. It occurs at the genes level (a binary level). The new chromosome is then created, and is called an offspring process.

(e) Mutation: The final step is the mutation of the chromosomes. The gene of offspring chromosomes will be mutated at the mutation probability, $P^{Mutation}$.

The details of the GA algorithms are seen, among others with Deb (2001) and T.-J. Chang et al. (2009). For further surveys in evolutionary algorithms including the GA see Xue, Zhang, Browne and Yao (2016) and Wei, Wang, Zhong, Liu and Zhang (2022). Note that those steps of the GA and CSA are repeated as a machine learning process at some stopping conditions, i.e., learning curve or tolerance. Note also that this paper initiates the parallel computing technique to the CSA and GA algorithms to mitigate the computational time. It is now worth discussing the similarity of the GA and CSA biological algorithm in computational intelligence. The above description of both biological backgrounds of the GA and the CSA are very distinguished, however, in computational intelligence, they are a neighbour algorithm which involves a random process of binary string exchanging selection by the objective function and transformation at some probability. Again, a particular specification of the GA in the proposed PO problems is discussed in the empirical experiments of Section 5.5.

There are no prior works in the PO problem literature on performance measures in the application of the GA and the CSA with the parallel computing technique to the proposed R-vine based forecasting model and the massive 13 candidates of bivariate copula function, which are described in Section 5.3.2. In addition, this study analyses the PO problems in the interesting asset class of cryptocurrency into various optimisation models to observe the recent performance during the new normal economic of the COVID-19 pandemic and the pre-COVID-19 economic.

5.4.2 Implementation Step

This section presents the steps for the proposed dynamic portfolio trading strategies along with various benchmark dynamic portfolio trading strategies of stock portfolios with (and without) cryptocurrency in the portfolios. The main step is the procedure of the assets forecasting return as the author proposes an R-vine copula and ICAPM-EGARCH-Mixture based model. The algorithm is a well-known process which is referred to as the Inference for the Margins method (IFM) (Al Janabi et al., 2017), then it plug-in the forecasting returns to the PO models 5.3.1, 5.3.2 and 5.3.3.

To begin with, let T = $[t_1, ..., t_\mathbf{T}]$ be the length of the backtesting period, t_0 = the starting point of trading day, S = the number of Monte Carlo simulations from the multivariate return distribution for each iteration between the backtesting period (T) in the maximum likelihood parameter estimation method.

The first forecasting model is the proposed (**m1**) R-vine-ICAPM-EGARCH(1,1,1) with the mixture innovation model[1] and the other two forecasting models being competing

[1] There are 13 bivariate copula candidates in the R-vine model.

models including (**m2**) Standard-Student-*t*-Copula-AR(1)-GARCH(1,1)-EVT model[2] of Bruhn and Ernst (2022) and (**m3**) R-vine-ARMA(1,1)-GARCH(1,1) with Student-*t* innovation model[3]. The whole procedure can be briefly summarised as follows:

a. *Data standardisation*: use the ICAPM-EGARCH(1,1,1)-Mixture model in Sections 4.3.3 and competing models (AR(1)-GARCH(1,1)-EVT model and ARMA(1,1)-GARCH(1,1)-Student-*t* model) to estimate univariate marginal model parameters from the training data set, $\mathbf{r}_t = [r_{t,1}, ..., r_{t,n}]$, $r_{t,j}$ where $j = \{1, .., n\}, t \in S_r$ where standardised residual $\hat{\mathbf{z}}_t$ is

$$\hat{\mathbf{z}}_t = [\hat{z}_{t,1}, ..., \hat{z}_{t,n}], \hat{z}_{t,j} \approx iid. \tag{5.4.1}$$

b. *Distribution function estimation*: use the estimated standardised residual vector $\hat{z}_t$ to obtain

$$\hat{\mathbf{u}}_{t,n} = \hat{F}_n(\hat{z}_{t,n}), \hat{\mathbf{u}}_{t,n} \sim Unif(0,1). \tag{5.4.2}$$

c. *Multivariate copula parameter estimation*, Matrix $\hat{\Omega}$: apply Sklar's theorem in equation 5.3.4 by inserting uniform $\hat{\mathbf{u}}$ in a chosen multivariate copula of the particular forecasting model:

$$\hat{F}_n(\hat{\mathbf{u}}_{t,n}) = \hat{C}(\hat{F}_1(\hat{u}_{t,1}), \hat{F}_2(\hat{u}_{t,2}), ..., \hat{F}_n(\hat{u}_{t,n})|\hat{\Omega}) \tag{5.4.3}$$

where

c.1 Model **m1** applies R-vine copula function in equation 4.3.5 to obtain its R-vine parameters, hence $\hat{\Omega} = \hat{\mathbf{P}}_{m1}$ and matrix of copula types $\hat{\mathbf{T}}_{m1}$ and matrix of copula specification $\hat{\mathbf{M}}_{m1}$,

[2]This model could be referred to as Statistics and Machine Learning Toolbox in MATLAB where it is also the popular model in literature in QRM.

[3]The model is referred to the study of Goel and Mehra (2021), however, there are only five candidates of bivariate copula which are Gumbel, Frank, Clayton, Gaussian and Joe. While the study has 13 bivariate copula functions for the candidate in R-vine copula with the aim of capturing all possible (a)symmetrical dependence properties of multivariate financial returns

c.2 Model **m2** applies the standard Student-*t* copula function in equation 5.3.5 to estimate its standard Student-*t* parameter, hence, $\hat{\Omega} = \{\hat{\upsilon}_{m2}, \hat{\Sigma}_{m2}\}$ and

c.3 Model **m3** applied the R-vine copula function in equation 4.3.5 again and then obtained its R-vine parameters, types and specifications which are $\hat{\mathbf{P}}_{m3}, \hat{\mathbf{T}}_{m3}$ and $\hat{\mathbf{M}}_{m3}$, respectively.

d. *Estimated multivariate dependency structure*: let S be a uniform random number, simulate $M = [m_{s,t,1}, m_{s,t,2}, ..., m_{s,t,n}]$ where $s = 1,..,S$ and $t = 1,...,T$, then insert M into the estimated empirical multivariate copula distribution of step c. given by each model $\hat{\Omega}$ and obtained

$$\hat{\mathbf{v}}_{s,t,n} = [\hat{\upsilon}_{s,t,1}, \hat{\upsilon}_{s,t,2}, ..., \hat{\upsilon}_{s,t,n}] = \hat{C}(\hat{F}_1(m_{s,t,1}), \hat{F}_2(m_{s,t,2}), ..., \hat{F}_n(m_{s,t,n})|\hat{\Omega}), \hat{\upsilon} \sim Unif(0,1). \tag{5.4.4}$$

Note that $\hat{\mathbf{v}}$ required an additional $\hat{\mathbf{T}}$ and $\hat{\mathbf{M}}$ if it is model **m1** and model **m3**.

e. *Calculation of the inverse of the estimated marginal cdf function*: use $\hat{\mathbf{v}}$ of step d. in order to obtain the estimated new standardised residuals for dynamic PO models:

$$\hat{\mathbf{Z}} = [\hat{z}_{s,t,1}, \hat{z}_{s,t,2}, ..., \hat{z}_{s,t,n}] = [\hat{F}_1^{-1}(\hat{\upsilon}_{s,t,1}), \hat{F}_2^{-1}(\hat{\upsilon}_{s,t,2}), ..., \hat{F}_n^{-1}(\hat{\upsilon}_{s,t,n})]. \tag{5.4.5}$$

f. *Forecasting of returns*: use $\hat{\mathbf{Z}}$ of each model to calculate forecast return $\hat{r}_t$ for $\forall n$ assets at all backtesting time periods, $t = [t_0 + 1, ..., t_0 + T]$ by each univariate marginal model. Hence, at one-step return forecasts, a particular estimated return obtains given by the marginal model and its parameters,

$$\hat{\mathbf{r}}_t = [\hat{r}_{t,1}, \hat{r}_{t,2}, ..., \hat{r}_{t,n}]. \tag{5.4.6}$$

g. *Optimisation methods*: perform daily PO problems with machine learning algorithms which are explained in Section 5.4.1 and also the benchmark algorithm to PO models 5.3.1, 5.3.2 and 5.3.3 using $\hat{\mathbf{r}}_t$ in order to obtain optimal weights $\mathbf{W}_t = [w_{t_0+1}, ..., w_{t_0+T}]$ of the backtesting period. Note further details of prior stock selection perform this step - see Section 5.5.2 stock selection.

h. *Optimal weights model comparisons*: calculate real optimal portfolio risk and return and other performance indicators using optimal asset weights of vector $\mathbf{W}_t$ and real asset returns $\mathbf{r}_t$. Empirical one-period portfolio return is $R_t = \sum_{i=1}^{n} w_{i,t} r_{i,t} = \mathbf{w}_t^{'} \mathbf{r}_t$

Note that the parallel computing technique is plugged in to all possible steps that required intensive computing such as in *step a.* of data standardisation, *step c.* of multivariate copula parameter estimation and *step g.* of optimisation methods in computational intelligence. For further discussion of the proposed forecasting R-vine-based model in PO models see Chapter 4.

5.5 Empirical Optimal Portfolio Strategy

This section presents backtesting results of the proposed portfolio strategies performance in two main classes of financial products: stock and cryptocurrency in up to 35 dimensions. There are two main scenarios in the empirical experiments with the same PO models in Section 5.3.1 and forecasting models in Section 5.3.2. The main aim of these scenarios is to measure the three main performances such as the risk-reward, convergence, and diversity indicators. Overall, the aim of Scenario 1 is the performance measure of the risk-reward indicator for the proposed portfolio strategies among the current global economic crisis due to the COVID-19 pandemic (and others) (World Bank, 2020), while the aim of Scenario 2 is the performance measure of diversity-based and convergence-based indicators using the data before the COVID-19 pandemic. The details of these two scenarios and data descriptive statistic are provided in the next section. Then, Subsection 5.5.2 is fundamental analysis in PO problems with the section ending with 5.5.3 backtesting results against the two scenarios.

5.5.1 Data and Scenario

The data in this paper were spit into two parts: the first was a training set and the second was a backtesting set in two scenarios. All data were retrieved via the MATLAB data feed function from Yahoo Finance. There were two scenarios in which the aim was to analyse the proposed portfolio strategy performance in three perspective. Two daily financial asset classes were used in these experiments which included up to 30 stocks and five cryptocurrencies. For the training set, stock data were the adjusted price of 30 stocks of the Nasdaq stock market in the top 100 market capitalisation[4], while cryptocurrency data were the top five market capitalisations in the cryptocurrency market[5].

In the literature, the portfolio strategy consisted of different models and algorithms. The portfolio strategy performance measures can be categorised in four main performance measures of metrics such as the indicators of reward/risk return, convergence, diversity, and hybridisation Kalayci et al. (2019, Section 5). This paper therefore follows this idea in performance measures and performed the backtesting experiments of the two main scenarios as follows:

> Scenario 1: Reward and risk indicator in portfolio strategy for the new normal of the financial Markets. This scenario aims to find an optimal portfolio weight allocation for the current and ongoing COVID-19 pandemic using all possible data availability on the cryptocurrency market, where it was between 10 November 2017 and 20 September 2021 (1,411 observations). The stock data used was between 05 January 2009 and 20 September 2021 (3,200 observations). The stocks were selected after 2008 to avoid the effect of the global financial crisis. For the

[4]The list of the 30 stocks chosen is: HDB, BMY, AMT, SBUX, BLK, SCHW, UNP, BBL, AXP, GS, RTX, AMD, BA, AMAT, AMGN, ISRG, IBM, SNY, TGT, TD, TTE, EL, CVS, GE, DEO, SPGI, RIO, JBHT, DE and CAT. See details in Appendix C

[5]The list of the top five cryptocurrencies is: BTC, ETH, USDT, BNB and XRP. See details in Appendix C

backtesting set, both the stock and cryptocurrency data were tested using the data that were between 21 September 2021 to 12 October 2022 (268 observations)[6].

Scenario 2: Diversity and convergence indicators. This scenario was an experiment of the performance indicator of diversity and convergence using pre-COVID-19 data. Both the stock and cryptocurrency for the training period were between 10 November 2017 and 07 November2018 (250 observations), while the backtesting period was between 08 November 2018 and 31 October 2019 (246 observations) [7]. For a full list of all securities see Appendix C.1.

Table 5.1 shows descriptive statistics in brief and can be summarised as follows. Total market capitalisation of all selected assets in analysis is approximately US$4,048 billion. This includes stock market capitalisation of US$3,470 billion and cryptocurrency market capitalisation of approximately US$578 billion as of 11 November 2022. The 30 selected stock business sectors comprise Finance (6 firms), Health Care (5 firms), Capital Goods (5 firms), Technology (3 firms), Consumer Non-Durable (2 firms), Energy (2 firms), Transportation (2 firms), Unidentified Sector (2 firms), Basic Industries (1 firm), Capital Goods (1 firm), Consumer Durables (1 firm) and Consumer Services (1 firm).

For all stocks, the average returns are positive and the average return range is [0.03%, 0.184%]. The range of standard deviation is [1.311%, 3.679%]. The range of skewness is [-1.000%, 1.302%], and the range of kurtosis is [5.952%, 23.818%]. This indicates that all stock returns are asymmetrical and fat-tailed distribution it confirmed by the Jarque-Bera test statistic value of all stock returns that it rejects the null hypothesis that comes from a Gaussian distribution at the 1% significance level.

For all cryptocurrencies, average returns are also positive and the return range is [0%, 0.006%]. The range of standard deviation is [0.005%, 0.728%]. The range of skewness is [-0.273%, 2.619%], the range of kurtosis is [8.552%, 37.475%]. It indicates that all cryptocurrencies are asymmetrical and fat-tailed distribution and are confirmed

[6] The training/backtesting ratio for cryptocurrency data is 84/16. The training/backtesting ratio for stock data is 92/8.

[7] The training/backtesting ratio for both cryptocurrency data and stock data is 50/50.

Table 5.1: Stock Business Sector in Analysis and Descriptive Statistics including Cryptocurrency

Stock Business Sector	**No. Firm**	
Basic Industries	1	
Capital Goods	5	
Consumer Durables	1	
Consumer Non-Durables	2	
Consumer Services	1	
Energy	2	
Finance	6	
Health Care	5	
Technology	3	
Transportation	2	
Unidentified Sector	2	
All Stocks Returns, %	**Min**	**Max**
Average	0.03	0.18
Standard Deviation	1.31	3.68
Skewness	-1.00	1.30
Kurtosis	5.95	23.82
All Cryptocurrencies Returns, %	**Min**	**Max**
Average	0.00	0.01
Standard Deviation	0.01	0.73
Skewness	-0.27	2.62
Kurtosis	8.55	37.48

by the Jarque-Bera test statistic value of all cryptocurrency returns that it rejects the null hypothesis that comes from a Gaussian distribution at the 1% significance level. It is worth mentioning that in Scenario 1, during the backtesting period, the study could see that the stocks seemed to perform better than the cryptocurrencies. However, it is interesting to study the quantitative empirical experiment and confirm that the optimal portfolio strategy and portfolio of stock without cryptocurrency can beat optimal portfolio strategy of stock with cryptocurrency on the current COVID-19 situation.

5.5.2 Stock Selection

This section presents fundamental analysis in portfolio management where there is stock selection. Stock selection is an essential key in portfolio selection during market turmoil, especially, the current economic crisis of the COVID-19 pandemic and others. In a fund manager's point of view, stock selection has to be considered in order to choose the most plausible stock through their research and experiences, since there is such a large amount of stock available in the market, and it is clear that a small amount of stock in the OP problem would be more convenient and efficient for the solution of optimal portfolio weight (Nti, Adekoya & Weyori, 2020). Many qualitative and quantitative indicators can be determined to choose stock assets such as financial ratios, the real value of the organisation, forecasted stock returns, technical analysis, and sentiment analysis (Fazel Zarandi, Rezaee, Turksen and Neshat (2009), Kamble (2017), Shen and Tzeng (2015) and Feuerriegel and Gordon (2018)). Since the COVID-19 pandemic, many countries announced a lockdown policy with a restriction of social activities to stop the spread of the Coronavirus. Due to the lockdown, many businesses were disrupted; however, other businesses took off because customer behaviour was forced to change. It is a positive (or a negative) factor to the business. Consequently, it either gains (or loses) to the stock values.

Therefore, this paper chose economy sentiment analysis for stock selection to the PO problems since it could foresee which business sector could potentially gain according to the COVID-19 pandemic. Therefore, three business sectors including the sector of technology, health care and energy were selected for the PO problems. Along with sentiment analysis, the empirical results can be seen in the Scenario 1 reward and risk indicator in Section 5.5.3. For a further literature review of fundamental analysis see Andriosopoulos, Doumpos, Pardalos and Zopounidis (2019).

5.5.3 Portfolio Performance and Findings

This section presents the backtesting results of the proposed portfolio strategies and methodology for optimal investment asset weight. There are three main panels of portfolio strategies in each scenario: the two main panels present the CSA and GA optimisation algorithm with various proposed optimisation models, and another panel presenting the benchmark models.

Note that model **m3** is referred to in the paper of Goel and Mehra (2021); however, Goel and Mehra (2021) offered only five candidates of bivariate copula function, while the current paper offers a magnitude of 13 bivariate copula functions based on copula literature in QRM, as discussed in Section 5.4.2.

For each scenario, the details of the three portfolio strategies are split into three panels or 18 portfolio strategies as follows:

Panel A portfolio strategy is a representative of all competing models and algorithms. The benchmark portfolio strategies consist of: 1) the optimal weights solution of all stocks and cryptocurrencies (total 35 assets); and 2) the optimal weights solution of all stocks (total 30 assets). The benchmark forecasting model and optimisation problem are popular in the literature including two forecasting models such as the Standard-Student-*t*-copula-AR(1)-GARCH(1,1)-EVT model (Bruhn & Ernst, 2022) and the R-vine-ARMA(1,1)-GARCH(1,1)-Student-*t* model (Goel & Mehra, 2021). The benchmark portfolio objective is to minimise CVaR portfolio risk and the selected optimal portfolio strategy is by the highest reward and CVaR risk ratio. See Section 5.4.2 for further discussion, as well as the benchmark equally-weighted portfolio strategy (EQW). Thus, there are six benchmark portfolio strategies.

Panel B presents mainly the CSA optimisation algorithm in Section 5.4.1, where the forecasting model is the proposed R-vine-ICAPM-EGARCH(1,1,1)-Mixture in Section 5.3.2 with stock selection analysis, and where analysis was discussed in

Section 5.5.2. Again, there are two asset classes including stocks with sentiment analysis and cryptocurrency where the asset class in the portfolios is either both classes or only selected stocks. Then, the optimal weights solution could be explored by the proposed portfolio models in Section 5.3.1 and CSA algorithm. Thus, this panel has six optimal portfolio weight solutions.

Panel C is similar to Panel B. but, the main difference is that the panel considers the different machine learning optimisation method - the GA algorithm. Thus, again, this panel has six optimal portfolio weight solutions in the GA algorithm.

In the scenario experiments, the number of dimensions in the portfolio were up to 35 assets and in an R-vine copula-based model in Panels B and C, there are 595 pair copula functions. Hence, 1,190 parameters must be estimated due to 13 candidate copula functions of candidates in the R-vine copula model. A maximum spanning tree algorithm was conducted in order to obtain the best fitting copula candidate (Dißmann et al., 2013).

Table 5.2 presents an empirical decomposition of a multivariate R-vine copula-based model in both Scenario 1 and Scenario 2. It confirmed that the dependence structure of the financial market has statistically changed according to the COVID-19 disruption. In particular, the co-movement of a set of data in the analysis has been changed to a behaviour of asymmetric and tail dependence, since, one of the top five fitting copula has been replaced by Galambos and B5/Joe copulae from BB10 copula. Furthermore, since the COVID-19 crisis and others, market co-movement is more likely to present a tails dependency according to the shift of the second best fitting copula (Replaced Gaussian copula by Student-*t* copula). The best fitting copula between the pre-COVID-19 and COVID-19 period is still the same copula (Frank copula). See Table 5.2 for the empirical results of the decomposition of multivariate R-vine copula for Scenario 1 and 2 by a maximum spanning tree algorithm.

In the experiments, the initial setting up for both CSA and GA algorithms was as follows: the Hamming shape-space with binary string was employed to represent a real optimal weight, while over the range of the long position, $w_{t,i} \in [0, 1]$. The

Table 5.2: Empirical Multivariate R-vine Copula Decomposition into a Cascade of Pair Copulae of the Proposed and Competing Forecasting Model by Scenario

Scenario	1				2			
Forecasting Model	*MD1*		*MD2*		*MD1*		*MD2*	
Dimension	35	30	35	30	35	30	35	30
Total Pair-Copula	595	435	595	435	595	435	595	435
Total Parameter Estimation	779	643	779	606	701	534	709	519
Gaussian	81	39	89	44	115	71	122	91
Student-*t*	86	126	118	118	43	49	70	55
Frank	193	136	207	178	179	113	171	113
Clayton	78	32	59	24	122	95	91	71
Gumbel	13	12	18	6	14	15	17	22
B5/Joe	25	7	16	6	35	22	36	29
Galambos	23	1	22	6	37	36	44	25
BB1	6	6	1	3	3	1	3	2
BB3	6	3	2	0	6	3	0	0
BB4	9	1	1	0	3	2	3	4
BB6	0	0	0	0	0	0	0	0
BB7	14	13	4	1	2	2	3	4
BB10	61	59	58	49	36	26	35	19

Note that *MD1* is R-vine-ICAPM-EGARCH(1,1,1)-Mixture and *MD2* is R-vine-ARMA(1,1)-GARCH(1,1)-Student-*t*.

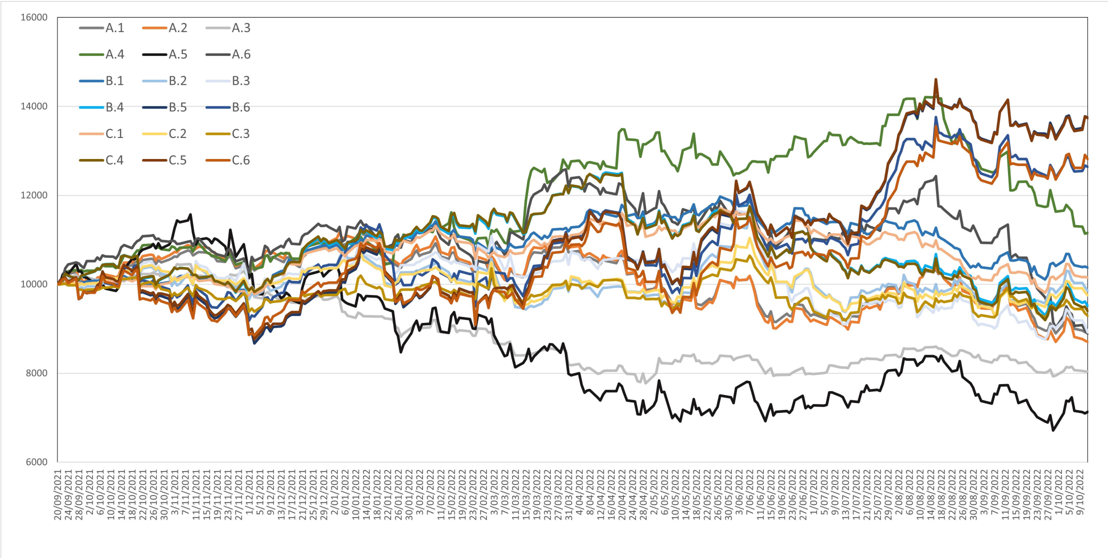
selected bitstring length was $L = 66$, corresponding to 18 decimal places of precision. In the CSA algorithm of both scenarios, the study set population size = 100, mutation probability = 0.010, number of clones per candidate = 10, tolerance = 1.0e-03, break event = 2, except for a maximum number of generations = 90 and 300 for Scenario 1 and Scenario 2, respectively. In the GA algorithm of both scenarios, the study set the number of chromosomes = 1000, crossover probability = 0.500, mutation probability = 0.030, tolerance = 1.0e-05, break event = 2 and maximum number of generations = [300,200,100] for PO models 5.3.3, 5.3.2 and 5.3.1, respectively. Note that the higher the tolerance, the more computational time. The number of chromosomes in the GA or population size in the CSA also significantly increases the computational time.

Reward and Risk Indicator

This section presents analysis of the reward and risk performance of the portfolio strategy in Scenario 1, the optimal portfolios among the new normal economic. Table 5.3 summarises performance metrics of the reward and risk indicator of Scenario 1. Overall, all risk-adjusted reward indicators are fundamentally reasonable in the rational investment perspective, and the proposed strategies in Panel B and Panel C outperform the competing models in Panel A. The optimal portfolio weights of stocks without cryptocurrencies are likely to perform better than portfolios with both two asset classes. This is an unsurprising result due to the crash of the cryptocurrency market in the past few years. In general, the GA algorithm is likely to have more outstanding risk and reward indicators than the CSA algorithm according to more number in the top five best indicators and a much faster convergence to the optimal solution. However, if the ultimate investing goal is portfolio wealth, the CSA algorithm seems to be the best portfolio strategy (portfolio strategy B.5). It is clear that PO models 5.3.2 and 5.3.3 can beat benchmark CET model 5.3.1 given by the indicators of average return (Avg. Return), standard deviation (Std. Deviation), Sortino Ratio, Sharpe Ratio, STARR Ratio, Mean/VaR and cumulative portfolio wealth, except for CVaR portfolio risk and VaR portfolio risk.

With the ultimate goal of portfolio management, a fund manager may be interested in profit and loss at the end of an investment period where, supposing an initial investment at day one is \$10,000, the highest empirical cumulative portfolio wealth model is model (B.5) with \$13,748 (34.57% p.a.). The second highest empirical cumulative portfolio wealth belongs to model (C.5) with \$13,738 (34.48% p.a.). The 3^{rd}, 4^{th} and 5^{th} highest empirical cumulative portfolio wealth are model (C.6) with \$12,816 (26.05% p.a.), model (B.6) with \$12,642 (24.45% p.a.) and model (A.4) with \$11,155 (10.73% p.a.), respectively. See Figure 5.1 for all portfolio strategies including the benchmark results of empirical cumulative portfolio wealth during the backtesting period in Scenario 1 (the New Normal Economy).

Figure 5.1: Scenario 1: An Empirical Cumulative Portfolio Wealth of All Portfolio Strategies Including the Benchmark Portfolio Strategy of A.1-A.6, B.1-B.6 and C.1-C.6



Note that the investment is $10,000 at time t_0. The investment portfolio of backtesting period is 268 days (between September 09, 2021 and October 12, 2022).

In the new normal economy, according to the COVID-19 disruption, the CSA approaches display better risk and reward performance than the GA approaches, whereas the computational time of the CSA approaches are not as fast as the GA approaches where the results are in harmony with the literature (Fernández & Gómez, 2007).

Table 5.3: Scenario 1: The Performance Measure of Portfolio Strategy among the Reward and Risk Indicator

Portfolio Strategy	Avg. Return (%)	Std. (%)	Sortino Ratio	Sharpe Ratio	CVaR (%)	VaR (%)	STARR Ratio	Mean /VaR	Avg. Turnover	Portfolio Wealth	Time (Sec.)
Panel A: Benchmark Portfolio											
EQW Portfolios											
(A.1) Stock and Cryptocurrency	-0.037	**1.158**	-1.435	-0.960	**3.332**	**3.176**	-0.011	-0.012	3.191	8,903	n.a.
(A.2) Stock	-0.043	1.349	-1.238	-0.829	3.876	3.688	-0.011	-0.012	3.177	8,707	n.a.
Asset Class: Stock and Cryptocurrency											
(A.3) Min CVaR with Model **m2**	-0.076	**0.977**	-1.417	-1.179	3.915	3.510	-0.020	-0.022	0.067	8,042	4
(A.4) Min CVaR with Model **m3**	**0.052**	1.476	-0.920	-0.694	4.869	3.703	**0.011**	**0.014**	0.074	**11,155**	2
Asset Class: Stock											
(A.5) Min CVaR with Model **m2**	-0.103	2.144	**-0.781**	**-0.550**	6.338	5.518	-0.016	-0.019	0.092	7,135	5
(A.6) Min CVaR with Model **m3**	-0.032	1.541	-0.952	-0.719	6.058	5.217	-0.005	-0.006	0.076	8,894	3
Panel B: CSA Optimisation Algorithm											
Asset Class: Stock and Cryptocurrency											
(B.1) Max SR	0.018	**0.878**	-1.796	-1.206	**2.654**	**2.275**	0.007	0.008	**0.033**	10,374	61
(B.2) Max Expected Return/CVaR	0.004	1.424	-1.139	-0.753	4.329	4.050	0.001	0.001	0.045	9,850	73
(B.3) Max Expected Return/VaR	-0.033	1.377	-1.154	-0.805	3.893	3.795	-0.009	-0.009	0.046	8,920	50
Asset Class: Stock											
(B.4) Max SR	-0.010	1.344	-1.148	-0.807	4.557	4.069	-0.002	-0.002	0.038	9,515	20
(B.5) Max Expected Return/CVaR	**0.136**	1.858	**-0.726**	**-0.506**	5.878	4.619	**0.023**	**0.029**	**0.037**	**13,748**	38
(B.6) Max Expected Return/VaR	**0.103**	1.786	**-0.779**	**-0.544**	5.562	4.608	**0.019**	**0.022**	0.038	**12,642**	27
Panel C: GA Optimisation Algorithm											
Asset Class: Stock and Cryptocurrency											
(C.1) Max SR	0.009	**0.887**	-1.757	-1.203	**2.736**	**2.560**	0.003	0.004	**0.033**	10,147	7
(C.2) Max Expected Return/CVaR	-0.003	1.166	-1.428	-0.925	**3.243**	**2.739**	-0.001	-0.001	0.044	9,753	16
(C.3) Max Expected Return/VaR	-0.021	**1.137**	-1.498	-0.964	**3.421**	**3.242**	-0.006	-0.006	**0.022**	9,292	11
Asset Class: Stock											
(C.4) Max SR	-0.014	1.342	-1.167	-0.812	4.533	4.079	-0.003	-0.003	0.038	9,400	5
(C.5) Max Expected Return/CVaR	**0.136**	1.848	**-0.734**	**-0.509**	5.827	4.590	**0.023**	**0.030**	**0.037**	**13,738**	18
(C.6) Max Expected Return/VaR	**0.109**	1.823	**-0.730**	**-0.530**	6.101	5.192	**0.018**	**0.021**	0.039	**12,816**	16

Note that VaR and CVaR risks are estimated empirically at the 99% confidence level. The threshold is set to $u^l = 0.1$ for the lower threshold and $u^r = 0.9$ for the upper threshold where the study follows the study of Khanthaporn and Wichitaksorn (2022) and Z.-R. Wang et al. (2010). Economic indicators are an average turnover and cumulative portfolio wealth for the portfolio at the end of backtesting period assuming a $10,000 initial investment. Bold font indicates the top five best optimal weights portfolio for each indicator. n.a. = data not available. The benchmark portfolio strategy (A.3)-(A.6) in Panel A are used for the optimisation algorithm from the *fmincon* function in MATLAB. A risk-free rate = U.S. 3-Month Treasury Bill. Time = an average computational time at time t where $t = 1, \ldots, T$ in second. T = backtesting period. The ten selected stocks by sentimental analysis are BMY, BBL, AMD, AMGN, ISRG, IBM, SNY, TTE, CVS and SPGI.

Diversity and Convergence

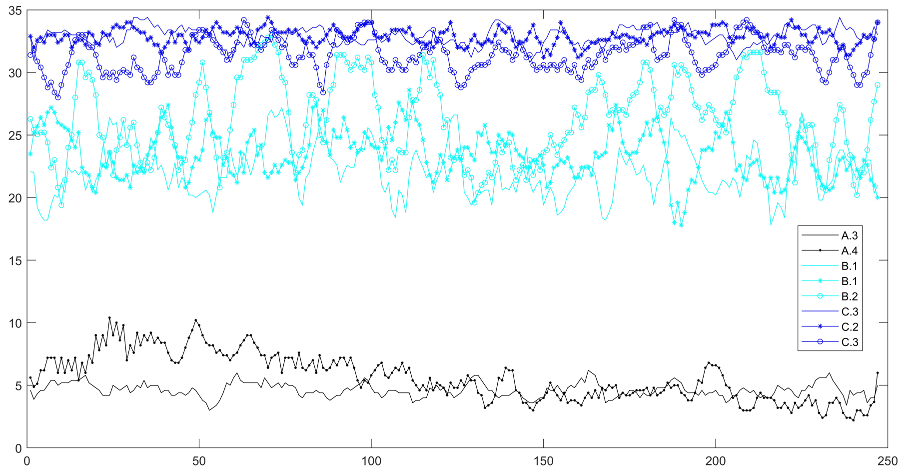
This section presents two more important indicators in portfolio performance measures - the diversity and convergence indicators against the empirical experiment in Scenario 2 using pre-COVID-19 data.

Another important performance measure in the PO problem is the diversity of the optimal weight solution. The diversity indicator may also imply risk diversification of the portfolio investment, namely, the higher the number of assets in portfolio, the better the investment strategy, given n is the total number of assets in the optimisation problems. Figure 5.2 depicts the number of optimal investment assets at the backtesting period, $t = 1, ..., T$, of all portfolio strategies in Scenario 2. The summary can be described as follows.

The GA and CSA algorithms performed better empirically due to this diversity indicator than the benchmark algorithms, where a percentage average[8] of algorithmic selected asset in investment is, respectively, 67% and 38%, with a benchmark is only 15%. In the details, the GA algorithm performed better than the CSA algorithm at all time T, while the benchmark algorithms seem to be the worst. Out of 35 investment assets in the PO problem, the optimal number of assets in an investment by the GA algorithm is approximately 32 assets, while the CSA algorithm's optimal number of investment assets is approximately 23 and the benchmarks algorithm is approximately five assets for the whole period of the backtesting T.

[8] The result of a percentage average of a selected asset in portfolio optimisation is an indicative value which is based on empirical CVaR risk and calculated by the investment asset proportion being greater than 0.1%.

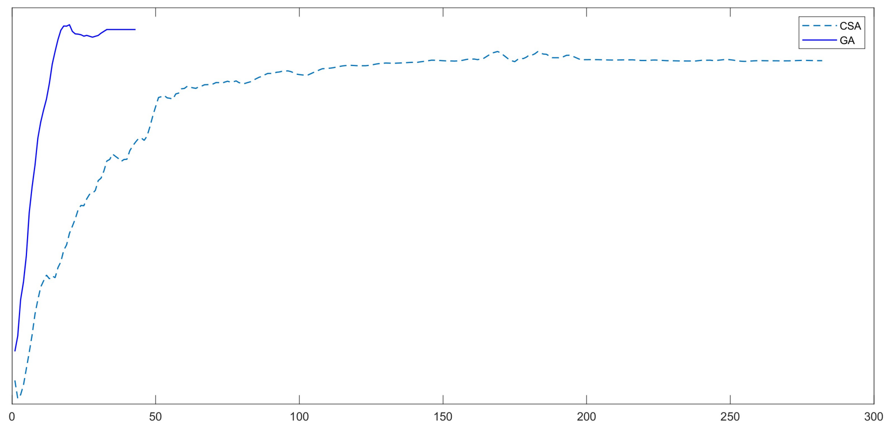
Figure 5.2: Scenario 2: The Smooth Diversity Indicator for All Portfolio Strategies in Table 5.4



Note that the plot depicts the number of optimal investment assets by optimisation algorithm in portfolio strategy (A.3), (A.4), (B.1), (B.2), (B.3), (C.1), (C.2) and (C.3) when investment proportion is greater than a constant c=0.05%.

For the convergence indicator, again, it confirms that the GA algorithm converges to the optimal solution faster than the CSA algorithm. See an empirical computational time of the CSA and GA algorithms in Table 5.4 for Scenario 2 and the Scenario 1 in Table 5.3. Furthermore, the indication of the learning process of the GA algorithm, when compared to the CSA algorithm, also indicates evidence of the GA outstanding performance in the convergence indicator - see Figure 5.3 for the demonstration of the empirical convergence plot of the CSA portfolio strategy B.4 and the GA portfolio strategy C.4 in Scenario 2.

Figure 5.3: Scenario 2: The Smooth Learning Process of the CSA and GA Optimisation Algorithms in Portfolio Strategy (B.4) and (C.4) in Table 5.4



Note that the plot depicts the learning process of portfolio strategy (B.4) and (C.4) by risk adjusted return at the end of the investment period.

In addition, the risk and reward indicator metrics of Scenario 2 using the pre-COVID-19 pandemic data presents in Table 5.4. The summary can be found as follows. Table 5.4 shows that the highest portfolio wealth at the end of the investment period is the benchmark portfolio strategy A.6, with the **m3** forecasting model. However, with this benchmark performance,the **m3** forecasting model is in the modified benchmark model where the study put a massive 13 bivariate copula candidates to the R-vine copula model, whereas the previous study did not follow the proposed forecasting model as the second-best portfolio wealth (portfolio strategy B.1). According to the scenario experiments, it is recommended that fundamental analysis is a significant, additional qualitative analysis in portfolio management according to the cumulative wealth indicator. For the rest of the risk and reward indicators in Scenario 2, see Table 5.4.

Table 5.4: Scenario 2: The Performance Measure of Portfolio Strategy among the Reward and Risk Indicator

Portfolio Strategy	**Avg. Return (%)**	**Std. (%)**	**Sortino Ratio**	**Sharpe Ratio**	**CVaR (%)**	**VaR (%)**	**STARR Ratio**	**Mean /VaR**	**Avg. Turnover**	**Portfolio Wealth**	**Time (Sec.)**
EQW Portfolios											
(A.1) Stock and Cryptocurrency	0.058	0.893	-3.035	-2.384	2.895	2.502	**0.020**	0.023	2.598	**11,425**	n.a.
(A.2) Stock	**0.068**	1.041	-2.590	-2.035	3.383	2.914	0.020	**0.023**	2.572	**11,657**	n.a.
Asset Class: Stock and Cryptocurrency											
(A.3) Min CVaR with Model m2	0.034	**0.405**	-7.560	-5.316	**1.673**	**1.053**	**0.020**	**0.032**	**0.026**	10,838	2
(A.4) Min CVaR with Model m3	**0.080**	1.081	-2.725	-1.949	4.852	2.450	0.016	**0.033**	0.083	**12,000**	1
Asset Class: Stock											
(A.5) Min CVaR with Model m2	-0.043	2.109	**-1.318**	**-1.057**	8.087	7.337	-0.005	-0.006	0.075	8,516	2
(A.6) Min CVaR with Model m3	**0.097**	1.434	-2.274	**-1.458**	4.307	4.047	**0.023**	**0.024**	0.080	**12,386**	1
Asset Class: Stock and Cryptocurrency											
(B.1) Max SR	**0.078**	**0.793**	-3.488	-2.659	**2.501**	**2.289**	**0.031**	**0.034**	**0.032**	**12,025**	232
(B.2) Max Expected Return/CVaR	-0.026	1.247	-2.284	-1.774	5.300	4.225	-0.005	-0.006	0.062	9,209	191
(B.3) Max Expected Return/VaR	-0.020	1.176	-2.237	-1.877	5.920	4.650	-0.003	-0.004	0.063	9,366	165
Asset Class: Stock											
(B.4) Max SR	0.013	1.535	**-1.762**	**-1.416**	6.086	4.554	0.002	0.003	0.066	10,029	151
(B.5) Max Expected Return/CVaR	0.022	1.654	**-1.578**	**-1.309**	7.427	4.790	0.003	0.005	0.063	10,199	208
(B.6) Max Expected Return/VaR	**0.061**	1.181	-2.384	-1.801	3.966	3.139	0.015	0.019	**0.034**	11,411	154
Asset Class: Stock and Cryptocurrency											
(C.1) Max SR	0.056	**0.813**	-3.318	-2.622	**2.642**	**2.437**	**0.021**	0.023	**0.030**	11,397	14
(C.2) Max Expected Return/CVaR	0.007	**0.667**	-4.465	-3.267	**2.460**	**1.983**	0.003	0.004	0.044	10,123	20
(C.3) Max Expected Return/VaR	0.015	**0.649**	-4.615	-3.348	**2.168**	**1.682**	0.007	0.009	0.044	10,316	21
Asset Class: Stock											
(C.4) Max SR	0.043	1.094	-2.595	-1.960	3.683	3.174	0.012	0.014	**0.034**	10,954	9
(C.5) Max Expected Return/CVaR	0.049	1.299	**-2.031**	**-1.645**	5.288	4.164	0.009	0.012	0.052	11,053	19
(C.6) Max Expected Return/VaR	0.040	1.261	**-2.073**	-1.702	5.344	3.769	0.008	0.011	0.054	10,822	15

Note that VaR and CVaR risks are estimated empirically at the 99% confidence level. The threshold is set to $u^l = 0.1$ for the lower threshold and $u^r = 0.9$ for the upper threshold where the study follows Khanthaporn and Wichitaksorn (2022) and Z.-R. Wang et al. (2010). Economic indicators are an average turnover and cumulative portfolio wealth for the portfolio at the end of backtesting period assuming a $10,000 initial investment. Bold font indicates the top five best optimal weights portfolio for each indicator. n.a. = data not available. The benchmark portfolio strategy (A.3)-(A.6) in Panel A are used in the optimisation algorithm from the *fmincon* function in MATLAB. A risk-free rate = U.S. 3-Month Treasury Bill. Time = an average computational time at time where $t = 1, \ldots, T$ in second. T = backtesting period. All stock portfolios consider 30 stocks in the PO problems.

5.6 Conclusion and Discussion

The study utilises machine-learning optimisation methods, namely CSA and GA, to improve the forecasting capabilities of the n-dimensional portfolio strategy based on a flexible and intensive R-vine copula-based model. The copula-based model is integrated with the univariate marginal function of the hybrid of ICAPM and asymmetric EGARCH model. The proposed innovation is the mixture distribution, where the component of mixture consists of two heavy-tails GPD distribution in EVT and a Gaussian distribution. This paper also proposes the magnitude 13 bivariate copula functions as the candidate for the multivariate R-vine model. There are two main scenarios with different performance measure objectives. Scenario 1 is to address the risk and reward indicator among the current market turmoil due to COVID-19 and others crisis. Scenario 2 is to address the other two main performance measures in the literature such as convergence and diversity indicators using the pre-COVID-19 data.

The study found that, overall, the proposed portfolio strategy in the R-vine based model, with computational intelligence, performs significantly better than the benchmarks. In particular, the CSA algorithm with the objective of expected return/CVaR and only stock in the portfolio performs the best in terms of the risk and reward indicator among the new normal economic. The GA algorithm is likely to perform better than the CSA algorithm in terms of the convergence and diversity indicators. Furthermore, the empirical study recommends that fundamental analysis of stock selection is a significant qualitative analysis in order to improve portfolio performance. In part of the proposed forecasting model, the virtue of it is to capture the many important characteristics of financial time series such as asymmetric fat-tail and volatility clustering, where it may also be a benefit to the portfolio solutions.

To conclude, it empirically claims that GA performance is very close to CSA performance. With an ultimate goal in investment wealth, the CSA algorithm empirically performs better than the GA algorithm. However, the GA algorithm can significantly beat the CSA algorithm in the term of computational time, and performs approximately three times faster. Even when the number of assets in the portfolio increase, the GA algorithm can be much faster than the CSA algorithm. Besides this, the paper statistically reveals evidence of change in the co-movement of the financial securities according to the impact of the current COVID-19 pandemic.

Regarding further research, there are many more areas where the paper may be extended. For example, the author's research interests include part of the multivariate dependence model which can be extended to other models, i.e., factor model. The stochastic volatility model is also a vast-class in univariate marginal models that could be explored in the future. The technique of forecasting model parameter estimation is another area of the author's research interests, especially in Bayesian and machine learning inference. Last, further research may focus on the area of optimisation, for example, other current machine learning algorithms, the multi-objective optimisation model, the class of Swarm algorithm or Black–Litterman portfolio optimisation.

Part III Conclusion and Discussion

Chapter 6

Conclusion

6.1 Findings and Contributions

This thesis provides several findings and contributes to the literature on quantitative portfolio risk management and financial markets under the new normal economy due to the impact of the current COVID-19 crisis and others.

First, the results of the first paper point out the outstanding forecasting returns performance of asymmetric generalised autoregressive conditional heteroscedastic and intertemporal capital asset pricing structure model in global stock markets around the world during times of current market turmoil of the COVID-19 pandemic. The model innovation investigation includes (1) the mixture distribution of two generalised Pareto and one Gaussian distributions, and (2) generalised error distribution. While the contribution of the model parameter estimate is an efficient Griddy Gibbs Bayesian Markov chain Monte Carlo algorithm integrated with parallel computing technique, and indicates better fitting algorithm to the global stock market indices than the traditional algorithm. The paper findings and contributions are proved through the magnitude of empirical and simulation experiments, where the proposed model returns the better forecast errors and forecast distribution than the benchmarks. It confirms that the application of the proposed model to predict financial returns during the current and on-going COVID-19 economic crisis.

Second, the next findings and contributions in literature are in the second paper, where is the research extension of the first paper. The results of the second paper discover an application of parallelise Bayesian machine learning inference to a regular-vine-copula-based model in graph theory, which it is comprehensive research in multivariate general n-dimensional analysis using stock data. The multivariate forecasting distribution of stock returns according to the advanced regular-vine-copula-based with ICAPM-EGARCH-Mixture model statistically confirms the model flexibility to capture the important financial data characteristics in the literature which is (non-)linear dependency, volatility clustering and asymmetric and heavy-tails distribution among the new normal. Furthermore, the second paper proves that the proposed vine-copula-based model is very flexible model in combining all significant parametric bivariate copula functions since the paper initiates a massive thirteen parametric bivariate copula functions to the proposed model with an outstanding performance using Bayesian machine learning inference among the benchmarks.

To be precise, the second paper indicates the improvement of forecasting performance of multivariate regular-vine-copula-based model and variational Bayes algorithm among the benchmarks. In additional, this paper reveals the algorithms of data simulation and density function simulation in a general n-dimensions of multivariate regular-vine-copula-based model and also compatible with two special sub-class models (drawable and canonical vine copulae) and any univariate marginal models. The study confirms that, during and ongoing the global COVID-19 crisis, the application of variational Bayes approaches to the multivariate regular-vine-copula-based model are still an outperforming model, and are one of the best candidate in returns forecasting in n-dimensional data in the current market circumstances.

Third, the next findings and contributions in literature are the ultimate thesis purpose in analytic methods in finance with applications to portfolio and risk management, which provided in the third paper. The findings of this paper are based on the exploration of machine learning optimisation algorithm in computational intelligence to the proposed forecasting model in the previous paper to the dynamic portfolio optimisation models. The paper reveals that the proposed portfolio strategies with computational intelligence are, in overall, an outstanding investment decision making among the benchmark portfolio strategies.

In particular, under the current market turmoil according to the COVID-19 pandemic and others, the paper confirms that the stocks portfolios is likely to be a better performance than portfolios of stocks and cryptocurrencies. While the clonal selection algorithm generally performs better performance than the genetic algorithm among multiple risk and reward indicators, however, not fast convergence to the solution as the genetic algorithm. The paper also empirically recommends that fundamental analysis is an essential preliminary analysis prior to formulate the optimisation problem because of the sake of solution convergence and reward per risk maximisation, i.e. an analysis of stock selection. Furthermore, this paper empirically confirms that optimisation methods in computational intelligence is significant far more diversity of the optimal weights solution than the traditional algorithm.

To conclude of the thesis findings and contributions, given the current financial market turbulence caused by events such as the COVID-19 crisis and the Russia-Ukraine crisis, the forecasting models are required to take into account several perspective of multivariate distribution of returns including fat-tail behaviors, non-normality, tail dependence and volatility clustering. According to several returns characteristics, copulae can plug in it into any univariate marginal models and any bivariate copula functions, and gain the sophisticated forecasting modelling to cope with those characteristics. Ultimately, the findings and contributions in literature of the thesis statistically discover the optimal solution of portfolio strategies offering to the investors through the advantages of the complex mathematical model of regular-vine-copula-based model, mixture distribution, Bayesian Markov chain Monte Carlo inference, Bayesian machine learning inference and machine learning optimisation.

6.2 Further Research

This thesis presents the application of the forecasting R-vine-ICAPM-EGARCH-Mixture model to portfolio problems and evolutionary computation in several perspective, where the thesis could be extended in many steps of research operation depending on the research interests. For instance, the future research may be constructed into four main categories as follows:

1. Univariate Marginal Model: This research area is a massive area in the literature. Since the copula flexibility, hence, the other marginal models in the application to copula models could be investigated, for example, the family of stochastic volatility model, i.e. see Hull and White (1987), the family of Brownian motion model, i.e see Lavenda (1985), the family of regression model, i.e. see Tibshirani (1996) or the model in data mining, i.e. see Friedman (2001).

2. Multivariate Copula Model: This research area is an active and popular not only in finance but also others. In last decade, many families of copula have been proposed, therefore, there are many potential in this area, for example, factor copula, mixture copula, or rotated copula (Creal & Tsay, 2015; Kakouris & Rustem, 2014; De Lira Salvatierra & Patton, 2015)

3. Model Parameter Estimation Method: The research extension in this area could be further investigation as well such that the other techniques in machine learning or Bayesian inference, for example and among others, Bayesian variational auto-encoder (B. Zhou, Gao, Tran & Gerlach, 2021).

4. Multi-objective Portfolio Problem and Optimisation Algorithm: There are a several possibilities of further research in this area and for example, can be found in the recent survey of Kalayci et al. (2019).

Besides this, the further research in a different data frequency among the model could be another possibility and also very popular in the recent QRM literature, for example, the study in high frequency data (De Lira Salvatierra & Patton, 2015) or the study in mixed frequency data (Oh & Patton, 2016). Computational time improvement is another interesting area in computer science through, for instance, the programming implementation in super computer. In addition, mixed asset class investment portfolio is another interesting study, and the financial products that could be considered in portfolio, for example, fixed income, derivatives, CDS, commodities or currencies, etc.

References

Aas, K., Czado, C., Frigessi, A. & Bakken, H. (2009). Pair-copula constructions of multiples dependence. *Insurance: Mathematics and Economics*, *44*(2), 182-198.

Agbeyegbe, T. D. (2022). Modeling jse stock returns dynamics: GARCH versus stochastic volatility. *Journal of Developing Areas*, *56*(1), 175 - 191.

Aghatise, O., Khan, F. & Ahmed, S. (2021). Reliability assessment of marine structures considering multidimensional dependency of the variables. *Ocean Engineering*, *230*, 109021.

Al Janabi, M. A. M., Arreola Hernandez, J., Berger, T. & Nguyen, D. K. (2017). Multivariate dependence and portfolio optimization algorithms under illiquid market scenarios. *European Journal of Operational Research*, *259*(3), 1121-1131.

Andrieu, C., Doucet, A. & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(3), 269-342.

Andriosopoulos, D., Doumpos, M., Pardalos, P. M. & Zopounidis, C. (2019). Computational approaches and data analytics in financial services: A literature review. *Journal of the Operational Research Society*, *70*(10), 1581-1599.

Ardia, D., Bluteau, K., Boudt, K. & Catania, L. (2018). Forecasting risk with Markov-switching GARCH models: A large-scale performance study. *International Journal of Forecasting*, *34*(4), 733-747.

Augustyniak, M. & Badescu, A. (2021). On the computation of hedging strategies in affine GARCH models. *Journal of Futures Markets*, *41*(5), 710-735.

Ausín, M. C. & Galeano, P. (2007). Bayesian estimation of the Gaussian mixture GARCH model. *Computational Statistics & Data Analysis*, *51*(5), 2636-2652.

Babaei, S., Sepehri, M. M. & Babaei, E. (2015). Multi-objective portfolio optimization considering the dependence structure of asset returns. *European Journal of Operational Research*, *244*(2), 525-539.

Bali, T. G. & Engle, R. F. (2010). The intertemporal capital asset pricing model with dynamic conditional correlations. *Journal of Monetary Economics*, *57*(4), 377-390.

Bauwens, L., Preminger, A. & Rombouts, J. V. K. (2010). Theory and inference for a Markov switching GARCH model. *The Econometrics Journal*, *13*(2), 218-244.

Bedford, T. & Cooke, R. M. (2001). Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial Intelligence*, *32*(1), 245-268.

Bedford, T. & Cooke, R. M. (2002). Vines: A new graphical model for dependent random variables. *The Annals of Statistics*, *30*(4), 1031-1068.

Bekiros, S., Hernandez, J. A., Hammoudeh, S. & Nguyen, D. K. (2015). Multivariate dependence risk and portfolio optimization: An application to mining stock portfolios. *Resources Policy*, *46*, 1-11.

Benczúr, A. A. (2011). Affordable supercomputing for data mining applications. *Procedia Computer Science*, *7*, 136-138.

Bera, A. K. & Higgins, M. L. (1993). ARCH models: Properties, estimation and testing. *Journal of Economic Surveys*, *7*(4), 305-366.

Berger, T. (2013). Forecasting value-at-risk using time varying copulae and EVT return distributions. *International Economics*, *133*, 93-106.

Bernardi, M. & Catania, L. (2018). Portfolio optimisation under flexible dynamic dependence modelling. *Journal of Empirical Finance*, *48*, 1-18.

Bildirici, M. & Ersin, Ö. Ö. (2009). Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul stock exchange. *Expert Systems with Applications*, *36*(4), 7355-7362.

Billio, M., Casarin, R. & Osuntuyi, A. (2016). Efficient Gibbs sampling for Markov switching GARCH models. *Computational Statistics Data Analysis*, *100*, 37-57.

Bin Shalan, S. A. & Ykhlef, M. (2015). Solving multi-objective portfolio optimization problem for Saudi Arabia stock market using hybrid clonal selection and particle swarm optimization. *Arabian Journal for Science and Engineering*, *40*(8), 2407-2421.

Bladt, M. & McNeil, A. J. (2022). Time series copula models using D-vines and v-transforms. *Econometrics and Statistics*, *24*, 27-48.

Blei, D. M., Kucukelbir, A. & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, *112*(518), 859-877.

Bodnar, T., Lindholm, M., Niklasson, V. & Thorsén, E. (2022). Bayesian portfolio selection using VaR and CVaR. *Applied Mathematics and Computation*, *427*.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307-327.

Brechmann, E. C., Czado, C. & Aas, K. (2012). Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, *40*(1), 68-85.

Breymann, W., Dias, A. & Embrechts, P. (2003). Dependence structures for multivariate high-frequency data in finance. *Quantitative Finance*, *3*(1), 1-14.

Bruhn, P. & Ernst, D. (2022). Assessing the risk characteristics of the cryptocurrency market: A GARCH-EVT-Copula approach. *Journal of Risk and Financial Management*, *15*(8), 346.

Burnet, F. M. (1959). *The clonal selection theory of acquired immunity*. Nashville: Vanderbilt University Press.

Butt, S. U., Arshad, M., Baqai, A. A., Saeed, H. A., Din, N. A. & Khan, R. A. (2021). Correction to: Locator placement optimization for minimum part positioning error during machining operation using genetic algorithm. *International Journal of Precision Engineering and Manufacturing*, *22*(6), 1169-1169.

Cai, J. (1994). A Markov model of switching-regime ARCH. *Journal of Business Economic Statistics*, *12*(3), 309-316.

Cambridge advanced learner's dictionary. (2013). Cambridge University Press.

Cao, H., Recknagel, F. & Orr, P. T. (2014). Parameter optimization algorithms for evolving rule models applied to freshwater ecosystems. *IEEE Transactions on Evolutionary Computation*, *18*(6), 793-806.

Caporale, G. M. & Zekokh, T. (2019). Modelling volatility of cryptocurrencies using Markov-Switching GARCH models. *Research in International Business and Finance*, *48*, 143-155.

Carnero, M. A., Peña, D. & Ruiz, E. (2004, 03). Persistence and Kurtosis in GARCH and Stochastic Volatility Models. *Journal of Financial Econometrics*, *2*(2), 319-342.

Chang, McAleer, M. & Wong, W.-K. (2018). Big data, computational science, economics, finance, marketing, management, and psychology: Connections. *Journal of Risk and Financial Management*, *11*(1).

Chang, B. & Joe, H. (2019). Prediction based on conditional distributions of vine copulae. *Computational Statistics Data Analysis*, *139*, 45-63.

Chang, T. J., Meade, N., Beasley, J. E. & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers Operations Research*, *27*(13), 1271-1302.

Chang, T.-J., Yang, S.-C. & Chang, K.-J. (2009). Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with Applications*, *36*(7), 10529-10537.

Chatrabgoun, O., Karimi, R., Daneshkhah, A., Abolfathi, S., Nouri, H. & Esmaeilbeigi, M. (2020). Copula-based probabilistic assessment of intensity and duration of cold episodes: A case study of Malayer vineyard region. *Agricultural and Forest Meteorology*, *295*, 108150.

Chellathurai, T. & Draviam, T. (2005). Dynamic portfolio selection with nonlinear transaction costs. *Proceedings: Mathematical, Physical and Engineering Sciences*, *461*(2062), 3183-3212.

Cheng, T., Gao, J. & Phillips, P. C. (2018). A frequentist approach to Bayesian asymptotics. *Journal of Econometrics*, *206*(2), 359-378.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association*, *90*(432), 1313-1321.

Chuffart, T. (2015). Selection criteria in regime switching conditional volatility models. *Econometrics*, *3*(2), 289-316.

Consigli, G. (2002). Tail estimation and mean–VaR portfolio selection in markets subject to financial instability. *Journal of Banking Finance*, *26*(7), 1355-1382.

Corus, D. & Oliveto, P. S. (2018). Standard steady state genetic algorithms can hillclimb faster than mutation-only evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *22*(5), 720-732.

Creal, D. D. & Tsay, R. S. (2015). High dimensional dynamic stochastic copula models. *Journal of Econometrics*, *189*(2), 335-345. doi: https://doi.org/10.1016/j.jeconom.2015.03.027

Czado, C. (2010). Pair-copula constructions of multivariate copulae. In P. Jaworski, F. Durante, W. K. Härdle & T. Rychlik (Eds.), *Copula theory and its applications* (p. 93-109). Springer Berlin Heidelberg.

Czado, C. (2013). Vine copulae and their applications to financial data. In *Actuarial and financial mathematics conference.*

Das, M. K., Goswami, A. & Rana, N. (2018). Risk sensitive portfolio optimization in a jump diffusion model with regimes. *SIAM Journal on Control and Optimization*, *56*(2), 1550-1576.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182-197.

De Castro, L. N. & Timmis, J. (2002). *Artificial immune systems: A new computational intelligence approach*. Springer Science & Business Media.

De Castro, L. N. & Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, *6*(3), 239-251.

De Lira Salvatierra, I. & Patton, A. J. (2015). Dynamic copula models and high frequency data. *Journal of Empirical Finance*, *30*, 120-135.

Deng, G.-F., Lin, W.-T. & Lo, C.-C. (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications*, *39*(4), 4558-4566.

Deresa, N. W., Van Keilegom, I. & Antonio, K. (2022). Copula-based inference for bivariate survival data with left truncation and dependent censoring. *Insurance: Mathematics and Economics*, *107*, 1-21.

Ding, C., Chen, L. & Zhong, B. (2019). Exploration of intelligent computing based on improved hybrid genetic algorithm. *Cluster Computing*, *22*, 9037-9045.

Dinh, V., Rundell, A. E. & Buzzard, G. T. (2017). Convergence of griddy Gibbs sampling and other perturbed Markov chains. *Journal of Statistical Computation and Simulation*, *87*(7), 1379-1400.

Dißmann, J., Brechmann, E. C., Czado, C. & Kurowicka, D. (2013). Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics Data Analysis*, *59*, 52-69.

Drost, F. C. & Werker, B. J. (1996). Closing the GARCH gap: Continuous time GARCH modeling. *Journal of Econometrics*, *74*(1), 31-57. Retrieved from `https://www.sciencedirect.com/science/article/pii/030440769501750X` doi: https://doi.org/10.1016/0304-4076(95)01750-X

Du, J. (2022). Mean–variance portfolio optimization with deep learning based-forecasts for cointegrated stocks. *Expert Systems with Applications*, *201*, 117005.

Du, J.-H., Guo, Y. & Wang, X. (2022). High-dimensional portfolio selection with cardinality constraints. *Journal of the American Statistical Association*, 1-13.

Duan, J.-C. (1997). Augmented GARCH (p,q) process and its diffusion limit. *Journal of Econometrics*, *79*(1), 97-127.

D'Amico, G. & Petroni, F. (2018). Copula based multivariate semi-Markov models with applications in high-frequency finance. *European Journal of Operational Research*, *267*(2), 765-777.

Engle, R. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, *50*(4), 987-1007.

Engle, R. (2004). Risk and volatility: Econometric models and financial practice. *The American Economic Review*, *94*(3), 405-420.

Engle, R., Lilien, D. & Robins, R. (1987). Estimating time varying risk premia in the term structure: The ARCH-M model. *Econometrica*, *55*(2), 391-407.

Fan, Y. & Patton, A. J. (2014). Copulae in econometrics. *Annual Review of Economics*, *6*(1), 179-200.

Fazel Zarandi, M. H., Rezaee, B., Turksen, I. B. & Neshat, E. (2009). A type-2 fuzzy rule-based expert system model for stock price analysis. *Expert Systems with Applications*, *36*(1), 139-154.

Fengler, M. R. & Okhrin, O. (2016). Managing risk with a realized copula parameter. *Computational Statistics & Data Analysis*, *100*, 131-152.

Fernández, A. & Gómez, S. (2007). Portfolio selection using neural networks. *Computers & Operations Research*, *34*(4), 1177-1191.

Feuerriegel, S. & Gordon, J. (2018). Long-term stock index forecasting based on text mining of regulatory disclosures. *Decision Support Systems*, *112*, 88-97.

Forrest, S. & Mitchell, M. (1993). Relative building-block fitness and the building-block hypothesis. In L. D. Whitley (Ed.), *Foundations of genetic algorithms* (Vol. 2, p. 109-126). Elsevier.

Francq, C., Wintenberger, O. & Zakoïan, J.-M. (2013). GARCH models without positivity constraints: Exponential or log GARCH? *Journal of Econometrics*, *177*(1), 34-46.

Francq, C., Wintenberger, O. & Zakoïan, J.-M. (2018). Goodness-of-fit tests for Log-GARCH and EGARCH models. *TEST*, *27*(1), 27-51.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189-1232.

Fujiwara, T., Li, J. K., Mubarak, M., Ross, C., Carothers, C. D., Ross, R. B. & Ma, K.-L. (2018). A visual analytics system for optimizing the performance of large-scale networks in supercomputing systems. *Visual Informatics*, *2*(1), 98-110.

Ganapathy, K. & Wah, B. (1992). Optimal design of lower dimensional processor arrays for uniform recurrences. In *[1992] proceedings of the international conference on application specific array processors* (p. 636-648).

García-Mejía, J. F., Guzma, D. B., Rodriguez-López, E. T., Navarrete, M. d. J., Martinez-Garduño, Y. & Torres-Reyes, C. E. (2021). A clonal selection algorithm for the design of an optimal investment portfolio. In *2021 ieee international summer power meeting/international meeting on communications and computing (rvp-ai/rocc)* (p. 1-6).

Geboers, H., Depaire, B. & Annaert, J. (2022). A review on drawdown risk measures and their implications for risk management. *Journal of Economic Surveys*, *00, 1-25*(00).

Geweke, J. (1986). Comment. *Econometric Reviews*, *5*(1), 57-61.

Glosten, L., Jagannathan, R. & Runkle, D. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *The journal of finance*, *48*(5), 1779-1801.

Goel, A. & Mehra, A. (2021). Robust omega ratio optimization using regular vines. *Optimization Letters*, *15*(6), 2067-2108.

Goel, A., Sharma, A. & Mehra, A. (2019). Robust optimization of mixed CVaR STARR ratio using copulae. *Journal of Computational and Applied Mathematics*, *347*, 62-83.

Golmakani, H. R. & Alishah, E. J. (2008). Portfolio selection using an artificial immune system. In *2008 ieee international conference on information reuse and integration*

(p. 28-33).
Gray, S. F. (1996). Modeling the conditional distribution of interest rates as a regime-switching process. *Journal of Financial Economics*, *42*(1), 27-62.
Gunawan, D., Tran, M. N., Suzuki, K., Dick, J. & Kohn, R. (2019). Computationally efficient Bayesian estimation of high-dimensional Archimedean copulae with discrete and mixed margins. *Statistics and Computing*, *29*(5), 933-946.
Górecki, J., Hofert, M. & Okhrin, O. (2021). Outer power transformations of hierarchical Archimedean copulae: Construction, sampling and estimation. *Computational Statistics and Data Analysis*, *155*.
Hafner, C. M. & Kyriakopoulou, D. (2021). Exponential-type GARCH models with linear-in-variance risk premium. *Journal of Business & Economic Statistics*, *39*(2), 589-603.
Hafner, C. M. & Manner, H. (2012). Dynamic stochastic copula models: estimation, inference and applications. *Journal of Applied Econometrics*, *27*(2), 269-295.
Hamilton, J. D. & Susmel, R. (1994). Autoregressive conditional heteroskedasticity and changes in regime. *Journal of Econometrics*, *64*(1), 307-333.
Harik, G. R., Lobo, F. G. & Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, *3*(4), 287-297.
Hentschel, L. (1995). All in the family nesting symmetric and asymmetric GARCH models. *Journal of Financial Economics*, *39*(1), 71-104.
Holland, J. H. (1995). *Adaptation in natural and artificial systems* (4th ed.). MIT press.
Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, *81*(10), 3088-3092.
Hosking, J. R. M. & Wallis, J. R. (1987). Parameter and quantile estimation for the generalized Pareto distribution. *Technometrics*, *29*(3), 339-349.
Hull, J. & White, A. (1987). The pricing of options on assets with stochastic volatilities. *The Journal of Finance*, *42*(2), 281-300.
Hurley, J. (2019). *Supercomputing overview* (Web Page Nos. August 10, 2019). Retrieved from `https://astronomy.swin.edu.au/supercomputing/`
Jacquier, E., Polson, N. G. & Rossi, P. E. (1994). Bayesian analysis of stochastic volatility models. *Journal of Business Economic Statistics*, *12*(4), 371–389.
Jalota, H. & Thakur, M. (2018). Genetic algorithm designed for solving portfolio optimization problems subjected to cardinality constraint. *International Journal of System Assurance Engineering and Management*, *9*(1), 294-305.
Jeon, J. & Park, K. (2021). Portfolio selection with drawdown constraint on consumption: a generalization model. *Mathematical Methods of Operations Research*, *93*(2), 243-289.
Joe, H. (2015). *Dependence modelling with copulae*. CRC Press.
Joe, H., Li, H. & Nikoloulopoulos, A. K. (2010). Tail dependence functions and vine copulae. *Journal of Multivariate Analysis*, *101*(1), 252-270.
Jose Rodriguez, M. & Ruiz, E. (2012). Revisiting several popular GARCH models with leverage effect: Differences and similarities. *Journal of Financial Econometrics*, *10*, 637-668.
Kakouris, I. & Rustem, B. (2014). Robust portfolio optimization with copulae. *European Journal of Operational Research*, *235*(1), 28-37.

Kalayci, C. B., Ertenlice, O. & Akbay, M. A. (2019). A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Systems with Applications*, *125*, 345-368.

Kamble, R. A. (2017). Short and long term stock trend prediction using decision tree. In *2017 international conference on intelligent computing and control systems (iciccs)* (p. 1371-1375).

Karmakar, M. & Paul, S. (2018). Intraday portfolio risk management using VaR and CVaR:A CGARCH-EVT-Copula approach. *International Journal of Forecasting*.

Karypis, G. & Kumar, V. (1993). Efficient parallel mappings of a dynamic programming algorithm: a summary of results. In *[1993] proceedings seventh international parallel processing symposium* (p. 563-568).

Khanthaporn, R. & Wichitaksorn, N. (2022). Modelling and forecasting COVID-19 stock returns using asymmetric GARCH-ICAPM with mixture and heavy-tailed distributions. *Applied Economics*, 1-20.

Kim, S., Shephard, N. & Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with arch models. *The Review of Economic Studies*, *65*(3), 361–393.

Klüppelberg, C., Lindner, A. & Maller, R. (2004). A continuous-time GARCH process driven by a Lévy process: Stationarity and second-order behaviour. *Journal of Applied Probability*, *41*(3), 601–622.

Kolm, P. N., Tütüncü, R. & Fabozzi, F. J. (2014). 60 years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, *234*(2), 356-371.

Koop, G., Poirier, D. J. & Tobias, J. L. (2007). *Bayesian econometric methods*. Cambridge ; New York : Cambridge University Press, 2007.

Kurowicka, D. & Cooke, R. M. (2005). Distribution-free continuous Bayesian belief nets. In *Modern statistical and mathematical methods in reliability* (Vol. 10, p. 309-322). WORLD SCIENTIFIC.

Kurowicka, D. & Joe, H. (Eds.). (2010). *Dependence modeling: Vine copula handbook*. WORLD SCIENTIFIC.

Kırış, & Ustun, O. (2012). An integrated approach for stock evaluation and portfolio optimization. *Optimization*, *61*(4), 423-441.

Lam, H.-F., Hu, J., Zhang, F.-L. & Ni, Y.-C. (2019). Markov chain Monte Carlo-based Bayesian model updating of a sailboat-shaped building using a parallel technique. *Engineering Structures*, *193*, 12-27.

Lanzafame, R., Timmermans, M., Orlin, F., Valls, S. S. & Nápoles, O. M. (2021). Probabilistic design for civil engineering infrastructure using vine-copulae. In *Proceedings of the 31st european safety and reliability conference, esrel 2021* (p. 3326-3333).

Lavenda, B. H. (1985). Brownian motion. *Scientific American*, *252*(2), 70-85.

Li, F. & Kang, Y. (2018). Improving forecasting performance using covariate-dependent copula models. *International Journal of Forecasting*, *34*(3), 456-476.

Li, H., Liu, Z. & Wang, S. (2022). Vines climbing higher: Risk management for commodity futures markets using a regular vine copula approach. *International Journal of Finance & Economics*, *27*(2), 2438-2457.

Li, L., Lin, Q. & Ming, Z. (2022). A survey of artificial immune algorithms for multi-objective optimization. *Neurocomputing*, *489*, 211-229.

Li, X. & Babovic, V. (2019). Multi-site multivariate downscaling of global climate model outputs: an integrated framework combining quantile mapping, stochastic weather generator and empirical copula approaches. *Climate Dynamics*, *52*(9), 5775-5799.

Liang, C., Xu, Y., Chen, Z. & Li, X. (2022). Forecasting china's stock market volatility with shrinkage method: Can adaptive lasso select stronger predictors from numerous predictors? *International Journal of Finance & Economics*, *n/a*(n/a).

Liu, Y.-J. & Zhang, W.-G. (2015). A multi-period fuzzy portfolio optimization model with minimum transaction lots. *European Journal of Operational Research*, *242*(3), 933-941.

Loaiza-Maya, R. & Smith, M. S. (2019). Variational Bayes estimation of discrete-margined copula models with application to time series. *Journal of Computational and Graphical Statistics*, *28*(3), 523-539.

Lopatka, K. & Czyzewski, A. (2014). Acceleration of decision making in sound event recognition employing supercomputing cluster. *Information Sciences*, *285*, 223-236.

Malossini, A., Blanzieri, E. & Calarco, T. (2008). Quantum genetic optimization. *IEEE Transactions on Evolutionary Computation*, *12*(2), 231-241.

Mantel, N. & Myers, M. (1971). Problems of convergence of maximum likelihood iterative procedures in multiparameter situations. *Journal of the American Statistical Association*, *66*(335), 484-491.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77-91.

Maya, R. A. L., Gomez-Gonzalez, J. E. & Velandia, L. F. M. (2015). Latin american exchange rate dependencies: A regular vine copula approach. *Contemporary Economic Policy*, *33*(3), 535-549.

McNeil, A. J., Frey, R. & Embrechts, P. (2015). *Quantitative risk management : Concepts, techniques and tools*. Princeton : Princeton University Press, [2015].

Mendes, R. R. A., Paiva, A. P., Peruchi, R. S., Balestrassi, P. P., Leme, R. C. & Silva, M. B. (2016). Multiobjective portfolio optimization of ARMA–GARCH time series based on experimental designs. *Computers Operations Research*, *66*, 434-444.

Merton, R. C. (1973). An intertemporal capital asset pricing model. *Econometrica*, *41*(5), 867-887.

Milhøj, A. (1987). A multiplicative parameterization of ARCH models. In *Universitetets statistiske institut.*

Mitchell, M. (1998). *An introduction to genetic algorithms*. Cambridge, MA, USA: MIT Press.

Morales Napoles, O., Cooke, R. M. & Kurowicka, D. (2010). *About the number of vines and regular vines on n nodes (submitted for publication).*

Müller, D. & Czado, C. (2018). Representing sparse Gaussian DAGs as sparse R-vines allowing for non-Gaussian dependence. *Journal of Computational and Graphical Statistics*, *27*(2), 334-344.

Müller, D. & Czado, C. (2019). Dependence modelling in ultra high dimensions with vine copulae and the graphical Lasso. *Computational Statistics Data Analysis*, *137*, 211-232.

Nadarajah, S. (2005). A generalized normal distribution. *Journal of Applied Statistics*, *32*(7), 685-694.

Nagler, T., Bumann, C. & Czado, C. (2019). Model selection in sparse high-dimensional vine copula models with an application to portfolio risk. *Journal of Multivariate Analysis*, *172*, 180-192. (Dependence Models)

Nelson, D. B. (1990). Arch models as diffusion approximations. *Journal of Econometrics*, *45*(1), 7-38.

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica*, *59*(2), 347-370.

Neumeyer, N., Omelka, M. & Hudecová, (2019). A copula approach for dependence modeling in multivariate nonparametric time series. *Journal of Multivariate Analysis*, *171*, 139-162.

Nezamdoust, S. & Eskandari, F. (2022). Ridge shrinkage estimators in finite mixture of generalized estimating equations. *Journal of Mathematics and Modeling in Finance*, *2*(2), 91-106.

Nikoloulopoulos, A. K., Joe, H. & Li, H. (2012). Vine copulae with asymmetric tail dependence and applications to financial return data. *Computational Statistics Data Analysis*, *56*(11), 3659-3673.

Nti, I. K., Adekoya, A. F. & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, *53*(4), 3007-3057.

Oh, D. H. & Patton, A. J. (2016). High-dimensional copula-based distributions with mixed frequency data. *Journal of Econometrics*, *193*(2), 349-366.

Oh, D. H. & Patton, A. J. (2018). Time-varying systemic risk: Evidence from a dynamic copula model of CDS spreads. *Journal of Business Economic Statistics*, *36*(2), 181-195.

Ong, V. M. H., Nott, D. J. & Smith, M. S. (2018). Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*, *27*(3), 465-478.

Pareek, M. K. & Thakkar, P. (2015). Surveying stock market portfolio optimization techniques. In *2015 5th Nirma university international conference on engineering (nuicone)* (p. 1-5).

Patton, A. J. (2012). A review of copula models for economic time series. *Journal of Multivariate Analysis*, *110*, 4-18.

Popović, B. V., Genç, A. I. & Domma, F. (2022). Generalized proportional reversed hazard rate distributions with application in medicine. *Statistical Methods and Applications*, *31*(3), 459-480.

Prudencio, E. & Cheung, S. H. (2012). Parallel adaptive multilevel sampling algorithms for the Bayesian analysis of mathematical models. *International Journal for Uncertainty Quantification*, *2*(3), 215-237.

Puelz, A. V. (2002). A stochastic convergence model for portfolio selection. *Operations Research*, *50*(3), 462-476.

Qin, Z. (2015). Mean-variance model for portfolio optimization problem in the simultaneous presence of random and uncertain returns. *European Journal of Operational Research*, *245*(2), 480-488.

Ritter, C. & Tanner, M. A. (1992). Facilitating the Gibbs sampler: The Gibbs stopper and the Griddy-Gibbs sampler [Journal Article]. *Journal of the American Statistical*

Association, *87*(419), 861-868.
Rockafellar, R. T. & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, *2*, 21-42.
Sahamkhadam, M., Stephan, A. & Östermark, R. (2018). Portfolio optimization based on GARCH-EVT-Copula forecasting models. *International Journal of Forecasting*, *34*(3), 497-506.
Sanchez-Vazquez, C., Avila-Costa, M. & Cervantes-Pérez, F. (2014). Implementation of a computational model for information processing and signaling from a biological neural network of neostriatum nucleus. *Journal of Applied Research and Technology*, *12*(3), 568-584.
Scherr, S. & Zhou, J. (2020). Automatically identifying relevant variables for linear regression with the lasso method: A methodological primer for its application with r and a performance contrast simulation with alternative selection strategies. *Communication Methods and Measures*, *14*(3), 204-211.
Schinke-Nendza, A., von Loeper, F., Osinski, P., Schaumann, P., Schmidt, V. & Weber, C. (2021). Probabilistic forecasting of photovoltaic power supply — a hybrid approach using D-vine copulae to model spatial dependencies. *Applied Energy*, *304*, 117599.
Segnon, M. & Trede, M. (2018). Forecasting market risk of portfolios: Copula-Markov switching multifractal approach. *The European Journal of Finance*, *24*(14), 1123-1143.
Sehgal, R., Sharma, A. & Mansini, R. (2023). Worst-case analysis of Omega-VaR ratio optimization model. *Omega (United Kingdom)*, *114*.
Serban, F., Stefanescu, V. & Ferrara, M. (2013). Portfolio optimization in the framework mean-variance-var. *Economic Computation & Economic Cybernetics Studies Research*, *47*(1), 61-67.
Sharma, A., Hazra, B. & Sekharan, S. (2021). Stochastic seepage and slope stability analysis using vine-copula based multivariate random field approach: Consideration to non-Gaussian spatial and cross-dependence structure of hydraulic parameters. *Computers and Geotechnics*, *130*, 103918.
Sharpe, W. F. (1963). A simplified model for portfolio analysis. *Management Science*, *9*(2), 277-293.
Sharpe, W. F. (1994). The Sharpe ratio. *The Journal of Portfolio Management*, *21*(1), 49.
Shaw, D. X., Liu, S. & Kopman, L. (2008). Lagrangian relaxation procedure for cardinality-constrained portfolio optimization. *Optimization Methods and Software*, *23*(3), 411-420.
Shen, K.-Y. & Tzeng, G.-H. (2015). Combined soft computing model for value stock selection based on fundamental analysis. *Applied Soft Computing*, *37*, 142-155.
Siddhartha, C. & Edward, G. (1996). Markov chain Monte Carlo simulation methods in econometrics. *Econometric Theory*, *12*(3), 409.
Singh, V. P. & Guo, H. (1995). Parameter estimation for 3-parameter generalized Pareto distribution by the principle of maximum entropy (POME). *Hydrological Sciences Journal*, *40*(2), 165-181.
Sivanandam, S. N. & Deepa, S. N. (2008). Genetic algorithms. In S. N. Sivanandam & S. N. Deepa (Eds.), *Introduction to genetic algorithms* (p. 15-37). Berlin,

Heidelberg: Springer Berlin Heidelberg.
Sklar, M. (1959). Fonctions de repartition an dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Universit'e de Paris*, *8*, 229-231.
Smith, M. S., Loaiza-Maya, R. & Nott, D. J. (2020). High-dimensional copula variational approximation through transformation. *Journal of Computational and Graphical Statistics*, *29*(4), 729-743.
Smith, R. E., Forrest, S. & Perelson, A. S. (1993). Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation*, *1*(2), 127–149.
So, M. K. P. & Yeung, C. Y. T. (2014). Vine-copula GARCH model with dynamic conditional dependence. *Computational Statistics Data Analysis*, *76*, 655-671.
Sucarrat, G. & Escribano, A. (2018). Estimation of log-GARCH models in the presence of zero returns. *The European Journal of Finance*, *24*, 809-827.
Sucarrat, G., Gronneberg, S. & Escribano, A. (2016). Estimation and inference in univariate and multivariate log-GARCH-X models when the conditional density is unknown. *Computational Statistics & Data Analysis*, *100*, 582-594.
Sun, F., Fu, F., Liao, H. & Xu, D. (2020). Analysis of multivariate dependent accelerated degradation data using a random-effect general wiener process and d-vine copula. *Reliability Engineering System Safety*, *204*, 107168.
Swinburne University of Technology. (2019). *Welcome to OzSTAR documentation! — OzSTAR user guide documentation* (Web Page Nos. August 11, 2019). Retrieved from `https://supercomputing.swin.edu.au/docs/index.html`
Tao, Y., Wang, Y., Wang, D., Ni, L. & Wu, J. (2021). A C-vine copula framework to predict daily water temperature in the Yangtze river. *Journal of Hydrology*, *598*, 126430.
Taylor, S. J. (1986). *Modelling financial time series.* Wiley: Chichester.
Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267-288.
Timmis, J., Andrews, P., Owens, N. & Clark, E. (2008). An interdisciplinary perspective on artificial immune systems. *Evolutionary Intelligence*, *1*(1), 5-26.
Topaloglou, N., Vladimirou, H. & Zenios, S. A. (2008). A dynamic stochastic programming model for international portfolio management. *European Journal of Operational Research*, *185*(3), 1501-1524.
Tran, M.-N., Nguyen, T.-N. & Dao, V.-H. (2021). A practical tutorial on variational Bayes..
University of Surrey. (2019). *UNIX tutorial for beginners, department of electrical and electronic engineering* (Web Page Nos. August 11, 2019). Retrieved from `http://www.ee.surrey.ac.uk/Teaching/Unix/`
Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, *57*(2), 307-333.
Wang, X., Zhang, H. & Zhao, Y. (2017). Parameters estimations for continuous-time stochastic volatility models. In *2017 36th chinese control conference (ccc)* (p. 2315-2320).
Wang, Z.-R., Chen, X.-H., Jin, Y.-B. & Zhou, Y.-J. (2010). Estimating risk of foreign exchange portfolio: Using VaR and CVaR based on GARCH–EVT-Copula model. *Physica A: Statistical Mechanics and its Applications*, *389*(21), 4918-4928.
Wei, T., Wang, S., Zhong, J., Liu, D. & Zhang, J. (2022). A review on evolutionary

multitask optimization: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, *26*(5), 941-960.

Weiß, G. N. F. & Supper, H. (2013). Forecasting liquidity-adjusted intraday Value-at-Risk with vine copulae. *Journal of Banking Finance*, *37*(9), 3334-3350.

Wichitaksorn, N. & Choy, S. B. (2015). Bayesian parallel computation for intractable likelihood using Griddy-Gibbs sampler. In S. K. Upadhyay (Ed.), *Current trends in Bayesian methodology with applications* (p. 627-628). CRC Press.

Wichitaksorn, N., Gerlach, R. & Choy, S. T. B. (2019). Efficient MCMC estimation of some elliptical copula regression models through scale mixtures of normals. *Applied Stochastic Models in Business and Industry*, *35*(3), 808-822.

World Bank. (2020). *The global economic outlook during the COVID-19 pandemic: A changed world*.

Xie, Y. (2009). Consistency of maximum likelihood estimators for the regime-switching GARCH model. *Statistics*, *43*(2), 153-165.

Xu, D., Bai, Z., Jin, X., Yang, X., Chen, S. & Zhou, M. (2022). A mean-variance portfolio optimization approach for high-renewable energy hub. *Applied Energy*, *325*, 119888.

Xu, Q., Zhou, Y., Jiang, C., Yu, K. & Niu, X. (2016). A large CVaR-based portfolio selection model with weight constraints. *Economic Modelling*, *59*, 436-447.

Xue, B., Zhang, M., Browne, W. N. & Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, *20*(4), 606-626.

Zakoian, J.-M. (1994). Threshold heteroskedastic models. *Journal of Economic Dynamics and Control*, *18*(5), 931-955.

Zeiler, M. D. (2012). *ADADELTA: an adaptive learning rate method* (Vol. abs/1212.5701). Retrieved from `http://arxiv.org/abs/1212.5701`

Zhang, B., Wei, Y., Yu, J., Lai, X. & Peng, Z. (2014). Forecasting VaR and ES of stock index portfolio: A vine copula method. *Physica A: Statistical Mechanics and its Applications*, *416*, 112-124.

Zhang, Q. & Li, H. (2007). MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712-731.

Zhang, Y., Li, X. & Guo, S. (2018). Portfolio selection problems with Markowitz's mean–variance framework: a review of literature. *Fuzzy Optimization and Decision Making*, *17*(2), 125-158.

Zhou, B., Gao, J., Tran, M.-N. & Gerlach, R. (2021). Manifold optimization-assisted Gaussian variational approximation. *Journal of Computational and Graphical Statistics*, *30*(4), 946-957.

Zhou, R. & Ji, M. (2021). Modelling mortality dependence: An application of dynamic vine copula. *Insurance: Mathematics and Economics*, *99*, 241-255.

Zhu, H., Wang, Y., Wang, K. & Chen, Y. (2011). Particle swarm optimization (PSO) for the constrained portfolio optimization problem. *Expert Systems with Applications*, *38*(8), 10161-10169.

Appendix A

Chapter 3 Additional Information

A.1 Additional Simulation Study Results

A.1.1 Prior Specification

This section shows prior specification of all proposed models in the Bayesian MCMC method. The prior distribution is vital in Bayesian statistics because it will determine the moment of posterior distribution. Regarding the estimation, the specified priors of full posterior distribution of all proposed models are provided in Table A.1.

Table A.1: Prior Specification of All Proposed Models

Model/Prior		
EGARCH(1,1,1)-ICAPM-Mixture	**LogGARCH(1,1)-ICAPM-Mixture**	**GJR-GARCH(1,1)-ICAPM-Mixture**
$\lambda_1 \backsim N\left(0,10^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$
$\lambda_2 \backsim N\left(0,1\right)$	$\lambda_2 \backsim \Gamma\left(70,0.00065\right)$	$\lambda_2 \backsim \Gamma\left(100,0.00035\right)$
$\omega \backsim N\left(0,1\right)$	$\omega \backsim B\left(300,7000\right)$	$\omega \backsim B\left(220,25000\right)$
$\gamma \backsim N\left(0,1\right)$	$\alpha^{+} \backsim N\left(0,0.00017^2\right)$	$\alpha \backsim B\left(40,1000\right)$
$\delta \backsim B\left(2,5\right)$	$\alpha^{-} \backsim B\left(2,200\right)$	$\gamma \backsim \Gamma\left(10,0.0006\right)$
$\nu \backsim B\left(2,4\right)$	$\nu \backsim \Gamma\left(70,1590\right)$	$\nu \backsim \Gamma\left(70,1570\right)$
$\xi^L \backsim Unif\left(0,0.5\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$
$\xi^R \backsim Unif\left(0,0.5\right)$	$\xi^R \backsim B\left(5,5\right)$	$\xi^R \backsim B\left(15,55\right)$
EGARCH(1,1,1)-ICAPM-GED	**GARCH(1,1)-ICAPM-Mixture**	
$\lambda_1 \backsim B\left(2,900000\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	
$\lambda_2 \backsim N\left(38.63,1^2\right)$	$\lambda_2, \backsim \Gamma\left(70,0.00065\right)$	
$\omega \backsim N\left(0.001,0.0044^2\right)$	$\omega \backsim B\left(60,7000\right)$	
$\gamma \backsim N\left(385,10^2\right)$	$\alpha \backsim B\left(660,7000\right)$	
$\delta \backsim \Gamma\left(2,48.5\right)$	$\nu \backsim \Gamma\left(70,0.014\right)$	
$\nu \backsim \Gamma\left(1,0.0028\right)$	$\xi^L \backsim \Gamma\left(70,0.00093\right)$	
$\xi \backsim \Gamma\left(125,0.02\right)$	$\xi^R \backsim B\left(5,10\right)$	

A.1.2 Posterior Samples of Model Parameters

This section shows convergence posterior sample draws and its density of model parameters for all proposed models including the GJR-GARCH(1,1)-ICAPM-Mixture process, LogGARCH(1,1)-ICAPM-Mixture process, GARCH(1,1)-ICAPM-Mixture process and EGARCH(1,1,1)-ICAPM-GED process in Figure A.1 - Figure A.4.

Figure A.1: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GJR-GARCH(1,1)-ICAPM-Mixture Process

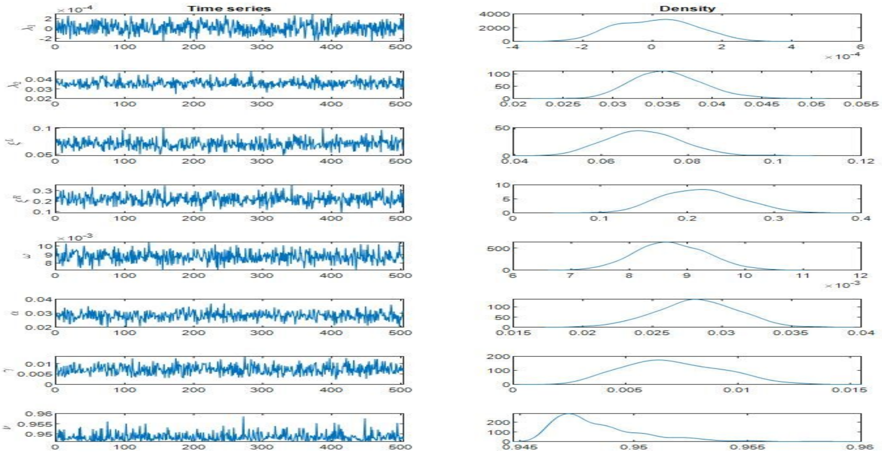


Figure A.2: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from LogGARCH(1,1)-ICAPM-Mixture Process

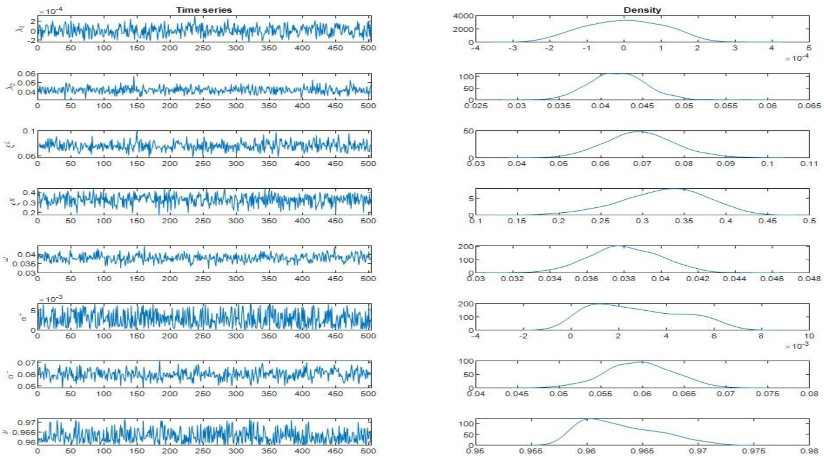


Figure A.3: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GARCH(1,1)-ICAPM-Mixture Process

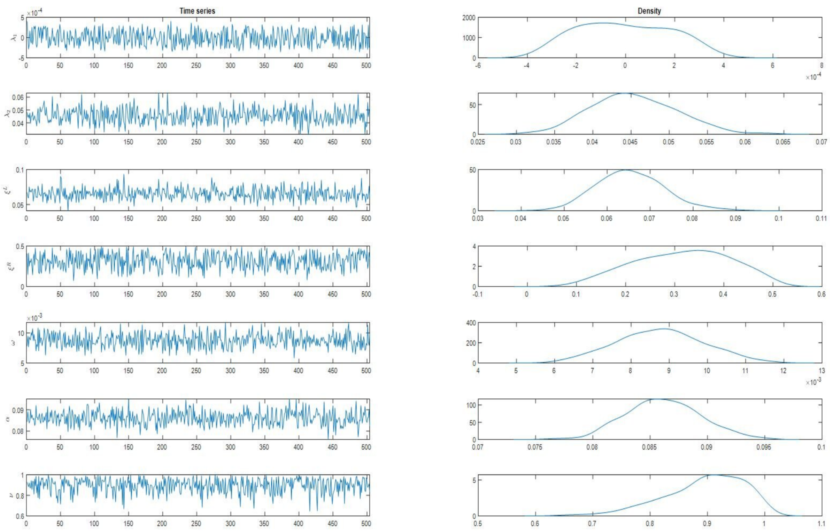
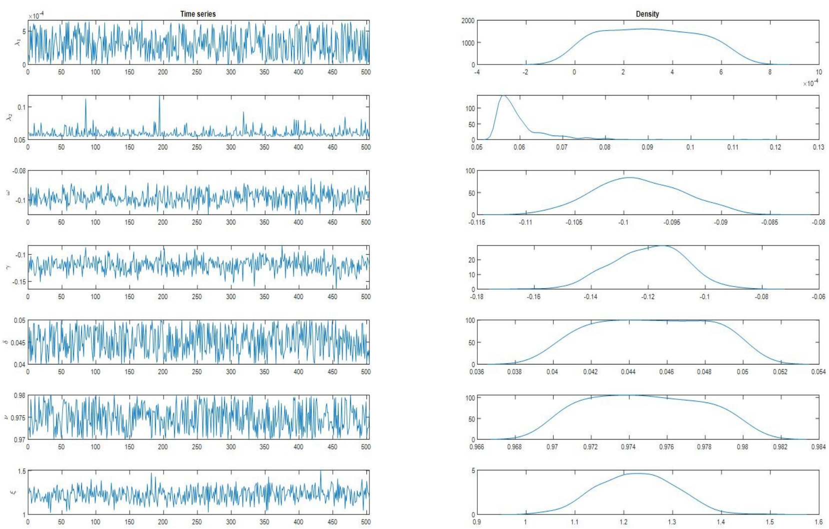


Figure A.4: Convergence draws and density plots of posterior samples for estimated parameters from EGARCH(1,1,1)-ICAPM-GED Process



A.2 Additional Empirical Study Results

A.2.1 Prior Specification

This section shows prior distribution of all proposed models of all global stock returns including the S&P500, HSI, ASX200, NZX50, FTSE100, Nikkei225 and SET in Table A.2 and Table A.3.

Table A.2: Priors Specification of All Proposed Models in Real-world Stock Markets

	Model/Prior				
Stock Index	*AM1*	*AM2*	*AM3*	*AM4*	*AM5*
S&P500	$\lambda_1 \backsim N\left(0,10^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim B\left(2,900000\right)$	$\lambda_1 \backsim N\left(0,0.01^2\right)$
	$\lambda_2 \backsim N\left(0,1\right)$	$\lambda_2 \backsim N\left(0.04,0.01^2\right)$	$\lambda_2 \backsim \Gamma\left(210,0.00035\right)$	$\lambda_2 \backsim \Gamma\left(118,0.0002\right)$	$\lambda_2, \backsim N\left(0,10\right)$
	$\omega \backsim N\left(0,1\right)$	$\omega \backsim B\left(90,4000\right)$	$\omega \backsim B\left(220,25000\right)$	$\omega \backsim N\left(0.001,0.0044^2\right)$	$\omega \backsim Unif\left(0,1\right)$
	$\gamma \backsim N\left(0,1\right)$	$\alpha^+ \backsim N\left(0,0.01^2\right)$	$\alpha \backsim B\left(40,1000\right)$	$\gamma \backsim N\left(385,10^2\right)$	$\alpha \backsim Unif\left(0,1\right)$
	$\delta \backsim B\left(2,5\right)$	$\alpha^- \backsim \Gamma\left(250,0.0004\right)$	$\gamma \backsim \Gamma\left(10,0.0006\right)$	$\delta \backsim \Gamma\left(83,48.5\right)$	$\nu \backsim Unif\left(0,1\right)$
	$\nu \backsim B\left(2,4\right)$	$\nu \backsim \Gamma\left(70,1590\right)$	$\nu \backsim \Gamma\left(70,1570\right)$	$\nu \backsim \Gamma\left(1,0.00129\right)$	$\xi^L \backsim Unif\left(0,1\right)$
	$\xi^L \backsim Unif\left(0,0.5\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi \backsim \Gamma\left(345,0.02\right)$	$\xi^R \backsim Unif\left(0,1\right)$
	$\xi^R \backsim Unif\left(0,0.5\right)$	$\xi^R \backsim B\left(30,5\right)$	$\xi^R \backsim B\left(15,55\right)$	-	-
HSI	$\lambda_1 \backsim N\left(0,10^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim B\left(2,900000\right)$	$\lambda_1 \backsim N\left(0,0.01^2\right)$
	$\lambda_2 \backsim N\left(0,1\right)$	$\lambda_2 \backsim N\left(0.04,0.01^2\right)$	$\lambda_2 \backsim \Gamma\left(110,0.00035\right)$	$\lambda_2 \backsim \Gamma\left(105,0.0002\right)$	$\lambda_2, \backsim N\left(0,10\right)$
	$\omega \backsim N\left(0,1\right)$	$\omega \backsim B\left(90,4000\right)$	$\omega \backsim B\left(220,25000\right)$	$\omega \backsim N\left(0.001,0.00139^2\right)$	$\omega \backsim Unif\left(0,1\right)$
	$\gamma \backsim N\left(0,1\right)$	$\alpha^+ \backsim N\left(0,0.01^2\right)$	$\alpha \backsim B\left(40,1000\right)$	$\gamma \backsim N\left(385,10^2\right)$	$\alpha \backsim Unif\left(0,1\right)$
	$\delta \backsim B\left(2,5\right)$	$\alpha^- \backsim \Gamma\left(250,0.0004\right)$	$\gamma \backsim \Gamma\left(10,0.0006\right)$	$\delta \backsim \Gamma\left(79,48.5\right)$	$\nu \backsim Unif\left(0,1\right)$
	$\nu \backsim B\left(2,4\right)$	$\nu \backsim \Gamma\left(70,1590\right)$	$\nu \backsim \Gamma\left(70,1570\right)$	$\nu \backsim \Gamma\left(121,10\right)$	$\xi^L \backsim Unif\left(0,1\right)$
	$\xi^L \backsim Unif\left(0,0.5\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi \backsim \Gamma\left(1,0.02\right)$	$\xi^R \backsim Unif\left(0,1\right)$
	$\xi^R \backsim Unif\left(0,0.5\right)$	$\xi^R \backsim B\left(30,5\right)$	$\xi^R \backsim B\left(15,55\right)$	-	-
ASX200	$\lambda_1 \backsim N\left(0,10^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim B\left(2,900000\right)$	$\lambda_1 \backsim N\left(0,0.01^2\right)$
	$\lambda_2 \backsim N\left(0,1\right)$	$\lambda_2 \backsim N\left(0.04,0.01^2\right)$	$\lambda_2 \backsim \Gamma\left(154,0.00035\right)$	$\lambda_2 \backsim \Gamma\left(125,0.0002\right)$	$\lambda_2, \backsim N\left(0,10\right)$
	$\omega \backsim N\left(0,1\right)$	$\omega \backsim B\left(90,4000\right)$	$\omega \backsim B\left(220,25000\right)$	$\omega \backsim N\left(0.001,0.0027^2\right)$	$\omega \backsim Unif\left(0,1\right)$
	$\gamma \backsim N\left(0,1\right)$	$\alpha^+ \backsim N\left(0,0.01^2\right)$	$\alpha \backsim B\left(40,1000\right)$	$\gamma \backsim N\left(385,10^2\right)$	$\alpha \backsim Unif\left(0,1\right)$
	$\delta \backsim B\left(2,5\right)$	$\alpha^- \backsim \Gamma\left(250,0.0004\right)$	$\gamma \backsim \Gamma\left(10,0.0006\right)$	$\delta \backsim \Gamma\left(80,48.5\right)$	$\nu \backsim Unif\left(0,1\right)$
	$\nu \backsim B\left(2,4\right)$	$\nu \backsim \Gamma\left(70,1590\right)$	$\nu \backsim \Gamma\left(70,1570\right)$	$\nu \backsim \Gamma\left(121,10\right)$	$\xi^L \backsim Unif\left(0,1\right)$
	$\xi^L \backsim Unif\left(0,0.5\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi \backsim \Gamma\left(90,0.02\right)$	$\xi^R \backsim Unif\left(0,1\right)$
	$\xi^R \backsim Unif\left(0,0.5\right)$	$\xi^R \backsim B\left(30,5\right)$	$\xi^R \backsim B\left(15,55\right)$	-	-
FTSE100	$\lambda_1 \backsim N\left(0,10^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim B\left(2,900000\right)$	$\lambda_1 \backsim N\left(0,0.01^2\right)$
	$\lambda_2 \backsim N\left(0,1\right)$	$\lambda_2 \backsim N\left(0.04,0.01^2\right)$	$\lambda_2 \backsim \Gamma\left(85.3,0.00035\right)$	$\lambda_2 \backsim \Gamma\left(80,0.0002\right)$	$\lambda_2, \backsim N\left(0,10\right)$
	$\omega \backsim N\left(0,1\right)$	$\omega \backsim B\left(90,4000\right)$	$\omega \backsim B\left(220,25000\right)$	$\omega \backsim N\left(0.001,0.00315^2\right)$	$\omega \backsim Unif\left(0,1\right)$
	$\gamma \backsim N\left(0,1\right)$	$\alpha^+ \backsim N\left(0,0.01^2\right)$	$\alpha \backsim B\left(40,1000\right)$	$\gamma \backsim N\left(385,10^2\right)$	$\alpha \backsim Unif\left(0,1\right)$
	$\delta \backsim B\left(2,5\right)$	$\alpha^- \backsim \Gamma\left(250,0.0004\right)$	$\gamma \backsim \Gamma\left(10,0.0006\right)$	$\delta \backsim \Gamma\left(81.5,48.5\right)$	$\nu \backsim Unif\left(0,1\right)$
	$\nu \backsim B\left(2,4\right)$	$\nu \backsim \Gamma\left(70,1590\right)$	$\nu \backsim \Gamma\left(70,1570\right)$	$\nu \backsim \Gamma\left(121,10\right)$	$\xi^L \backsim Unif\left(0,1\right)$
	$\xi^L \backsim Unif\left(0,0.5\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi \backsim \Gamma\left(90,0.02\right)$	$\xi^R \backsim Unif\left(0,1\right)$
	$\xi^R \backsim Unif\left(0,0.5\right)$	$\xi^R \backsim B\left(30,5\right)$	$\xi^R \backsim B\left(15,55\right)$	-	-

Note that *AM1* is EGARCH(1,1,1)-ICAPM-Mixture model, *AM2* is LogGARCH(1,1)-ICAPM-Mixture model, *AM3* is GJR-GARCH(1,1)-ICAPM-Mixture model, *AM4* is EGARCH(1,1,1)-ICAPM-GED model and *AM5* is GARCH(1,1)-ICAPM-Mixture model

Table A.3: Priors Specification of All Proposed Models in Real-world Stock Markets (continued)

	Model/Prior				
Stock Index	*AM1*	*AM2*	*AM3*	*AM4*	*AM5*
N225	$\lambda_1 \backsim N\left(0,10^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim B\left(2,900000\right)$	$\lambda_1 \backsim N\left(0,0.01^2\right)$
	$\lambda_2 \backsim N\left(0,1\right)$	$\lambda_2 \backsim N\left(0.04,0.01^2\right)$	$\lambda_2 \backsim \Gamma\left(93,0.00035\right)$	$\lambda_2 \backsim \Gamma\left(150,0.0002\right)$	$\lambda_2, \backsim N\left(0,10\right)$
	$\omega \backsim N\left(0,1\right)$	$\omega \backsim B\left(90,4000\right)$	$\omega \backsim B\left(220,25000\right)$	$\omega \backsim N\left(0.001,0.00315^2\right)$	$\omega \backsim Unif\left(0,1\right)$
	$\gamma \backsim N\left(0,1\right)$	$\alpha^+ \backsim N\left(0,0.01^2\right)$	$\alpha \backsim B\left(40,1000\right)$	$\gamma \backsim N\left(385,10^2\right)$	$\alpha \backsim Unif\left(0,1\right)$
	$\delta \backsim B\left(2,5\right)$	$\alpha^- \backsim \Gamma\left(250,0.0004\right)$	$\gamma \backsim \Gamma\left(10,0.0006\right)$	$\delta \backsim \Gamma\left(81.5,48.5\right)$	$\nu \backsim Unif\left(0,1\right)$
	$\nu \backsim B\left(2,4\right)$	$\nu \backsim \Gamma\left(70,1590\right)$	$\nu \backsim \Gamma\left(70,1570\right)$	$\nu \backsim \Gamma\left(121,10\right)$	$\xi^L \backsim Unif\left(0,1\right)$
	$\xi^L \backsim Unif\left(0,0.5\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi \backsim \Gamma\left(200,0.02\right)$	$\xi^R \backsim Unif\left(0,1\right)$
	$\xi^R \backsim Unif\left(0,0.5\right)$	$\xi^R \backsim B\left(30,5\right)$	$\xi^R \backsim B\left(15,55\right)$	-	-
SET	$\lambda_1 \backsim N\left(0,10^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim B\left(2,900000\right)$	$\lambda_1 \backsim N\left(0,0.01^2\right)$
	$\lambda_2 \backsim N\left(0,1\right)$	$\lambda_2 \backsim N\left(0.04,0.01^2\right)$	$\lambda_2 \backsim \Gamma\left(100,0.00035\right)$	$\lambda_2 \backsim \Gamma\left(105,0.0002\right)$	$\lambda_2, \backsim N\left(0,10\right)$
	$\omega \backsim N\left(0,1\right)$	$\omega \backsim B\left(90,4000\right)$	$\omega \backsim B\left(220,25000\right)$	$\omega \backsim N\left(0.001,0.0044^2\right)$	$\omega \backsim Unif\left(0,1\right)$
	$\gamma \backsim N\left(0,1\right)$	$\alpha^+ \backsim N\left(0,0.01^2\right)$	$\alpha \backsim B\left(40,1000\right)$	$\gamma \backsim N\left(385,10^2\right)$	$\alpha \backsim Unif\left(0,1\right)$
	$\delta \backsim B\left(2,5\right)$	$\alpha^- \backsim \Gamma\left(250,0.0004\right)$	$\gamma \backsim \Gamma\left(10,0.0006\right)$	$\delta \backsim \Gamma\left(84,48.5\right)$	$\nu \backsim Unif\left(0,1\right)$
	$\nu \backsim B\left(2,4\right)$	$\nu \backsim \Gamma\left(70,1590\right)$	$\nu \backsim \Gamma\left(70,1570\right)$	$\nu \backsim \Gamma\left(121,10\right)$	$\xi^L \backsim Unif\left(0,1\right)$
	$\xi^L \backsim Unif\left(0,0.5\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi \backsim \Gamma\left(1000,0.002\right)$	$\xi^R \backsim Unif\left(0,1\right)$
	$\xi^R \backsim Unif\left(0,0.5\right)$	$\xi^R \backsim B\left(30,5\right)$	$\xi^R \backsim B\left(15,55\right)$	-	-
NZX50	$\lambda_1 \backsim N\left(0,10^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim N\left(0,0.0001^2\right)$	$\lambda_1 \backsim B\left(2,900000\right)$	$\lambda_1 \backsim N\left(0,0.01^2\right)$
	$\lambda_2 \backsim N\left(0,1\right)$	$\lambda_2 \backsim N\left(0.04,0.01^2\right)$	$\lambda_2 \backsim \Gamma\left(154,0.00035\right)$	$\lambda_2 \backsim \Gamma\left(80,0.0002\right)$	$\lambda_2, \backsim N\left(0,10\right)$
	$\omega \backsim N\left(0,1\right)$	$\omega \backsim B\left(90,4000\right)$	$\omega \backsim B\left(220,25000\right)$	$\omega \backsim N\left(0.001,0.00315^2\right)$	$\omega \backsim Unif\left(0,1\right)$
	$\gamma \backsim N\left(0,1\right)$	$\alpha^+ \backsim N\left(0,0.01^2\right)$	$\alpha \backsim B\left(40,1000\right)$	$\gamma \backsim N\left(385,10^2\right)$	$\alpha \backsim Unif\left(0,1\right)$
	$\delta \backsim B\left(2,5\right)$	$\alpha^- \backsim \Gamma\left(250,0.0004\right)$	$\gamma \backsim \Gamma\left(10,0.0006\right)$	$\delta \backsim \Gamma\left(80,48.5\right)$	$\nu \backsim Unif\left(0,1\right)$
	$\nu \backsim B\left(2,4\right)$	$\nu \backsim \Gamma\left(70,1590\right)$	$\nu \backsim \Gamma\left(70,1570\right)$	$\nu \backsim \Gamma\left(121,10\right)$	$\xi^L \backsim Unif\left(0,1\right)$
	$\xi^L \backsim Unif\left(0,0.5\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi^L \backsim \Gamma\left(70,0.001\right)$	$\xi \backsim \Gamma\left(135,0.002\right)$	$\xi^R \backsim Unif\left(0,1\right)$
	$\xi^R \backsim Unif\left(0,0.5\right)$	$\xi^R \backsim B\left(30,5\right)$	$\xi^R \backsim B\left(15,55\right)$	-	-

Note that *AM1* is EGARCH(1,1,1)-ICAPM-Mixture model, *AM2* is LogGARCH(1,1)-ICAPM-Mixture model, *AM3* is GJR-GARCH(1,1)-ICAPM-Mixture model, *AM4* is EGARCH(1,1,1)-ICAPM-GED model and *AM5* is GARCH(1,1)-ICAPM-Mixture model

A.2.2 Posterior Samples of Model Parameters

This section shows convergence draws and density plots of posterior samples for all proposed models including the EGARCH(1,1,1)-ICAPM-Mixture process, GJR-GARCH(1,1)-ICAPM-Mixture process, LogGARCH(1,1)-ICAPM-Mixture process, GARCH(1,1)-ICAPM-Mixture process and EGARCH(1,1,1)-ICAPM-GED process of all global stock returns (the S&P500, HSI, ASX200, FTSE100, N225, SET and NZX50) as follows:

Figure A.5 - Figure A.10 show convergence draws and its density plots of posterior samples for estimated parameters from EGARCH(1,1,1)-ICAPM-Mixture process of the HSI, ASX200, FTSE100, N225, SET and NZX50.

Figure A.11 - Figure A.17 show convergence draws and its density plots of posterior samples for estimated parameters from GJR-GARCH(1,1)-ICAPM-Mixture process of the S&P500, HSI, ASX200, FTSE100, N225, SET and NZX50.

Figure A.18 - Figure A.24 show convergence draws and its density plots of posterior samples for estimated parameters from LogGARCH(1,1)-ICAPM-Mixture process of the S&P500, HSI, ASX200, FTSE100, N225, SET and NZX50.

Figure A.25 - Figure A.31 show convergence draws and its density plots of posterior samples for estimated parameters from GARCH(1,1)-ICAPM-Mixture process of the S&P500, HSI, ASX200, FTSE100, N225, SET and NZX50.

Figure A.32 - Figure A.38 show convergence draws and its density plots of posterior samples for estimated parameters from EGARCH(1,1,1)-ICAPM-GED process of the S&P500, HSI, ASX200, FTSE100, N225, SET and NZX50.

Figure A.5: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-Mixture Process of the HSI Returns

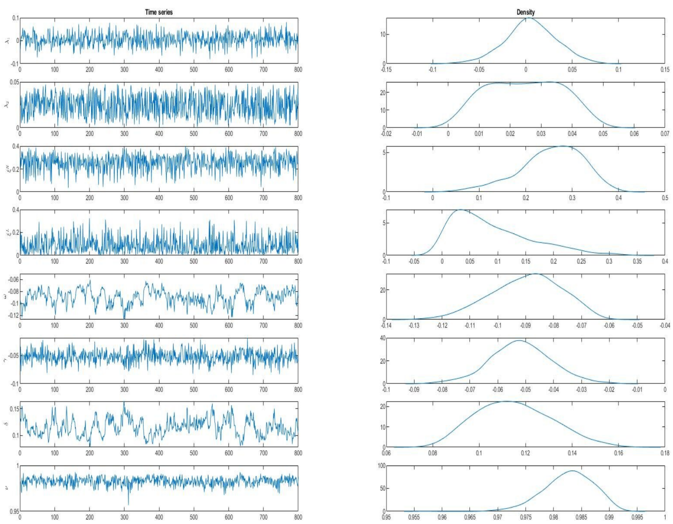


Figure A.6: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-Mixture Process of the ASX200 Returns

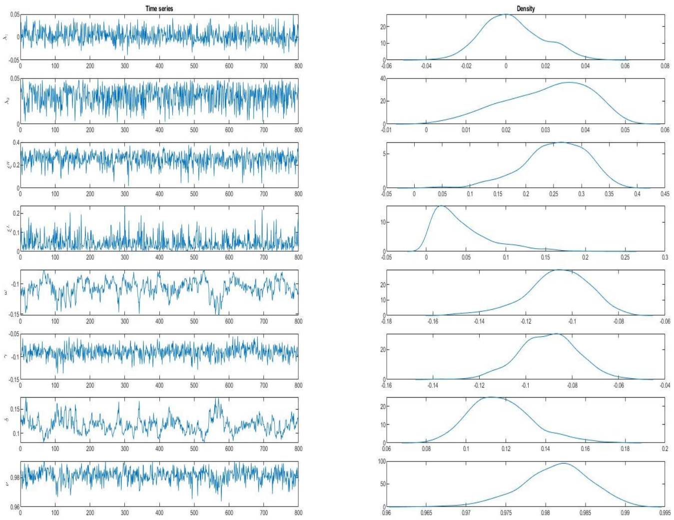


Figure A.7: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-Mixture Process of the FTSE100 Returns

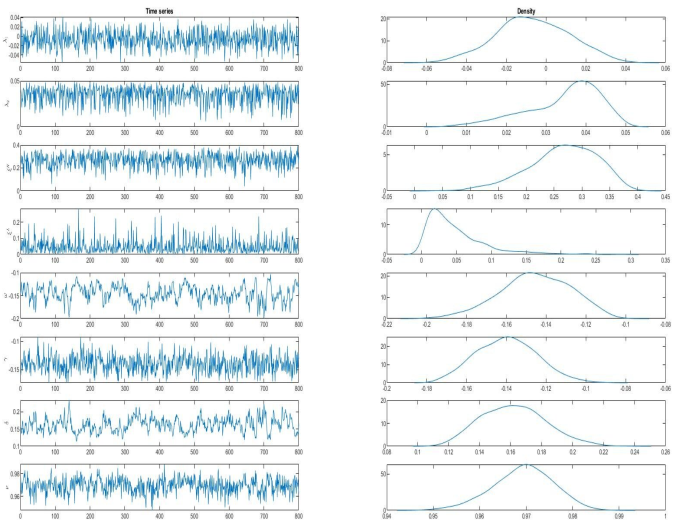


Figure A.8: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-Mixture Process of the Nikkei225 Returns

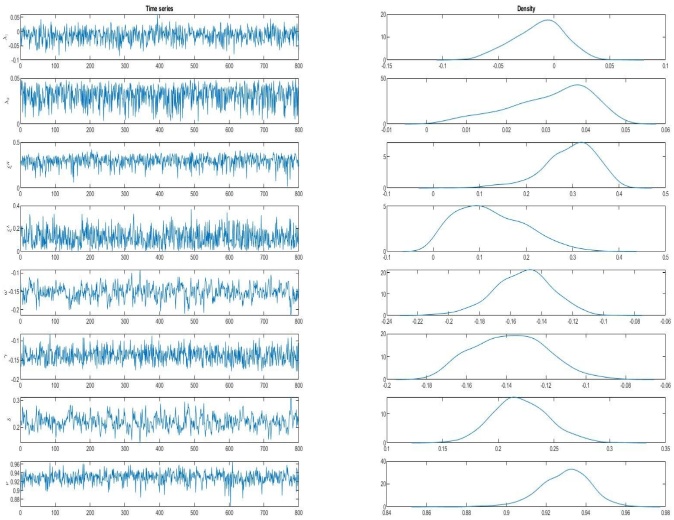


Figure A.9: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-Mixture Process of the SET Returns

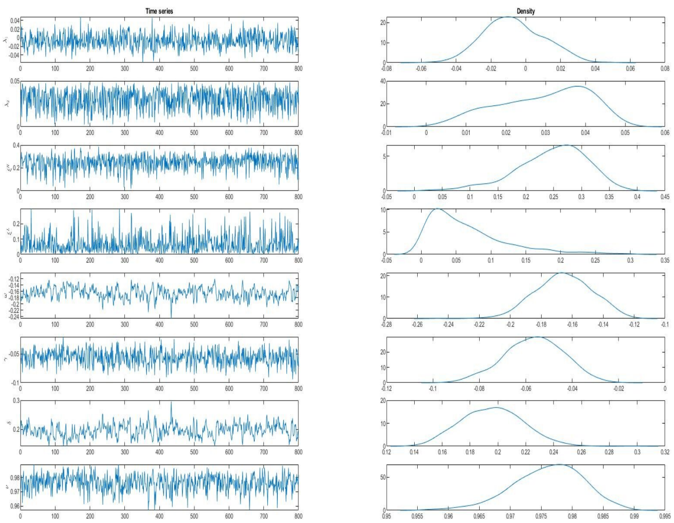


Figure A.10: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-Mixture Process of NZX50 Returns

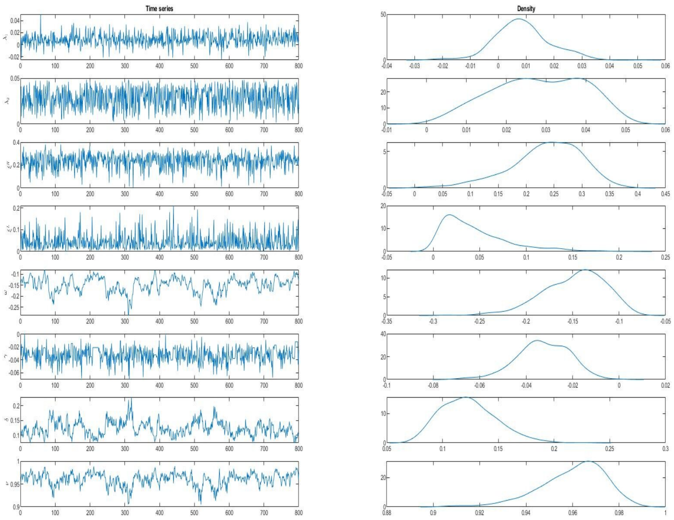


Figure A.11: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GJR-GARCH(1,1)-ICAPM-Mixture Process of the S&P500 Returns

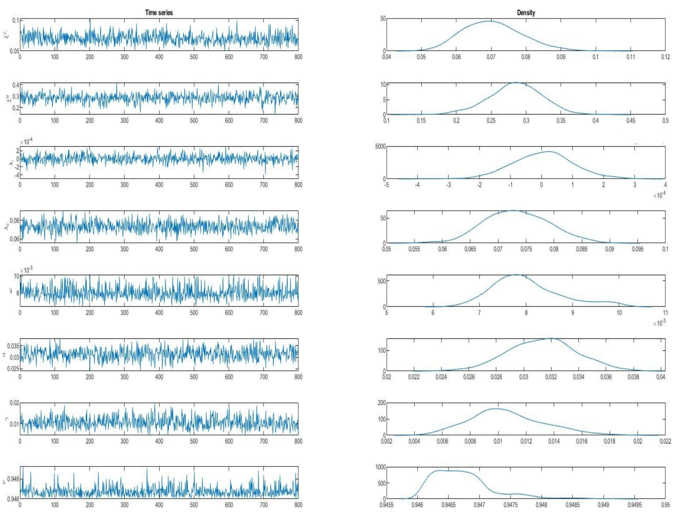


Figure A.12: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GJR-GARCH(1,1)-ICAPM-Mixture Process of the HSI Returns

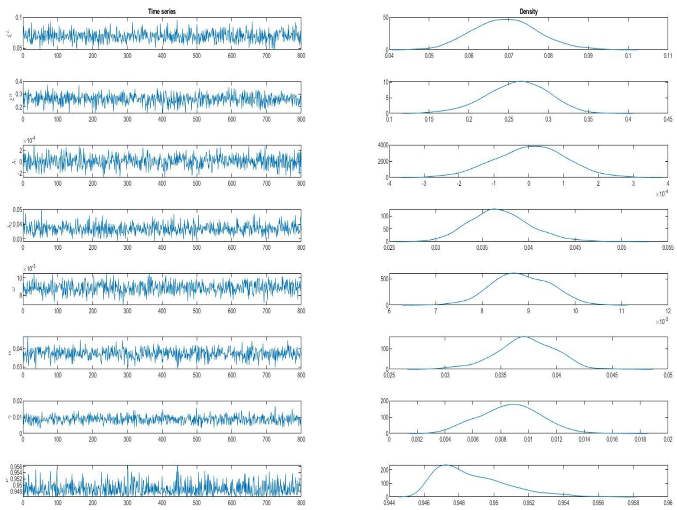


Figure A.13: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GJR-GARCH(1,1)-ICAPM-Mixture Process of the ASX200 Returns

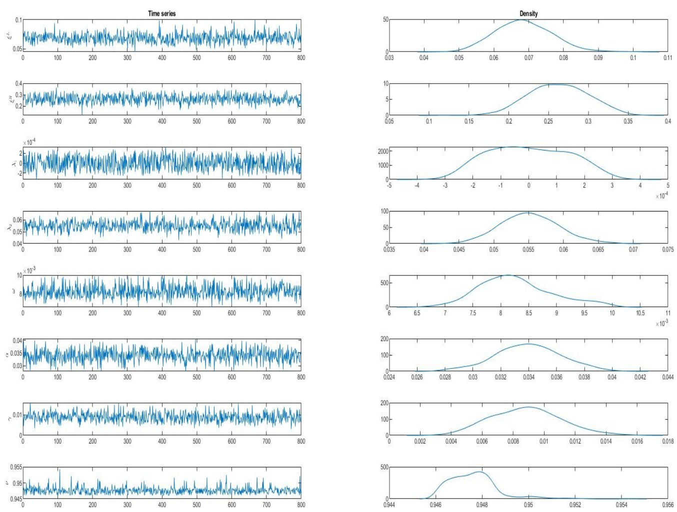


Figure A.14: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GJR-GARCH(1,1)-ICAPM-Mixture Process of the FTSE100 Returns

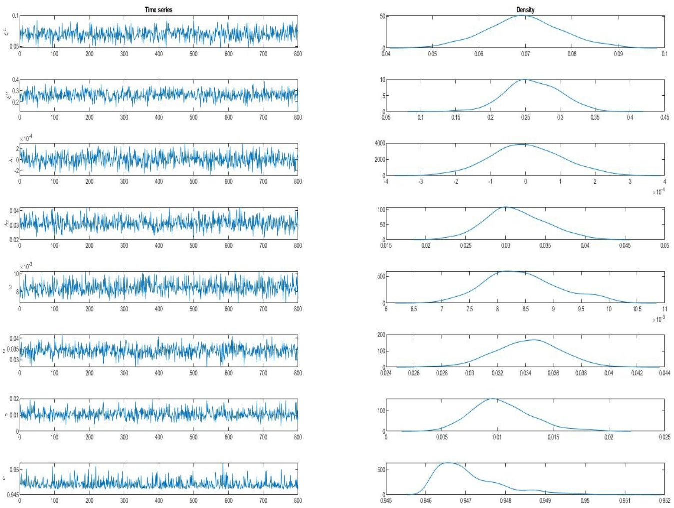


Figure A.15: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GJR-GARCH(1,1)-ICAPM-Mixture Process of the Nikkei225 Returns

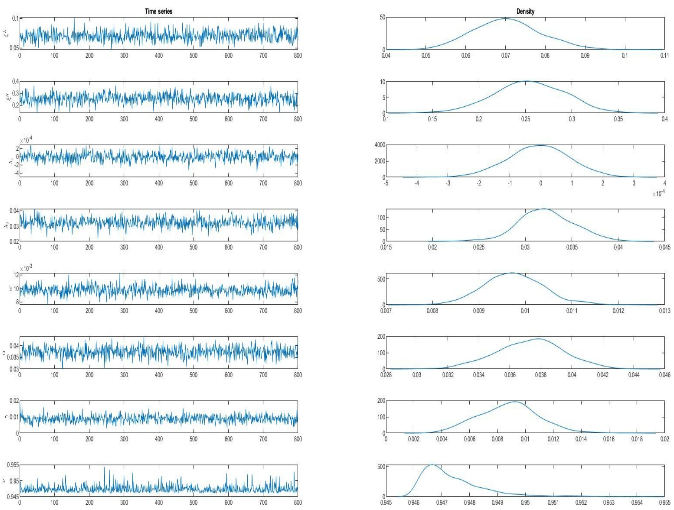


Figure A.16: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GJR-GARCH(1,1)-ICAPM-Mixture Process of the SET Returns

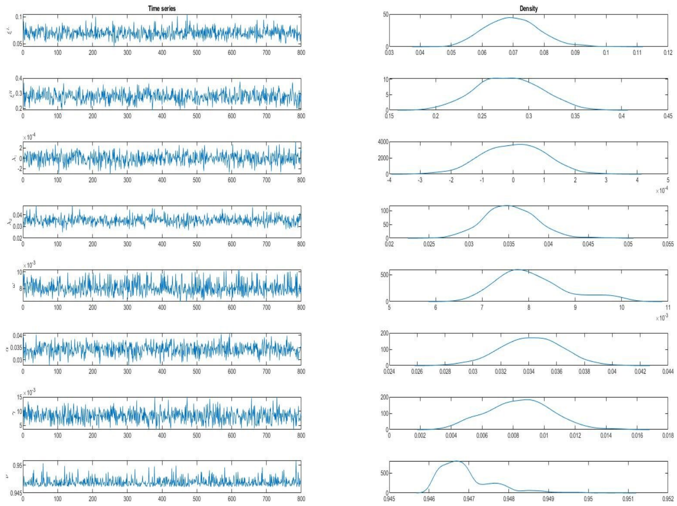


Figure A.17: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GJR-GARCH(1,1)-ICAPM-Mixture Process of NZX50 Returns

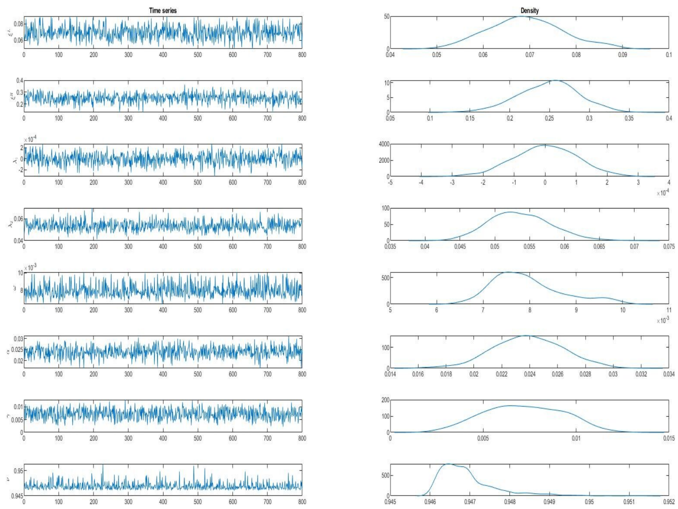


Figure A.18: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from LogGARCH(1,1)-ICAPM-Mixture Process of the S&P500 Returns

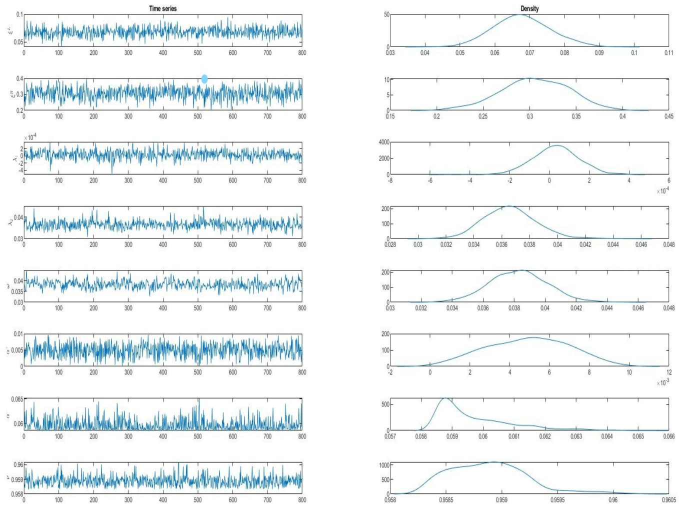


Figure A.19: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from LogGARCH(1,1)-ICAPM-Mixture Process of the HSI Returns

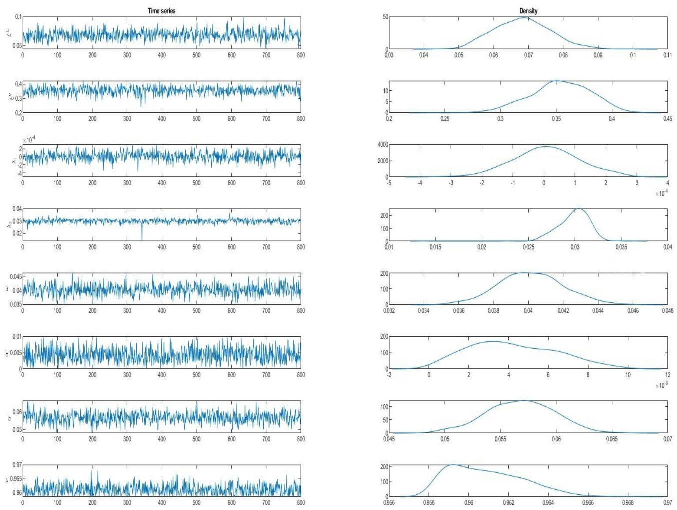


Figure A.20: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from LogGARCH(1,1)-ICAPM-Mixture Process of the ASX200 Returns

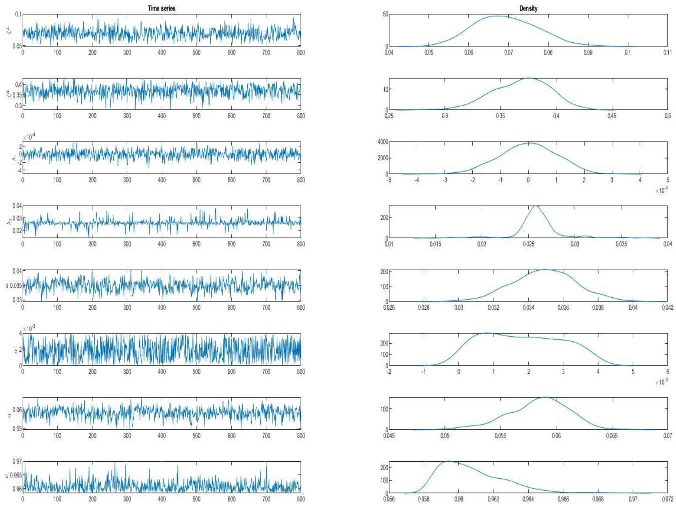


Figure A.21: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from LogGARCH(1,1)-ICAPM-Mixture Process of the FTSE100 Returns

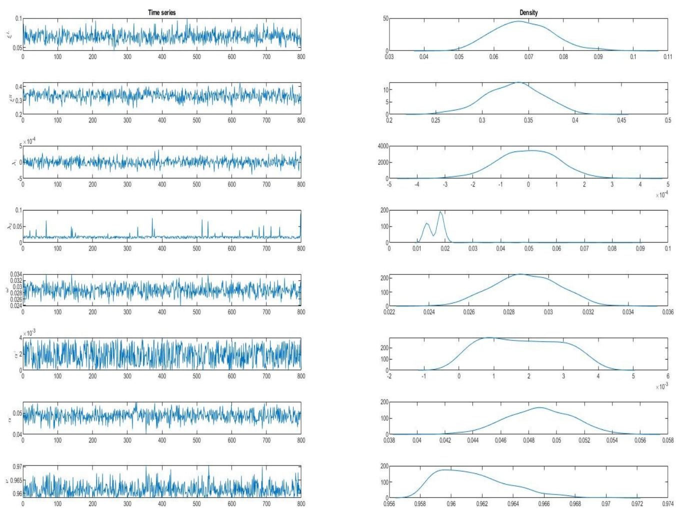


Figure A.22: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from LogGARCH(1,1)-ICAPM-Mixture Process of the Nikkei225 Returns

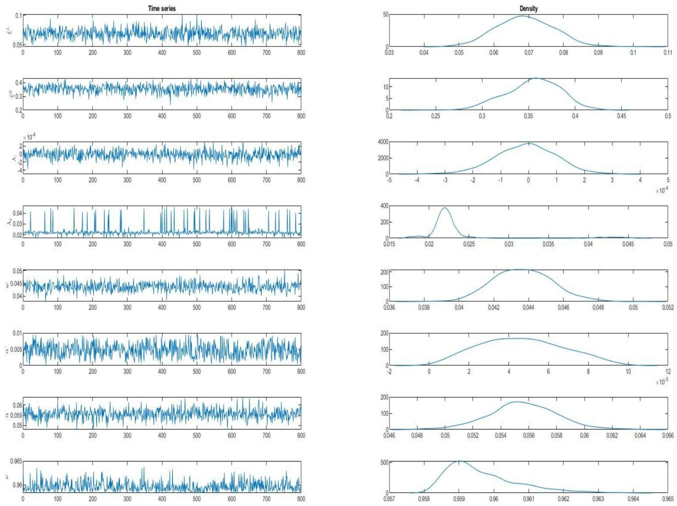


Figure A.23: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from LogGARCH(1,1)-ICAPM-Mixture Process of the SET Returns

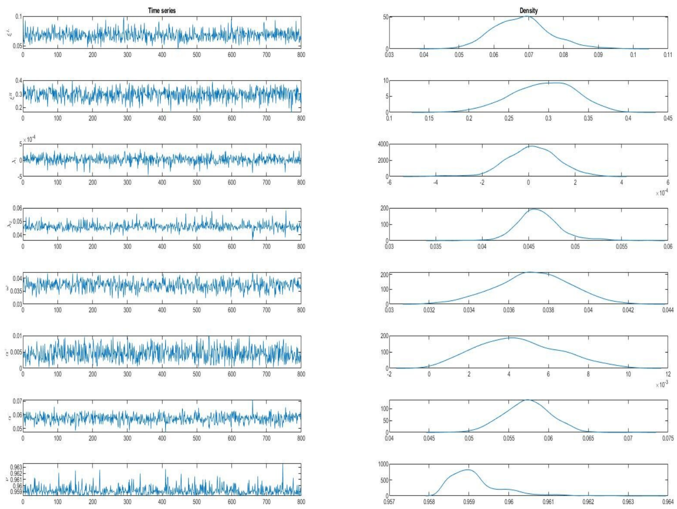


Figure A.24: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from LogGARCH(1,1)-ICAPM-Mixture Process of NZX50 Returns

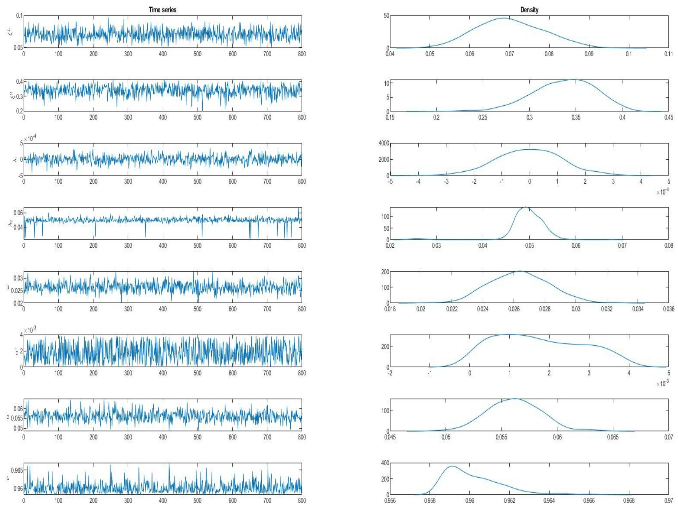


Figure A.25: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GARCH(1,1)-ICAPM-Mixture of the S&P500 Returns

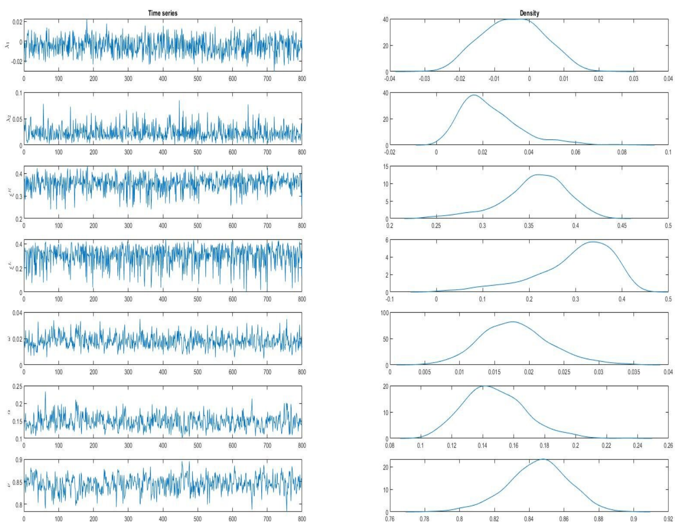


Figure A.26: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GARCH(1,1)-ICAPM-Mixture of the HSI Returns

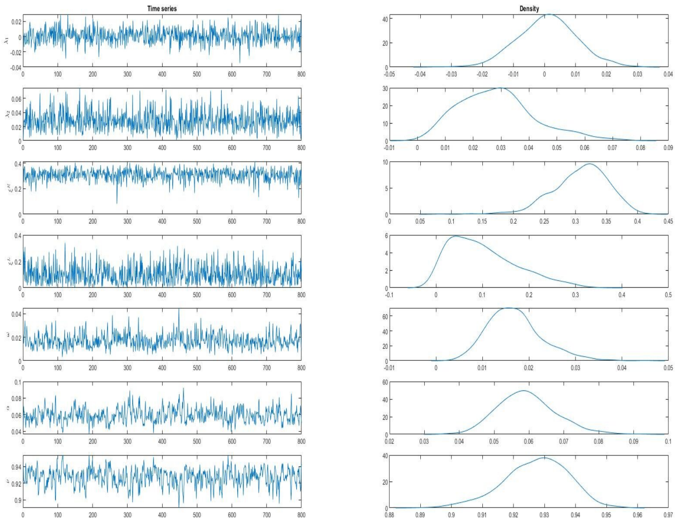


Figure A.27: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GARCH(1,1)-ICAPM-Mixture of the ASX200 Returns

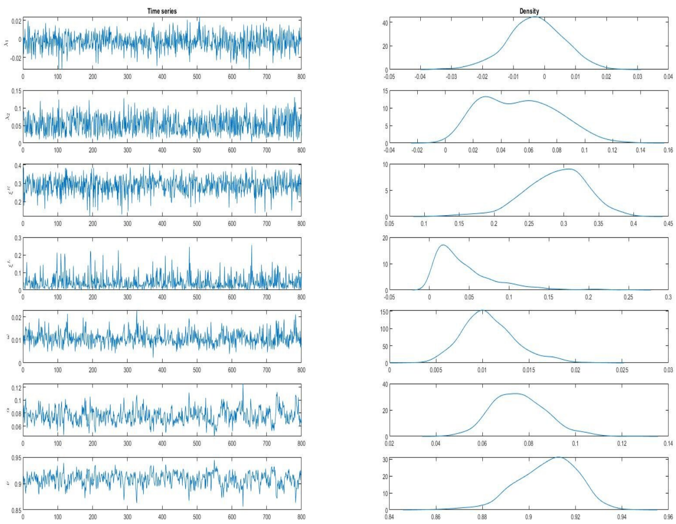


Figure A.28: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GARCH(1,1)-ICAPM-Mixture of the FTSE100 Returns

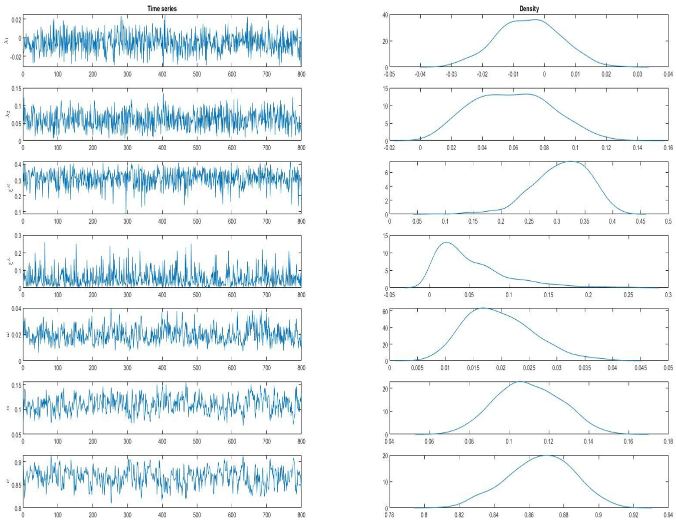


Figure A.29: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GARCH(1,1)-ICAPM-Mixture of the Nikkei225 Returns

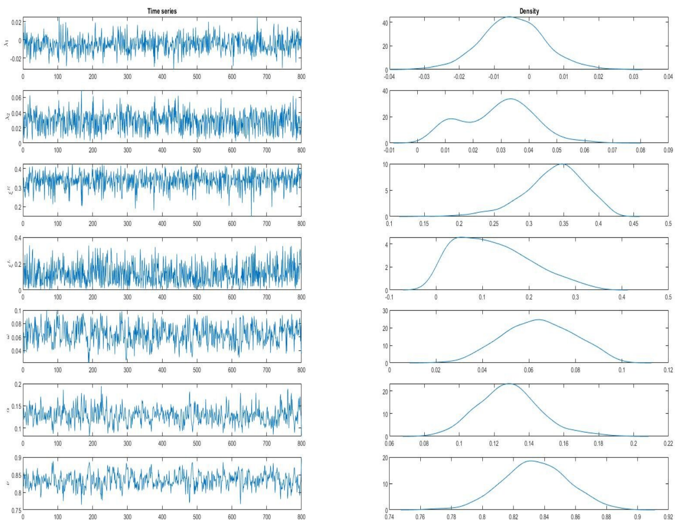


Figure A.30: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GARCH(1,1)-ICAPM-Mixture of the SET Returns

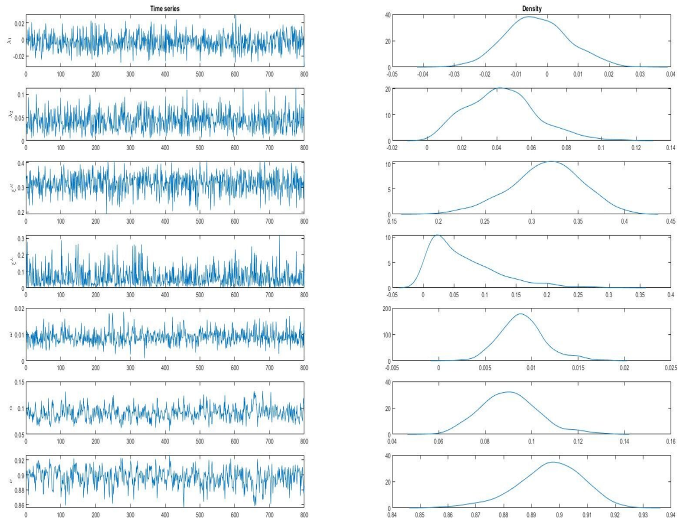


Figure A.31: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from GARCH(1,1)-ICAPM-Mixture of NZX50 Returns

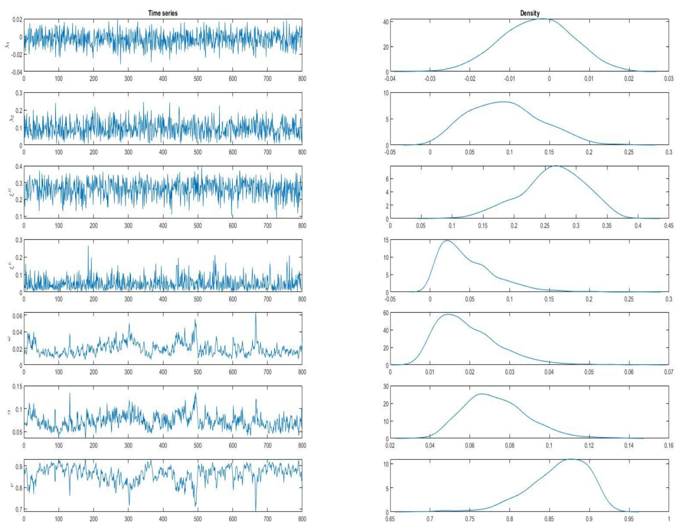


Figure A.32: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-GED of the S&P500 Returns

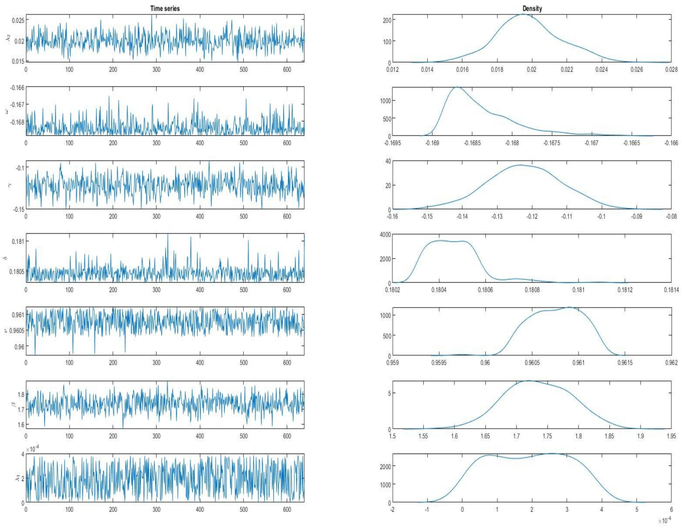


Figure A.33: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-GED of the HSI Returns

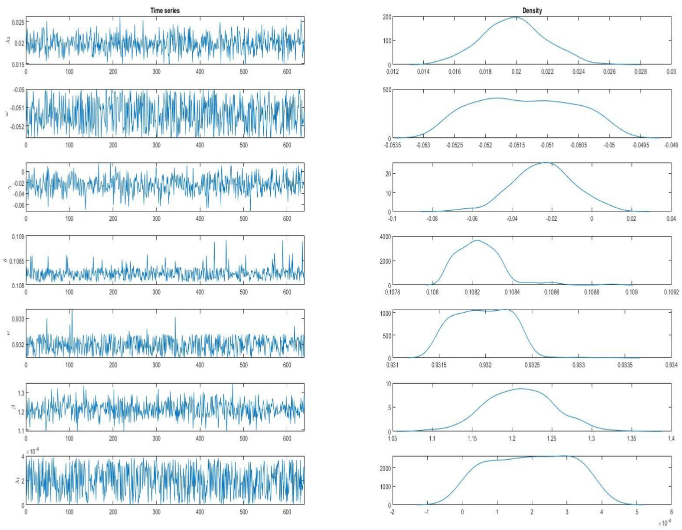


Figure A.34: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-GED of the ASX200 Returns

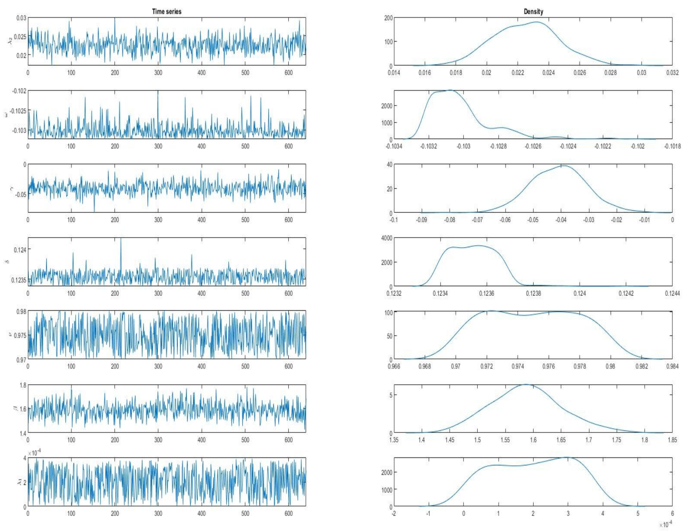


Figure A.35: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-GED of the FTSE100 Returns

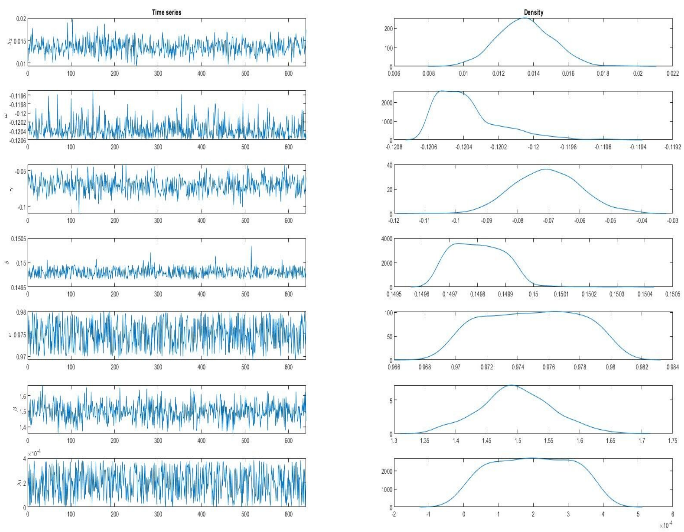


Figure A.36: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-GED of the Nikkei225 Returns

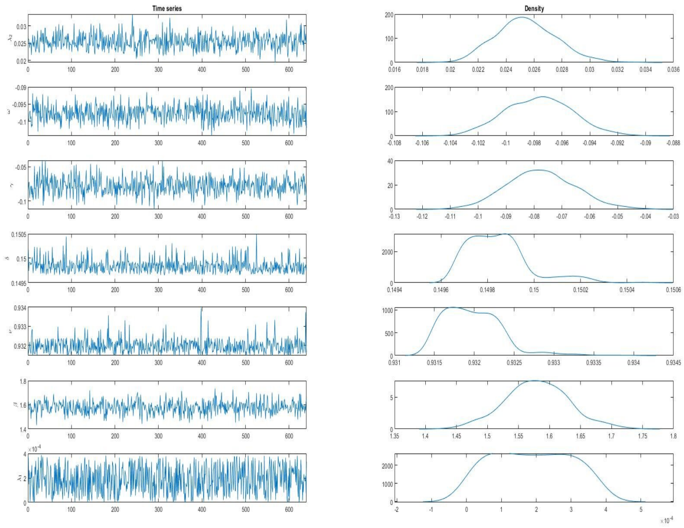


Figure A.37: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-GED of the SET Returns

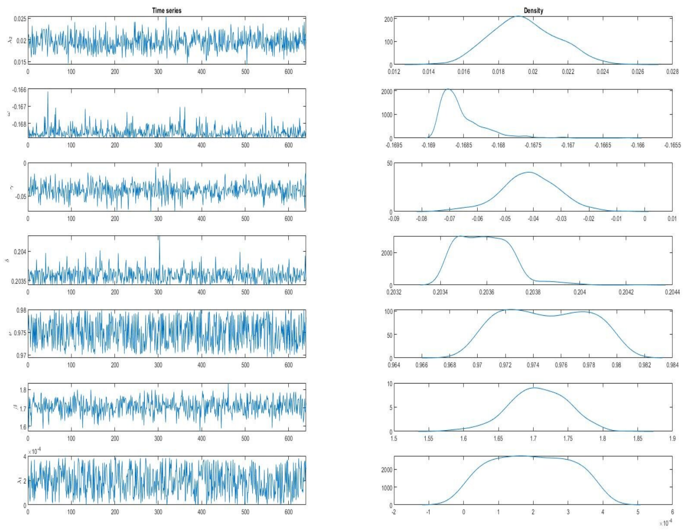
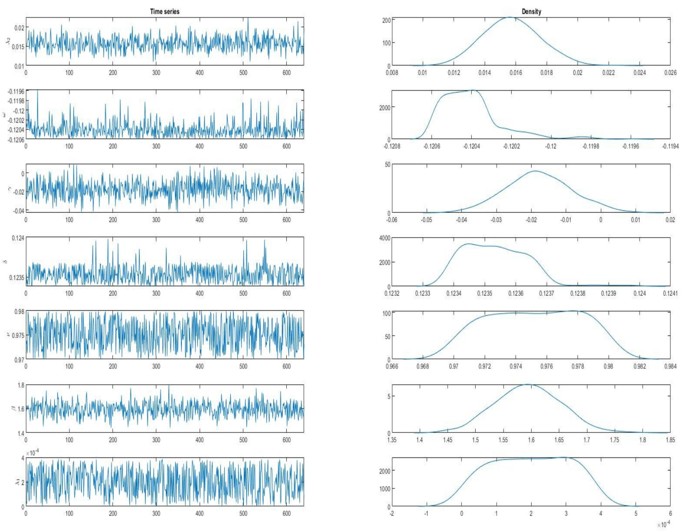


Figure A.38: Convergence Draws and Density Plots of Posterior Samples for Estimated Parameters from EGARCH(1,1,1)-ICAPM-GED of NZX50 Returns



Appendix B

Chapter 4 Additional Information

B.1 Simulation Study

B.1.1 Simulation Experiment

This paper sets up a comprehensive simulation study of the proposed Bayesian inference and machine learning in n-dimensional regular vine copula model. Seven different scenarios were designed which consist of two scenarios in a 15-dimensional R-vine model and five scenarios in a five-dimensional R-vine model with a simulated sample size of 1,250 approximate five years of daily data, and repeated up to 79 times each. Three estimation methods and five algorithms were applied and compared to all different scenarios in the R-vine model in order to measure the estimator performance. Five algorithms consist of QMLE, rw-MH sampling in Bayesian inference and VBDA (DA=0,1,2) in Bayesian inference and machine learning. In addition, this paper investigates and proposes an efficient variational approximation $q(\theta, \mathbf{u})$ in variational Bayes. For more details of an efficient variational approximation see in the last paragraph of this section. The scenarios considered are as follows.

- 15-dimensional R-vine model, pair-copula families are all Gaussian and mixed copulae. See example of 15-dimensions of mixed copulae in R-vines, $\mathcal{T}_{15-Dim}^{Mixed}$, $\mathcal{P}_{15-Dim}^{Mixed}$, $\mathcal{M}_{15-Dim}^{Mixed}$, in matrix B.1.1, matrix B.1.2 and matrix B.1.3, respectively.
- Five-dimensional R-vine model, pair-copula families are all Gaussian, all Student-*t*, all Gumbel, all Frank and mixed copulae. Mixed copulae use different families for each pair-copula in R-vines. Pair-copula families in mixed R-vines consist of a one-parameter copula; Gaussian, Student-*t*, Frank, Clayton, Gumbel, B5/Joe, Galambos copulae and a two-parameters copula; and BB1 and BB3 copulae. See example of mixed R-vines $\mathcal{T}_{5-Dim}^{Mixed}$, $\mathcal{P}_{5-Dim}^{Mixed}$ and $\mathcal{M}_{5-Dim}^{Mixed}$ in the matrices B.1.4, B.1.5 and B.1.6, respectively.

Matrix $\mathcal{P}$ is designed to decrease in dependency when the tree increasing, which is typically true for applications in financial data. For instance, the absolute correlations of the Gaussian copula are decreased in higher trees and close to zero in $(n-1)^{th}$ trees, from 0.5 to -0.05. Degrees of freedom of the Student-*t* copula are increased and start with 3 in the first tree and end with 25 in the last tree.

$$\mathcal{T}_{15-Dim}^{Mixed} = \begin{pmatrix} BB1 \\ BB1 & BB1 \\ BB1 & BB1 & BB1 \\ BB3 & BB3 & BB3 & BB3 \\ BB3 & BB3 & BB3 & BB3 & BB3 \\ C & C & C & C & C & C \\ Gu & Gu & Gu & Gu & Gu & Gu & Gu \\ Ga & Ga & Ga & Ga & Ga & Ga & Ga & Ga \\ B5 & B5 & B5 & B5 & B5 & B5 & B5 & B5 & B5 \\ N & N & N & N & N & N & N & N & N & N \\ N & N & N & N & N & N & N & N & N & N & N \\ F & F & F & F & F & F & F & F & F & F & F & F \\ F & F & F & F & F & F & F & F & F & F & F & F & F \\ t & t & t & t & t & t & t & t & t & t & t & t & t & t \end{pmatrix} \tag{B.1.1}$$

$$\mathcal{P}_{15-Dim}^{Mixed} =$$

$$\begin{pmatrix} [1.5,0.5] \\ [1.5,0.5] & [1.50,0.5] \\ [1.5,0.5] & [1.50,0.5] & [1.5,0.5] \\ [0.5,1.5] & [0.5,1.5] & [0.5,1.5] & [0.5,1.5] \\ [0.5,1.5] & [0.5,1.5] & [0.5,1.5] & [0.5,1.5] & [0.5,1.5] \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ -0.3 & -0.3 & -0.3 & -0.3 & -0.3 & -0.3 & -0.3 & -0.3 & -0.3 & -0.3 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\ 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 & 2.5 \\ 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & 1.5 \\ [-0.7,25] & [-0.65,25] & [-0.6,25] & [-0.55,25] & [-0.5,25] & [0.45,25] & [0.5,25] & [0.55,25] & [0.6,25] & [0.65,25] \end{pmatrix}$$

$$\left(row_{(1,\ldots,14)} \times column_{(1,\ldots,10)}\right)$$

$$\begin{pmatrix} 0.3 \\ 2.5 & 2.5 \\ 1.5 & 1.5 & 1.5 \\ [0.7,25] & [0.75,25] & [0.8,25] & [0.85,25] \end{pmatrix}$$

$$\left(row_{(11,\ldots,14)} \times column_{(11,\ldots,14)}\right) \tag{B.1.2}$$

$$\mathcal{M}_{15-Dim}^{Mixed} = \begin{pmatrix} 15 \\ 14 & 14 \\ 11 & 11 & 13 \\ 10 & 13 & 11 & 12 \\ 12 & 10 & 10 & 11 & 11 \\ 9 & 8 & 12 & 10 & 9 & 10 \\ 7 & 5 & 9 & 8 & 7 & 9 & 9 \\ 6 & 4 & 7 & 5 & 6 & 7 & 8 & 8 \\ 13 & 3 & 6 & 4 & 2 & 6 & 5 & 7 & 7 \\ 2 & 1 & 8 & 3 & 4 & 2 & 4 & 6 & 5 & 6 \\ 4 & 2 & 5 & 1 & 5 & 4 & 3 & 2 & 4 & 5 & 5 \\ 5 & 6 & 2 & 2 & 1 & 5 & 1 & 4 & 3 & 4 & 2 & 4 \\ 1 & 9 & 3 & 6 & 3 & 1 & 2 & 5 & 1 & 3 & 4 & 2 & 3 \\ 3 & 7 & 1 & 9 & 10 & 3 & 6 & 1 & 2 & 1 & 1 & 3 & 2 & 2 \\ 8 & 12 & 4 & 7 & 8 & 8 & 7 & 3 & 6 & 2 & 3 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{B.1.3}$$

Abbreviations for pair-copula families are N = Gaussian, t = Student-*t*, Gu = Gumbel, Ga = Galambos, F = Frank and C = Clayton. Note that $[\delta, \zeta]$ in matrix $\mathcal{P}$ for two-parameters copula families, e.g., the Student-*t*, BB1 and BB3 copula function. The variety of bivariate copula function in the R-vine model is worthy of study because it reveals different types of dependence and tail dependence properties. For instance, the Student-*t* copula shows reflection symmetric upper and lower tail dependence. The Frank copula shows tail independence. The Gumbel copula shows only upper tail dependence. The Clayton copula shows only lower tail dependence. The BB family of copulae shows different upper and lower tail dependence.

$$\mathcal{T}_{5-Dim}^{Mixed} = \begin{pmatrix} Gu \\ F & N \\ C & t & Ga \\ BB3 & BB1 & BB1 & BB1 \end{pmatrix} \tag{B.1.4}$$

$$\mathcal{P}_{5-Dim}^{Mixed} = \begin{pmatrix} 1.05 & & & \\ 0.15 & 0.25 & & \\ 0.34 & [0.35,15] & -0.25 & \\ [1,1.50] & [2,1.50] & [2,1.50] & [2,1.50] \end{pmatrix} \tag{B.1.5}$$

$$\mathcal{M}_{5-Dim}^{Mixed} = \begin{pmatrix} 5 & & & & \\ 3 & 4 & & & \\ 4 & 3 & 3 & & \\ 2 & 1 & 2 & 2 & \\ 1 & 2 & 1 & 1 & 1 \end{pmatrix} \tag{B.1.6}$$

In the literature, the first order lag of GARCH family appears to be the most popular specification in financial time series. This paper then follows that in simulation experiments. Hence, all true parameters of EGARCH(1,1,1)-ICAPM-Mixture model, $\{\lambda_1, \lambda_2, \xi^L, \xi^R, \omega, \gamma, \delta, \nu\}$, are as $\{0.00, 0.02, 0.15, 0.35, -0.1, -0.12, 0.13, 0.98\}$.

Variational Bayes algorithms are computationally intensive because of the training procedure. For every training stage, one of the most computationally time consuming processes is the simulation of θ_s and $\mathbf{u}_s$ where $s = 1, ..., S$, in **Line 3** and **Line 8** in Algorithm 3. Hence, the study explores this matter of different S in variational Bayes methods. Given the above true parameter of the marginal conditional volatility model, in Algorithm 3, the experiment was set $S = \{100, 200, 300, 500, 700, 1000, 2000, 3000\}$ for VBDA (DA=0,1,2). The efficient S in variational approximation was selected; for the results see next subsection and in Table B.2.

B.1.2 Simulation Result

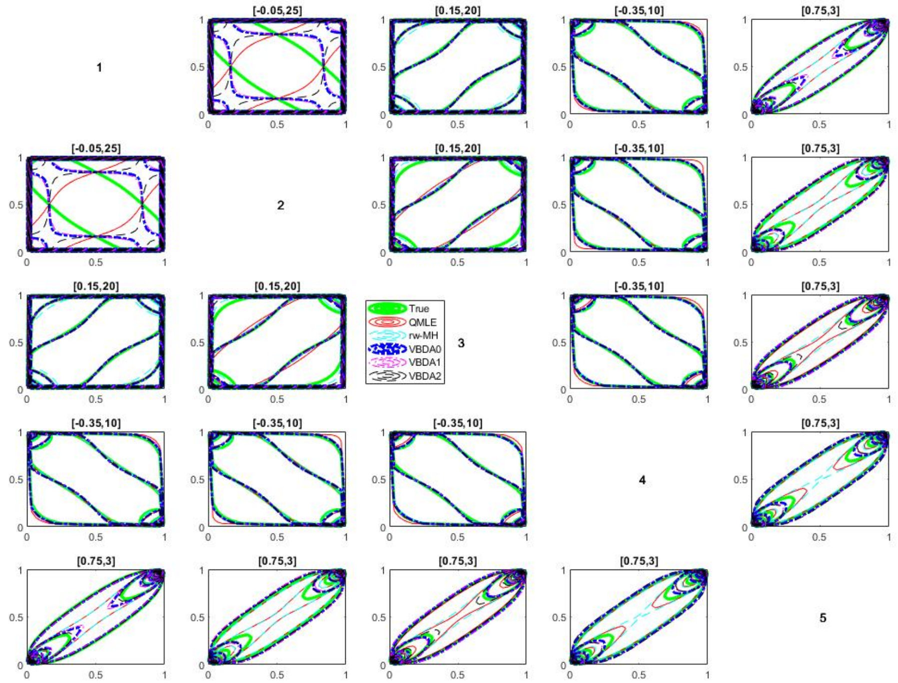
Given simulated data with above true parameters of the R-vine model, the parameter estimator of the three methods and five algorithms is determined by an estimation of performance using mean absolute deviation (MAD), root mean square error (RMSE), AIC, BIC and computational time. All necessary information on Bayesian inference and machine learning are also reported such as the acceptance rate in rw-MH MCMC methods and last VB step in the variational Bayes method in Table B.1. For the experiment in efficient S in variational approximation in variational Bayes methods of Algorithm 3, the study determined average MAD of all marginal model parameter estimators among 30 simulated data. Again, all control variables and necessary conditions are reported.

Table B.1: Simulation Results in IFM Method with Different Scenarios

Dimension	Scenario	Est. Method	EGARCH-M-Mix Margins Model							R-Vine Dependence Model						
			MAD	RMSE	AIC	BIC	Time	Accpt. Rate	Last nVB Step	MAD	RMSE	AIC	BIC	Time	Accpt. Rate	Last nVB Step
5	All Gaussian	*QMLE*	0.039	0.055	6630	6671	0.05	NA	NA	0.056	0.068	-1915	-1912	0.004	NA	NA
		rw-MH	0.030	0.040	6597	6638	4	42%	NA	0.055	0.068	-2003	-2000	3	40%	NA
		VBDA0	0.036	0.054	6265	6306	27	NA	851	0.021	0.025	-1831	-1828	7	NA	363
		VBDA1	0.027	0.046	6287	6328	19	NA	552	0.019	0.023	-1835	-1832	3	NA	312
		VBDA2	0.036	0.056	6250	6291	36	NA	844	0.021	0.025	-1832	-1829	14	NA	321
	All Student-t	*QMLE*	0.042	0.062	6555	6596	0.05	NA	NA	0.028	0.045	-4740	-4720	0.04	NA	NA
		rw-MH	0.018	0.022	6572	6613	4	44%	NA	0.025	0.045	-5320	-5300	18	45%	NA
		VBDA0	0.035	0.052	6380	6422	40	NA	903	0.011	0.018	-4746	-4727	223	NA	3019
		VBDA1	0.026	0.046	6282	6323	18	NA	543	0.012	0.020	-4645	-4625	36	NA	760
		VBDA2	0.034	0.052	6218	6259	34	NA	835	0.011	0.019	-4684	-4665	136	NA	1316
	All Gumbel	*QMLE*	0.042	0.059	6512	6553	0.05	NA	NA	0.044	0.055	-4886	-4883	0.005	NA	NA
		rw-MH	0.031	0.040	6510	6551	5	43%	NA	0.042	0.052	-5225	-5222	5	47%	NA
		VBDA0	0.037	0.057	6219	6260	28	NA	826	0.067	0.093	-2299	-2296	66	NA	1418
		VBDA1	0.030	0.051	6221	6262	20	NA	541	0.058	0.078	-2903	-2900	16	NA	1218
		VBDA2	0.036	0.057	6205	6246	32	NA	800	0.023	0.027	-4899	-4896	107	NA	1876
	All Frank	*QMLE*	0.042	0.059	6616	6657	0.05	NA	NA	0.049	0.059	-1251	-1248	0.004	NA	NA
		rw-MH	0.033	0.043	6605	6646	4	46%	NA	0.045	0.056	-1285	-1282	1	42%	NA
		VBDA0	0.035	0.055	6525	6566	36	NA	793	0.016	0.020	-1211	-1208	2	NA	392
		VBDA1	0.029	0.050	6269	6310	20	NA	549	0.018	0.021	-1211	-1208	2	NA	365
		VBDA2	0.034	0.054	6233	6274	32	NA	788	0.016	0.020	-1211	-1208	21	NA	561
	Mixed	*QMLE*	0.102	0.128	6929	6539	0.05	NA	NA	0.021	0.031	-6708	-6698	0.01	NA	NA
		rw-MH	0.038	0.052	6587	6257	19	59%	NA	0.021	0.030	-8472	-8462	7	45%	NA
		VBDA0	0.038	0.056	6344	6234	31	NA	876	0.046	0.064	-6400	-6390	32	NA	1089
		VBDA1	0.028	0.048	6360	6241	19	NA	563	0.046	0.064	-6474	-6463	23	NA	935
		VBDA2	0.038	0.055	6349	6234	34	NA	838	0.046	0.063	-6495	-6484	51	NA	1073
15	All Gaussian	*QMLE*	0.041	0.058	6572	6613	0.14	NA	NA	0.051	0.063	-33217	-32938	0.04	NA	NA
		rw-MH	0.031	0.037	6564	6605	19	51%	NA	0.028	0.036	-36410	-36131	32	47%	NA
		VBDA0	0.036	0.053	6331	6372	111	NA	904	0.022	0.027	-34999	-34720	146	NA	368
		VBDA1	0.027	0.047	6257	6298	72	NA	555	0.021	0.027	-35417	-35139	133	NA	336
		VBDA2	0.036	0.056	6232	6273	140	NA	839	0.022	0.027	-35088	-34810	279	NA	367
	Mixed	*QMLE*	0.039	0.053	6568	6609	0.17	NA	NA	0.054	0.104	-23102	-22714	0.12	NA	NA
		rw-MH	0.038	0.048	6572	6613	55	55%	NA	0.050	0.098	-25712	-25324	67	48%	NA
		VBDA0	0.036	0.055	6260	6301	100	NA	829	0.041	0.089	35759	36148	302	NA	1080
		VBDA1	0.028	0.050	6285	6326	72	NA	549	0.042	0.091	43311	43699	514	NA	837
		VBDA2	0.035	0.055	6259	6300	128	NA	812	0.042	0.091	46706	47094	710	NA	1119

The estimation methods are QMLE, rw-MH and variational Bayes (VBDA0, VBDA1, VBDA2). NA = data not available. The colour scale (from green to red) indicates the most preferable to the least preferable estimation method. All information is from on an average of up to 79 simulations. MAD and RMSE are represented in the model estimator's bias, while AIC and BIC are represented in sample fitting to the data. Column Time = the computational time in minute. Est. Method = estimation method. Accpt. Rate = acceptance rate in percentage. The last nVB step presents the last iteration in the VBDA estimation method.

Figure B.1: Simulation Results: All Estimated Parameters of the Student-t R-vine Copula Probability Density Plot in Five-Dimensions Against True Parameter



Note that the estimation methods are QMLE, rw-MH, VBDA0, VBDA1 and VBDA2. The true parameters of all Student-t R-vine copula probability density plots are in green. The estimated parameters of QMLE of all Student-t R-vine copula probability density plots are in red. The estimated parameters of rw-MH of all Student-t R-vine copula probability density plot are in cyan. The estimated parameters of VBDA0 of all Student-t R-vine copula probability density plot are in blue. The estimated parameters of VBDA1 of all Student-t R-vine copula probability density plot are in magenta. The estimated parameters of VBDA2 of all Student-t R-vine copula probability density plot are in black. The value above each plot is the true parameter $[\rho, \nu]$ value of bivariate Student-t copula in R-vine.

Table B.1 presents intensive simulation results of this paper. The IFM method was used. The summaries are as follows.

Overall, variational Bayes method performs the best method in R-vine model by MAD and RMSE value. While, given AIC and BIC value in R-vine model, rw-MH estimation method performs the best fitted model to the simulated data. In particular scenario, variational Bayes method outperforms again for all R-vine copula model for both five-dimensions and 15-dimensions according to MAD and RMSE value except in mixed R-vine copula model in five dimensions. While, given by AIC and BIC value, the model prefers either rw-MH and QMLE methods in general, however, variational Bayes performance is still in an acceptable situation with some attention in mixed R-vine model, which the study will see the performance again in empirical high dimensional study. For further details, see Figure B.1. See, for example, all parameter estimates of the Student-t R-vine copula probability density plot in five-dimensions R-vine model against the true parameter in Figure B.1.

Finally, this section presents experimental results of variational approximation in variational Bayes method.

Variational approximation: Efficient variable S

The experiment in efficient variable S in the variational approximation of Algorithm 3 in our marginal model with different S, with a simulated sample size of 500 and repeated 30 times, the control variables were specified as follows; $nVB = 20000,\ c = 500,\ t_w = 20$ and $P = 50$. The results are shown in Table B.2.

Table B.2: Variational Bayes Estimators Comparison Among Different S

S	**100**	**200**	**300**	**500**	**700**	**1000**	**2000**	**3000**
DA=2								
Computational time (mins.)	3	7	12	17	24	32	47	67
Total *nVB* steps	1001	993	1032	1034	1093	1070	1116	1111
MAD	0.0831	**0.0796**	**0.0805**	0.0812	0.0819	0.0814	0.0826	0.0827
DA=1								
Computational time (mins.)	1	3	5	7	10	15	23	34
Total *nVB* steps	629	709	685	756	754	816	812	824
MAD	**0.0591**	**0.0604**	**0.0619**	0.0663	0.0674	0.0697	0.0715	0.0743
DA=0								
Computational time (mins.)	4	8	12	18	25	33	47	65
Total *nVB* steps	979	1032	1012	1064	1047	1126	1127	1151
MAD	0.0824	**0.0807**	**0.0806**	0.0816	0.0819	0.0823	0.0820	0.0827

Table B.2 shows the results of different S experiments in VBDA algorithms. In general, the summaries are as follows. The *VB* method when DA=1 returns the fastest convergence and lowest MAD. The computational time is an exponential function of S. Given our stopping conditions, when S is increasing, the last training step nVB is increasing. This means a longer time in lower bound convergence. The second best method is the most complex VBDA method when DA=2, in terms of MAD and total *nVB* steps. Finally, the study selected S up to 300 because of the trade-off between computational time and accuracy. Furthermore, MAD is unlikely to improve when S is greater than 300.

B.1.3 Data Generation

This appendix presents how the data are generated for simulation experiments in Section B.1.2. In a bivariate copula, let $C(v_1, v_2; \theta, \delta)$ be a distribution function. v_1 and v_2 are independent uniformly distributed, $Unif(0,1)$. Given $u_1 = v_1$, the inverse of the conditional *cdf* copula or the inverse of the h-function is

$$C^{-1}_{2|1}(v_2|v_1; \theta, \delta) = h^{-1}(v_2, v_1; \theta, \delta) = u_2. \tag{B.1.7}$$

then $\mathbf{u} = (u_1, u_2)'$ is a sample from a particular copula with uniform margins. This idea can be performed recursively to the general regular vine sample in n-dimensions similar to the generation of v^{direct} and $v^{indirect}$ in equation 4.4.2. The full algorithm of data generation in Algorithm 4 is the expansion of data simulation, where the R-vine matrix is ordered from n to 1, and $z^{(1)}$ and $z^{(2)}$ selections are the same as in Algorithm 1.

Algorithm 4 presents how to generate the data for the simulation study. The inputs for the algorithm are as follows: n is number of dimensions in R-vines; T is observation number; q is a quantile level; θ is true parameter in marginal function; and matrices ($\mathcal{P},\ \mathcal{M},\ \mathcal{T}$) are true parameters in the R-vine function. In *Line 1-6*, the algorithm performs initial setup. After that follows, the main R-vine algorithm is presented in the first

Algorithm 4 Data generation with full specification of *R-vine* with univariate marginal model incorporated with mixture innovation.

Require: $n,\ T,\ q.$
Require: Marginal model specification: $\theta_i = \{\lambda_{1,i},\ \lambda_{2,i},\ \xi_i^L,\ \xi_i^R,\ \omega_i,\ \gamma_i,\ \delta_i,\ \nu_i\}$ where $i = 1, ..., n$.
Require: *R-vines* specification: matrix $\mathcal{P},\ \mathcal{M},\ \mathcal{T}.$

Generate $v_{j,i} \sim Unif(0,1)$, where $j = 1, ..., n$ and $i = 1, ..., T$.
Calculate $\mathbb{M} = \mathbf{m}_{i,k} | i, k = 1, ..., n$ with $\mathbf{m}_{i,k} = max\{\mathrm{m}_{i,k}, ..., m_{n,k}\}$ for all $k = n, ..., 1$ and $i = n, ..., k$.
Set $z_{n,j}^{(1)} \leftarrow v_{m(j,j),i}$, $j = 1, ..., n,\ i = 1, ..., T$.
Set $z_{n,j}^{(2)} \leftarrow v_{m(n,j),i}$, $j = 1, ..., n,\ i = 1, ..., T$.
Set $v_{n,j}^{direct} \leftarrow z_{j,i}$, $j = 1, ..., n,\ i = 1, ..., T$.
Set $u_1 \leftarrow v_{n,n}^{direct}$.
for $k \leftarrow n-1, 1$ **do**
  for $i \leftarrow k+1, n$ **do**
    if $\mathbf{m}_{i,k} = m_{i,k}$ **then**
      $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{direct}$
    else
      $z_{i,k}^{(2)} = v_{i,(n-\mathbf{m}_{i,k}+1)}^{indirect}$
    end if
    $v_{n,k}^{direct} = h^{-1}(v_{n,k}^{direct}, z_{i,k}^{(2)}; t_{i,k}, \mathfrak{p}_{i,k})$
  end for
  Set $u_{n-k+1} = v_{n,k}^{direct}$.
  for $i \leftarrow n, k+1$ **do**
    Set $z_{i,k}^{(1)} = v_{i,k}^{direct}$.
    Calculate $v_{i-1,k}^{direct} = h(z_{i,k}^{(1)}, z_{i,k}^{(2)} | t_{i,k}, \hat{\mathfrak{p}}_{i,k})$
    Calculate $v_{i-1,k}^{indirect} = h(z_{i,k}^{(2)}, z_{i,k}^{(1)} | t_{i,k}, \hat{\mathfrak{p}}_{i,k})$
  end for
end for
for $j \leftarrow 1, n$ **do**
  Set $\theta^* \leftarrow \theta_j$
  Perform $x_t(u_t) = \begin{cases} F_{GPD^L}^{-1}(u_t | 0, \beta^{L,*}, \xi^{L,*}) & , u_t \le q \\ \Phi^{-1}(u_t) & , q < u_t < 1-q \\ F_{GPD^R}^{-1}(u_t | 0, \beta^{R,*}, \xi^{R,*}) & , u_t \ge 1-q. \end{cases}$
  Calculate $\mathbf{h}_{t,j} | \omega^*, \delta^*, \gamma^*$ from equation 4.3.8.
  Calculate $y_{t,j} | \lambda_1^*, \lambda_2^*$ from equation 4.3.7.
end for
return y

Nested For loops where the outcome is a multivariate R-vine copula sample in n dimensions given by the R-vines specification, **u** = $(u_1, ..., u_n)'$. Given mixture innovation standardisation from equation 4.3.9, the study defines $\beta = (1 - \xi) / (1 - 2\xi)^{1/2}$. The last **For** loop then can obtain data sample y as a final outcome of the algorithm given by the true margin parameter, θ, and multivariate R-vine copula sample, **u**.

B.2 Additional Empirical Results

This is the full 100-dimensional empirical result of Kendall's $\hat{\tau}$ and Spearman $\hat{\rho}_s$ in VBDA1 in the ICAPM-EGARCH(1,1,1)-Mixture-R-vine model. The result is represented in matrix form in which the diagonal matrix presents the stock name, the upper triangular matrix presents Spearman $\hat{\rho}_s$ and the lower triangular matrix presents Kendall's $\hat{\tau}$ as follows.

Row\Column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	'AAPL'	0.02	0.04	-0.02	0.04	0.01	0.01	0.02	0.05	-0.01	-0.05	-0.02	0.00	0.05	-0.01	-0.04
2	0.01	'MSFT'	0.04	-0.06	-0.03	0.02	0.00	0.00	0.04	-0.01	-0.04	-0.03	-0.05	-0.05	-0.06	-0.01
3	0.03	0.03	'GOOG'	0.02	-0.03	0.53	-0.05	0.03	0.02	0.01	-0.03	0.01	-0.07	0.01	0.01	-0.01
4	-0.01	-0.04	0.01	'GOOGL'	0.14	-0.02	0.02	0.03	0.02	0.00	-0.06	0.02	-0.01	-0.05	-0.01	0.02
5	0.03	-0.02	-0.02	0.09	'AMZN'	-0.01	-0.03	0.03	0.05	-0.05	0.01	0.05	-0.05	0.01	-0.03	0.04
6	0.00	0.02	0.37	-0.01	-0.01	'MCHP'	0.03	0.03	0.01	0.00	-0.05	-0.01	0.00	0.16	0.00	0.04
7	0.01	0.00	-0.04	0.01	-0.02	0.02	'TSM'	-0.02	0.04	0.02	0.00	0.00	0.07	-0.04	-0.02	0.02
8	0.01	0.00	0.02	0.02	0.02	0.02	-0.01	'NVDA'	0.01	0.01	0.01	-0.03	0.00	0.05	0.01	-0.02
9	0.03	0.03	0.02	0.02	0.03	0.00	0.03	0.00	'V'	0.02	0.01	-0.03	-0.03	0.03	0.01	0.07
10	-0.01	-0.01	0.01	0.00	-0.04	0.00	0.02	0.01	0.01	'JPM'	0.00	-0.02	0.00	-0.05	-0.08	0.03
11	-0.03	-0.02	-0.02	-0.04	0.01	-0.03	0.00	0.01	0.01	0.00	'IDXX'	0.06	-0.02	0.01	0.01	0.05
12	-0.02	-0.02	0.01	0.01	0.03	-0.01	0.00	-0.02	-0.02	-0.01	0.04	'JNJ'	0.04	0.00	0.07	-0.01
13	0.00	-0.03	-0.05	-0.01	-0.03	0.00	0.05	0.00	-0.02	0.00	-0.01	0.03	'WMT'	0.01	0.00	0.10
14	0.03	-0.03	0.01	-0.03	0.01	0.11	-0.03	0.04	0.02	-0.03	0.01	0.00	0.01	'UNH'	0.03	0.00
15	0.00	-0.04	0.01	-0.01	-0.02	0.00	-0.01	0.01	0.01	-0.05	0.01	0.05	0.00	0.02	'ADI'	0.03
16	-0.03	-0.01	-0.01	0.01	0.03	0.02	0.01	-0.01	0.05	0.02	0.03	-0.01	0.07	0.00	0.02	'HD'
17	0.03	0.06	0.02	0.01	0.02	0.02	0.03	0.00	0.01	-0.01	0.02	0.04	-0.02	0.01	0.06	0.02
18	0.00	0.00	0.02	0.04	0.02	-0.03	0.00	0.02	0.01	0.00	0.01	0.06	0.02	-0.02	-0.02	0.01
19	0.02	0.00	-0.01	-0.01	0.07	-0.03	0.01	-0.01	-0.03	-0.03	-0.01	0.03	-0.01	0.08	-0.02	0.01
20	0.05	0.06	0.06	0.02	0.03	0.00	-0.01	0.01	0.00	0.00	-0.02	0.05	0.00	0.02	0.01	0.00
21	-0.01	0.01	0.05	0.03	0.11	0.00	-0.01	0.02	0.04	0.03	0.00	0.03	-0.01	0.01	0.02	0.04
22	-0.02	0.06	0.02	0.01	0.01	-0.01	-0.02	0.01	0.03	-0.02	0.02	-0.02	0.01	0.02	-0.01	-0.03
23	0.01	0.02	0.04	-0.01	0.03	0.02	-0.02	0.00	0.04	-0.04	0.01	0.04	0.02	-0.03	0.02	0.03
24	-0.03	0.02	0.03	0.00	0.04	0.00	0.01	0.00	0.01	-0.04	0.05	0.03	-0.01	0.03	-0.02	-0.02
25	0.01	0.01	-0.01	-0.01	0.01	0.00	0.02	0.04	0.01	-0.01	0.03	0.05	-0.03	-0.03	-0.04	0.04
26	-0.01	-0.03	-0.01	0.03	0.07	0.01	0.01	-0.02	0.01	0.01	0.01	0.08	0.01	0.01	0.01	-0.02
27	-0.03	0.00	0.02	-0.02	0.00	0.04	-0.02	0.00	0.01	-0.03	-0.06	0.01	-0.01	-0.01	-0.01	0.00
28	0.01	-0.02	-0.03	0.00	0.09	-0.01	-0.02	-0.04	0.00	0.01	0.03	0.00	0.02	0.01	0.01	0.00
29	-0.02	0.02	0.02	-0.02	0.03	0.02	-0.01	0.00	0.02	0.02	0.01	-0.01	-0.01	0.05	-0.04	0.00
30	-0.03	0.00	0.00	0.01	-0.03	0.03	0.03	-0.01	0.03	0.01	-0.02	0.00	0.02	-0.01	-0.03	0.01
31	-0.01	0.05	-0.04	0.00	0.02	-0.02	-0.01	-0.02	-0.03	-0.02	0.00	0.00	0.01	-0.01	-0.04	0.02
32	-0.02	0.04	-0.02	0.05	-0.02	-0.03	-0.02	-0.02	0.04	0.06	0.00	0.00	0.01	0.02	0.04	0.03
33	0.02	0.01	0.03	0.02	0.00	-0.01	-0.02	0.00	0.01	-0.02	0.01	-0.01	-0.01	0.03	0.04	-0.02
34	0.02	-0.03	-0.04	0.01	0.04	0.03	-0.02	-0.01	0.04	-0.02	-0.02	0.01	0.02	-0.03	0.02	0.01
35	0.01	0.05	-0.06	0.03	0.00	0.04	-0.04	0.02	0.02	0.03	0.04	0.03	0.03	0.03	0.01	0.01
36	-0.04	0.03	0.00	-0.03	-0.03	0.01	0.03	-0.03	0.00	-0.02	0.01	0.04	0.01	0.04	0.01	0.03
37	-0.03	-0.05	0.00	0.02	0.03	0.01	-0.02	0.04	0.02	-0.01	0.01	-0.01	0.05	0.01	-0.01	0.02
38	0.04	-0.02	0.00	-0.01	0.06	0.04	-0.02	0.00	-0.03	0.07	0.00	0.06	0.06	0.00	0.01	0.02
39	-0.02	0.03	-0.02	-0.01	-0.03	-0.03	0.05	0.04	0.01	0.00	-0.01	0.05	0.00	-0.01	0.00	-0.02
40	-0.03	0.03	0.08	0.02	0.01	-0.01	-0.01	0.04	0.03	-0.02	-0.01	0.07	0.01	0.01	-0.01	0.03
41	-0.02	-0.01	-0.02	0.04	0.02	-0.02	-0.01	-0.01	-0.03	-0.02	-0.02	0.09	-0.02	0.01	0.00	-0.01
42	0.00	0.09	0.01	0.00	-0.01	0.00	0.00	-0.01	-0.01	0.00	0.05	0.02	0.00	0.02	0.00	0.01
43	0.00	-0.01	0.06	0.00	0.01	-0.02	-0.02	0.01	0.02	0.02	-0.01	0.04	0.01	0.07	0.19	0.00
44	0.03	0.00	0.02	0.05	0.06	0.18	-0.01	0.00	0.01	0.04	0.02	0.00	0.03	-0.03	0.02	0.00
45	0.00	0.01	-0.03	0.01	-0.01	0.02	-0.01	0.02	-0.01	-0.03	0.01	0.02	0.05	0.02	0.04	0.00
46	-0.04	0.02	0.04	0.00	-0.03	0.00	-0.01	-0.01	0.03	-0.01	0.04	0.01	0.02	0.02	0.01	0.05
47	0.05	0.06	-0.01	0.02	0.02	-0.01	0.03	0.02	-0.03	-0.03	0.00	0.00	0.00	0.00	0.04	0.03
48	0.03	0.02	-0.02	0.03	0.04	0.05	0.03	-0.03	-0.01	0.02	-0.01	0.01	0.02	-0.01	0.00	0.02
49	0.02	0.05	-0.03	-0.02	-0.02	0.04	-0.01	0.05	0.01	0.16	-0.02	0.03	0.03	0.02	0.02	-0.01
50	0.01	-0.01	0.03	0.04	0.01	-0.04	0.00	0.04	0.02	0.01	0.00	0.02	-0.01	-0.01	0.01	-0.01
51	-0.01	-0.01	-0.03	0.01	0.00	-0.01	-0.03	-0.01	0.02	0.01	0.01	0.01	-0.01	-0.01	-0.01	0.01
52	0.06	0.00	0.00	0.02	-0.01	0.01	-0.02	0.00	0.01	-0.04	0.03	0.01	-0.02	0.04	0.03	0.02
53	-0.03	0.05	0.06	0.05	0.06	0.02	0.04	0.00	0.06	-0.03	0.03	0.02	0.00	0.04	0.01	0.03
54	-0.02	0.02	0.07	-0.01	-0.01	0.01	0.00	-0.03	-0.03	-0.03	0.02	0.02	0.01	0.04	-0.01	0.00
55	-0.01	-0.01	0.04	0.02	0.04	0.01	0.01	0.01	0.00	-0.01	0.00	-0.01	0.02	0.05	-0.02	-0.03
56	-0.02	-0.01	0.05	0.02	0.03	-0.01	-0.01	0.00	0.01	-0.02	-0.01	0.00	-0.02	0.00	0.00	0.02
57	-0.01	0.04	0.04	-0.01	0.04	0.04	0.01	-0.01	0.01	0.01	0.03	0.00	0.01	0.00	0.00	0.00
58	0.05	0.03	0.02	0.01	-0.01	0.01	0.00	-0.02	-0.01	-0.04	-0.04	0.00	0.02	0.02	-0.01	0.00
59	0.03	0.00	0.00	0.00	0.04	0.01	0.03	-0.04	0.02	0.13	-0.01	0.00	-0.01	0.02	0.03	-0.01
60	0.00	0.01	0.09	-0.01	-0.01	0.05	0.02	0.00	0.03	0.21	-0.04	0.05	0.02	0.03	-0.01	0.01
61	0.05	-0.02	-0.02	0.02	-0.03	0.02	0.00	-0.06	0.02	-0.01	0.01	0.05	0.00	0.08	-0.02	0.03
62	0.02	0.02	-0.01	0.00	-0.03	0.01	-0.01	0.03	0.06	0.01	-0.02	0.14	0.03	0.01	0.01	0.01
63	0.03	0.00	0.13	0.07	0.02	-0.01	0.10	0.00	0.02	0.00	-0.01	0.02	-0.02	0.04	-0.02	0.03
64	0.04	-0.02	0.04	0.05	0.02	0.01	0.01	0.08	-0.02	-0.01	-0.01	0.01	0.02	0.01	0.02	0.03
65	0.02	0.05	0.02	0.03	0.00	0.05	0.05	0.02	0.02	-0.01	0.08	0.10	0.01	0.07	0.02	0.05
66	-0.01	0.02	-0.01	0.04	0.01	0.00	0.00	-0.01	-0.02	-0.02	0.07	0.03	0.02	0.02	-0.01	-0.02
67	0.00	0.02	-0.03	0.01	-0.01	0.00	0.02	0.03	0.03	-0.04	0.04	0.02	0.06	0.09	0.15	0.01
68	-0.02	0.02	0.03	0.06	0.05	0.02	0.03	0.03	0.00	0.17	-0.02	0.01	-0.04	-0.01	0.00	0.01
69	0.06	0.00	0.03	-0.01	0.05	0.11	0.00	0.01	0.03	-0.04	-0.02	0.06	0.02	0.05	0.00	-0.01
70	0.07	0.02	0.06	0.01	0.03	-0.01	0.00	0.01	0.03	0.02	0.00	0.05	-0.02	0.01	0.08	0.03
71	0.01	0.00	0.09	0.04	0.01	-0.01	0.02	0.02	0.02	0.01	-0.02	0.00	0.01	-0.02	0.04	-0.03
72	0.05	0.01	0.03	0.00	0.05	0.01	-0.02	-0.03	-0.02	0.02	0.00	0.03	0.04	0.03	0.00	-0.06
73	-0.01	0.03	0.04	0.06	0.01	-0.01	0.03	0.02	-0.01	0.01	0.00	0.02	0.05	0.07	0.01	-0.02
74	0.03	0.02	0.04	0.02	0.02	-0.01	0.01	0.05	0.01	0.04	-0.02	-0.03	0.02	0.15	0.04	0.01
75	0.04	0.04	0.03	0.01	-0.02	0.00	0.03	-0.02	0.02	-0.01	0.04	0.05	0.01	0.01	-0.01	0.03
76	0.03	0.13	0.06	0.05	0.01	0.02	0.04	0.18	0.02	0.09	0.02	0.03	0.04	0.00	0.08	-0.02
77	0.01	0.04	0.02	0.03	0.02	0.05	0.02	-0.01	0.01	0.06	0.03	-0.01	0.03	0.02	0.03	0.03
78	0.01	0.03	-0.02	0.05	0.04	0.01	0.09	0.01	-0.02	-0.01	0.02	0.02	0.09	0.06	0.13	-0.04
79	0.01	0.05	-0.01	0.00	0.06	0.13	0.02	0.01	0.02	0.01	-0.04	0.03	0.04	0.08	-0.02	0.04
80	0.03	0.01	0.03	0.06	0.11	0.06	-0.01	0.00	0.00	0.01	0.06	0.06	0.01	-0.01	0.05	0.02
81	0.00	-0.02	0.01	0.02	0.07	-0.03	0.06	-0.01	0.02	0.11	0.04	0.01	0.03	0.04	0.24	-0.02
82	0.00	0.02	0.03	0.01	0.13	0.03	0.03	-0.04	0.01	0.05	-0.01	0.01	0.06	0.06	0.02	0.01
83	0.05	0.02	0.03	0.06	0.04	0.02	0.01	0.02	0.04	0.08	0.04	0.02	-0.02	0.09	0.09	0.01
84	-0.02	0.05	0.00	0.05	0.13	-0.02	0.03	0.00	-0.01	0.02	0.02	0.03	-0.02	0.02	-0.04	0.35
85	0.03	0.02	-0.02	0.01	0.04	0.04	0.04	0.02	0.04	0.02	0.04	0.03	0.00	0.11	0.01	0.03
86	0.03	0.05	0.05	0.01	0.07	0.08	0.03	0.00	0.04	0.04	-0.07	0.03	0.05	0.04	0.06	0.00
87	0.06	0.05	0.03	0.11	0.06	0.03	0.03	0.24	0.06	0.17	0.07	0.03	0.03	0.05	0.06	0.12
88	0.01	0.00	0.01	0.10	0.07	0.02	0.05	0.04	0.04	0.03	0.03	0.07	0.02	0.00	0.02	0.07
89	-0.01	0.05	-0.02	0.01	0.02	0.03	0.03	0.01	0.05	0.01	0.01	0.05	0.02	0.09	0.06	0.01
90	0.04	0.04	0.00	0.00	0.01	0.07	0.00	0.11	0.06	0.01	0.10	0.02	0.04	-0.02	-0.01	0.05
91	0.13	0.10	0.01	0.07	0.05	0.06	0.06	0.01	0.04	0.06	0.05	0.00	0.07	0.05	0.03	-0.02
92	0.04	0.16	0.04	0.03	0.06	0.03	0.04	0.00	0.04	0.31	0.05	0.06	0.04	0.07	0.11	0.04
93	0.04	0.02	0.08	0.04	0.01	0.06	0.04	0.06	0.01	-0.01	0.08	0.02	0.04	0.07	0.08	0.02
94	0.06	0.05	0.11	0.08	0.06	0.06	0.10	0.02	0.01	0.01	0.05	0.04	0.11	0.05	0.01	0.09
95	0.11	0.09	0.18	0.08	0.05	0.01	0.10	0.07	0.06	0.02	0.12	0.09	0.26	0.01	0.00	0.07
96	0.08	0.10	0.02	0.06	0.03	0.03	0.01	0.07	0.05	0.21	0.06	0.01	0.08	0.10	0.15	0.14
97	0.12	0.14	0.10	0.12	0.16	0.02	0.11	0.14	0.01	0.08	0.01	0.20	0.17	0.10	0.19	0.14
98	0.03	0.22	0.08	0.18	0.02	0.48	0.18	0.13	0.10	0.18	0.18	0.14	0.06	0.02	0.13	0.03
99	0.08	0.37	0.05	0.20	0.20	0.15	0.36	0.39	0.62	0.17	0.19	0.24	0.12	0.20	0.04	0.27
100	0.43	0.32	0.34	0.47	0.35	0.38	0.24	0.22	0.29	0.43	0.28	0.43	0.28	0.26	0.42	0.35

Row/Column	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	0.05	0.01	0.02	0.07	-0.02	-0.03	0.02	-0.04	0.02	-0.02	-0.04	0.01	-0.03	-0.04	-0.01	-0.03	0.03
2	0.09	0.00	-0.01	0.08	0.02	0.08	0.03	0.03	0.01	-0.04	-0.01	-0.03	0.04	0.00	0.07	0.06	0.01
3	0.03	0.04	-0.02	0.08	0.08	0.02	0.06	0.05	-0.02	-0.02	0.02	-0.04	0.03	0.01	-0.06	-0.02	0.05
4	0.02	0.06	-0.02	0.03	0.04	0.02	-0.01	0.01	-0.02	0.04	-0.03	0.00	-0.02	0.02	0.01	0.08	0.02
5	0.03	0.02	0.10	0.05	0.17	0.02	0.04	0.06	0.02	0.11	0.00	0.13	0.04	-0.04	0.03	-0.03	0.00
6	0.03	-0.04	-0.05	0.00	0.01	-0.02	0.03	0.01	-0.01	0.02	0.06	-0.02	0.03	0.04	-0.03	-0.04	-0.01
7	0.05	-0.01	0.01	-0.01	-0.02	-0.03	-0.03	0.02	0.03	0.01	-0.03	-0.02	-0.01	0.05	-0.02	-0.03	-0.03
8	-0.01	0.04	-0.02	0.01	0.03	0.02	0.01	-0.01	0.06	-0.02	0.00	-0.06	0.00	-0.01	-0.02	-0.03	0.00
9	0.01	0.02	-0.05	0.01	0.06	0.05	0.07	0.02	0.02	0.03	0.02	0.00	0.02	0.04	-0.05	0.06	0.01
10	-0.01	0.00	-0.05	-0.01	0.04	-0.03	-0.06	-0.05	-0.02	0.02	-0.04	0.01	0.03	0.01	-0.04	0.08	-0.03
11	0.03	0.02	-0.02	-0.03	0.00	0.03	0.02	0.07	0.05	0.02	-0.10	0.05	0.02	-0.03	0.00	-0.01	0.02
12	0.06	0.08	0.05	0.08	0.05	-0.02	0.06	0.05	0.08	0.12	0.02	0.00	-0.02	0.00	0.00	0.01	-0.01
13	-0.03	0.03	-0.02	0.01	-0.01	0.02	0.03	-0.01	-0.05	0.01	-0.01	0.04	-0.01	0.03	0.01	0.01	-0.01
14	0.02	-0.02	0.13	0.03	0.01	0.03	-0.04	0.04	-0.05	0.01	-0.01	0.01	0.08	-0.01	-0.01	0.03	0.05
15	0.09	-0.03	-0.04	0.02	0.03	-0.02	0.03	-0.03	-0.06	0.02	-0.01	0.01	-0.06	-0.05	-0.06	0.06	0.05
16	0.03	0.01	0.02	0.00	0.06	-0.04	0.04	-0.04	0.06	-0.03	0.00	0.01	0.00	0.02	0.02	0.04	-0.03
17	'ASML'	0.04	-0.02	0.01	0.13	-0.01	0.04	-0.01	-0.04	0.00	0.03	0.07	0.04	-0.03	-0.01	-0.02	0.04
18	0.02	'PG'	0.01	-0.03	-0.02	0.01	0.02	-0.02	0.02	0.00	-0.04	0.01	0.00	0.04	0.00	-0.03	-0.03
19	-0.02	0.00	'BAC'	0.07	0.01	0.03	0.01	0.00	0.05	0.03	-0.08	0.06	-0.01	0.11	0.02	0.00	0.03
20	0.01	-0.02	0.05	'MA'	-0.02	-0.02	0.03	-0.03	0.05	0.04	0.03	0.09	0.04	-0.01	-0.04	-0.02	0.00
21	0.09	-0.01	0.01	-0.01	'KLAC'	-0.03	0.02	0.02	0.07	-0.01	-0.03	0.02	0.06	0.01	-0.01	0.00	-0.01
22	0.00	0.01	0.02	-0.01	-0.02	'DIS'	0.02	0.00	0.13	0.07	0.03	-0.02	-0.01	-0.04	0.02	0.03	0.05
23	0.03	0.01	0.00	0.02	0.01	0.01	'ADBE'	-0.01	0.02	0.01	0.04	-0.01	0.06	0.00	-0.03	0.02	0.02
24	0.00	-0.01	0.00	-0.02	0.02	0.00	-0.01	'CMCSA'	0.00	0.01	0.03	0.03	-0.02	0.01	0.02	-0.01	-0.01
25	-0.03	0.01	0.03	0.03	0.04	0.09	0.01	0.00	'NFLX'	0.00	0.02	0.01	0.01	0.01	-0.04	-0.01	0.07
26	0.00	0.00	0.02	0.03	-0.01	0.04	0.01	0.00	0.00	'CRM'	-0.04	-0.02	0.01	0.01	0.00	-0.01	-0.03
27	0.02	-0.03	-0.06	0.02	-0.02	0.02	0.02	0.02	0.02	-0.03	'PFE'	0.03	-0.01	0.02	0.13	0.01	0.06
28	0.05	0.01	0.04	0.06	0.01	-0.01	-0.01	0.02	0.00	-0.01	0.02	'TM'	0.01	0.05	-0.05	-0.03	-0.01
29	0.03	0.00	-0.01	0.03	0.04	-0.01	0.04	-0.01	0.01	0.01	-0.01	0.00	'NKE'	0.05	-0.03	0.03	0.02
30	-0.02	0.02	0.07	-0.01	0.01	-0.03	0.00	0.01	0.01	0.01	0.01	0.04	0.03	'CSCO'	-0.02	0.05	0.02
31	0.00	0.00	0.01	-0.03	-0.01	0.02	-0.02	0.01	-0.03	0.00	0.09	-0.04	-0.02	-0.01	'ORCL'	-0.01	-0.04
32	-0.01	-0.02	0.00	-0.01	0.00	0.02	0.01	-0.01	-0.01	-0.01	0.01	-0.02	0.02	0.03	0.00	'KO'	0.05
33	0.03	-0.02	0.02	0.00	-0.01	0.03	0.01	-0.01	0.05	-0.02	0.04	-0.01	0.01	0.01	-0.03	0.03	'TMO'
34	0.03	0.04	0.03	0.04	-0.01	0.01	-0.04	0.02	-0.03	0.03	0.01	-0.05	0.02	-0.01	-0.04	-0.03	0.06
35	0.05	-0.02	0.05	-0.03	-0.01	0.04	0.02	0.00	0.01	-0.01	0.01	-0.01	0.01	-0.01	-0.02	-0.01	0.02
36	0.00	0.02	-0.03	-0.02	-0.04	0.02	0.02	0.02	-0.03	-0.01	-0.02	0.04	-0.01	0.01	0.01	-0.06	-0.02
37	0.03	0.01	0.00	-0.01	-0.01	0.02	-0.02	-0.04	0.05	0.04	-0.01	0.02	-0.01	-0.01	0.02	0.15	0.02
38	0.00	0.02	-0.02	0.04	-0.03	-0.02	0.01	0.05	0.01	0.01	0.02	0.02	0.03	0.01	0.02	0.03	0.04
39	0.03	0.01	0.02	0.06	0.00	0.05	0.05	0.02	0.04	0.00	0.05	0.00	-0.01	0.01	0.04	0.00	0.01
40	0.03	0.01	0.02	-0.01	-0.03	-0.03	-0.02	-0.01	0.02	0.02	0.00	0.00	0.02	-0.01	0.05	0.06	0.00
41	0.00	0.05	-0.01	-0.04	0.01	-0.02	-0.02	0.03	-0.03	-0.04	0.00	0.00	0.05	-0.01	0.02	0.02	0.00
42	0.00	0.05	0.01	-0.03	-0.03	0.02	-0.01	0.03	-0.01	0.00	0.00	0.05	0.01	0.03	0.00	0.02	-0.01
43	0.02	0.00	0.05	0.02	0.03	-0.01	-0.01	0.02	0.00	-0.01	0.03	0.00	-0.01	0.01	-0.02	0.03	-0.06
44	0.01	0.01	0.00	-0.03	-0.02	-0.01	-0.04	0.01	0.01	-0.04	0.01	-0.02	0.00	0.00	0.02	0.05	0.07
45	0.07	0.02	-0.01	-0.03	0.01	0.01	-0.01	0.00	-0.03	-0.02	-0.04	0.03	-0.02	-0.01	-0.02	0.00	-0.03
46	0.00	0.00	-0.03	0.05	0.02	-0.01	-0.02	0.00	0.04	0.03	0.01	0.00	0.03	-0.02	0.00	0.02	-0.01
47	-0.02	0.01	0.00	0.03	0.04	0.04	0.03	0.01	0.01	0.03	0.02	0.05	-0.01	0.02	0.02	0.00	0.09
48	0.04	0.00	0.02	-0.01	0.01	0.02	-0.04	-0.02	0.01	-0.04	0.02	-0.01	0.01	0.03	0.01	0.05	-0.02
49	0.05	0.02	-0.02	0.02	0.01	-0.01	0.00	0.01	-0.02	0.03	0.00	-0.02	0.01	0.00	0.02	-0.03	-0.02
50	-0.01	-0.01	0.03	0.05	0.03	0.04	0.02	0.06	0.03	0.00	0.03	0.00	-0.01	0.02	0.01	0.04	0.00
51	0.00	0.01	0.22	-0.01	-0.02	0.04	0.02	0.03	0.03	0.02	-0.02	0.01	0.00	0.01	-0.01	-0.03	0.09
52	-0.08	0.04	0.02	0.00	0.03	0.01	0.04	0.04	-0.01	0.11	-0.02	-0.02	0.03	0.00	0.02	0.02	-0.01
53	-0.01	0.00	-0.04	0.00	0.01	0.03	-0.01	0.05	0.03	-0.03	0.00	0.00	0.02	0.00	-0.01	-0.01	-0.04
54	0.00	0.00	-0.03	-0.02	0.03	0.03	0.00	0.03	0.02	0.00	0.00	0.01	0.01	0.02	0.00	0.00	0.02
55	0.09	0.06	-0.01	0.02	0.01	0.03	-0.02	-0.01	0.02	0.02	-0.01	-0.02	0.01	0.02	0.02	0.06	0.01
56	-0.02	0.01	-0.03	0.02	0.01	0.02	0.04	0.04	-0.01	0.04	0.01	0.09	0.01	-0.02	0.07	-0.03	0.00
57	0.01	0.02	0.01	0.01	-0.03	0.08	0.00	0.07	0.02	-0.02	0.00	0.03	0.00	0.00	0.03	0.02	0.01
58	0.07	0.03	0.07	0.02	0.02	0.03	0.05	0.01	-0.01	0.03	0.04	-0.02	0.02	0.01	0.03	0.01	0.00
59	0.01	0.00	-0.04	-0.03	0.03	0.04	0.02	0.01	0.03	0.08	0.01	0.00	0.01	0.05	0.00	0.04	0.01
60	0.01	-0.01	0.05	0.01	-0.04	0.02	0.00	0.01	0.00	0.06	0.01	0.02	0.00	0.02	-0.01	-0.01	0.04
61	0.03	0.01	0.01	0.02	0.00	0.00	-0.01	0.04	0.01	0.02	-0.02	0.01	0.00	0.03	0.04	0.10	0.00
62	0.02	-0.01	0.00	0.00	0.07	-0.02	-0.05	0.10	0.02	0.01	0.00	0.03	0.04	0.01	0.00	0.06	0.02
63	-0.03	0.02	-0.05	0.05	-0.03	-0.02	-0.01	0.02	-0.03	0.01	-0.01	-0.01	0.03	0.02	0.02	0.04	0.03
64	0.00	0.02	0.03	0.02	-0.01	0.04	-0.01	-0.01	-0.01	0.05	-0.03	0.04	0.00	0.04	0.00	0.07	-0.01
65	0.05	-0.01	0.05	0.04	0.05	-0.01	0.00	0.05	-0.01	0.02	0.06	-0.04	0.05	0.07	0.04	0.07	0.00
66	0.00	0.01	0.01	0.02	0.04	0.00	-0.01	0.06	-0.02	0.01	0.01	-0.01	0.03	0.01	-0.01	0.05	0.02
67	0.01	0.03	0.22	0.00	0.04	-0.02	0.02	0.03	0.00	0.01	0.02	-0.01	0.00	0.04	-0.05	-0.01	-0.04
68	-0.05	0.02	-0.01	0.00	0.04	0.01	0.09	0.01	0.00	0.02	-0.01	0.04	0.01	0.03	0.01	-0.01	0.04
69	-0.02	-0.02	0.00	0.05	0.00	0.01	0.05	0.05	0.01	0.01	0.03	-0.02	0.03	0.03	0.03	0.04	0.04
70	0.00	0.05	0.03	0.06	0.02	0.04	0.03	0.00	0.06	0.08	-0.01	0.03	0.03	0.02	0.02	0.07	-0.01
71	-0.06	-0.02	0.03	0.09	-0.02	0.01	0.07	0.03	-0.01	0.09	0.02	0.00	0.03	0.01	0.04	-0.01	0.02
72	0.01	-0.01	-0.01	0.05	-0.02	0.01	0.00	0.08	0.07	0.01	0.03	0.04	0.04	0.02	0.02	-0.02	-0.01
73	-0.02	0.02	-0.01	0.03	-0.01	0.02	-0.01	0.08	0.05	0.01	0.01	0.03	0.00	0.01	0.01	0.00	0.10
74	0.01	0.00	0.04	0.01	-0.03	0.01	-0.04	0.06	0.03	0.06	0.00	0.01	0.03	0.01	0.05	-0.02	0.17
75	0.02	0.04	0.01	0.05	0.04	0.04	0.01	-0.01	0.03	-0.02	-0.02	0.03	0.05	0.03	0.05	0.01	0.02
76	0.00	0.00	-0.01	0.01	0.03	0.03	0.09	0.02	0.02	0.04	0.03	0.00	0.00	0.04	0.03	-0.02	0.02
77	0.02	0.03	-0.04	-0.01	0.03	0.01	0.01	0.02	0.06	0.00	0.02	0.02	-0.03	0.00	0.01	0.20	0.03
78	-0.02	0.00	-0.03	0.07	0.02	-0.02	0.08	0.08	0.02	0.09	0.03	0.04	0.04	0.01	0.11	-0.01	0.04
79	0.07	-0.02	-0.04	0.05	0.02	0.04	0.03	0.05	0.03	0.07	0.11	-0.01	0.02	0.04	0.03	0.02	0.01
80	0.08	-0.01	-0.01	0.06	-0.01	0.05	0.02	0.03	0.02	0.02	0.06	0.08	0.04	0.01	0.06	0.05	-0.02
81	0.00	-0.01	0.21	0.01	0.01	-0.03	-0.05	0.03	0.01	0.00	0.04	0.04	0.02	0.02	0.14	-0.02	0.14
82	0.05	0.01	0.03	0.08	0.01	0.04	-0.01	0.02	0.03	0.06	0.04	0.00	-0.01	-0.02	-0.01	0.06	0.03
83	0.04	0.08	-0.02	0.08	-0.02	0.01	0.10	0.07	0.06	-0.02	0.00	0.06	0.00	0.04	0.08	0.01	0.02
84	0.02	0.03	0.03	0.02	0.04	0.04	-0.02	0.06	0.08	0.04	0.00	0.05	0.04	0.06	0.04	0.01	0.03
85	0.00	0.02	-0.02	0.00	0.04	0.01	0.03	0.06	0.05	0.12	0.04	0.02	0.01	0.06	0.01	0.02	0.06
86	-0.01	0.05	0.00	0.01	-0.01	0.07	0.02	0.07	0.05	0.05	0.02	0.03	0.05	0.07	0.02	-0.03	0.02
87	-0.03	0.02	0.02	0.05	0.04	0.01	0.05	0.06	0.05	0.01	0.05	0.00	0.00	0.06	0.06	0.01	0.01
88	0.04	0.05	0.02	0.07	0.15	0.00	0.01	0.07	0.00	0.02	0.01	0.09	0.02	0.10	0.05	0.00	0.07
89	-0.01	0.02	0.10	0.03	0.12	0.05	0.09	0.04	0.02	0.00	0.06	0.05	0.05	0.05	0.02	0.09	-0.01
90	-0.04	0.07	-0.02	0.03	0.04	0.03	0.05	0.07	0.04	0.06	0.04	0.10	0.01	0.08	0.05	0.02	0.08
91	0.04	0.10	0.06	0.07	0.03	0.08	0.06	0.05	0.06	0.01	0.04	0.05	0.04	0.07	0.05	0.01	0.07
92	0.00	0.01	0.20	0.07	0.11	0.07	0.01	0.02	0.01	0.14	0.01	0.06	-0.01	0.02	0.03	0.04	0.08
93	-0.03	0.06	0.03	0.07	0.38	-0.02	0.06	0.09	0.01	0.01	0.02	0.03	0.07	0.07	0.09	0.04	0.10
94	0.02	0.04	0.14	0.10	0.06	0.05	0.01	0.07	0.12	0.02	0.14	0.12	0.10	0.12	0.08	0.03	0.04
95	-0.01	0.05	0.07	0.10	0.04	0.03	0.06	0.07	0.04	0.25	0.06	0.03	0.02	0.06	0.10	0.02	0.12
96	0.06	0.06	0.11	0.06	0.06	0.12	0.05	0.11	0.05	0.04	0.05	0.07	0.02	0.04	0.16	0.03	0.01
97	0.05	0.00	0.11	0.19	0.11	0.13	0.10	0.05	0.06	0.07	0.07	0.12	0.08	0.15	0.16	0.12	0.07
98	0.23	0.16	0.30	0.20	0.16	0.16	0.21	0.29	0.03	0.12	0.10	0.18	0.21	0.10	0.17	0.12	0.12
99	0.11	0.22	0.14	0.31	0.06	0.22	0.29	0.19	0.26	0.03	0.18	0.32	0.27	0.33	0.28	0.25	0.20
100	0.56	0.33	0.08	0.30	0.32	0.53	0.43	0.31	0.19	0.27	0.49	0.28	0.31	0.40	0.32	0.36	0.42

Row\Column	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
1	0.03	0.01	-0.07	-0.05	0.05	-0.03	-0.04	-0.03	0.00	0.00	0.04	0.00	-0.06	0.07	0.04	0.03
2	-0.04	0.07	0.05	-0.07	-0.03	0.05	0.04	-0.01	0.13	-0.01	0.00	0.02	0.03	0.09	0.03	0.08
3	-0.06	-0.09	0.00	0.00	0.00	-0.03	0.11	-0.03	0.01	0.09	0.03	-0.05	0.05	-0.01	-0.03	-0.04
4	0.02	0.04	-0.05	0.02	-0.02	-0.02	0.02	0.07	0.00	0.01	0.08	0.02	0.00	0.04	0.04	-0.03
5	0.06	-0.01	-0.05	0.04	0.08	-0.05	0.01	0.02	-0.02	0.01	0.09	-0.02	-0.04	0.04	0.07	-0.03
6	0.04	0.06	0.01	0.01	0.06	-0.04	-0.01	-0.03	0.00	-0.04	0.26	0.02	0.00	-0.01	0.07	0.06
7	-0.03	-0.06	0.05	-0.03	-0.03	0.08	-0.02	-0.01	0.00	-0.03	-0.02	-0.01	-0.01	0.05	0.05	-0.02
8	-0.01	0.04	-0.04	0.06	0.00	0.05	0.06	-0.01	-0.01	0.01	0.00	0.03	-0.02	0.03	-0.04	0.07
9	0.06	0.03	0.01	0.03	-0.05	0.01	0.05	-0.04	-0.01	0.02	0.01	-0.01	0.04	-0.05	-0.01	0.02
10	-0.03	0.04	-0.02	-0.01	0.10	0.00	-0.04	-0.03	-0.01	0.03	0.05	-0.04	-0.02	-0.05	0.03	0.24
11	-0.03	0.06	0.01	0.01	0.00	-0.01	-0.02	-0.03	0.07	-0.01	0.03	0.02	0.06	0.00	-0.01	-0.03
12	0.01	0.05	0.06	-0.01	0.08	0.08	0.11	0.13	0.03	0.06	0.00	0.03	0.01	0.00	0.01	0.04
13	0.03	0.05	0.01	0.08	0.09	0.00	0.02	-0.03	0.00	0.01	0.05	0.07	0.03	0.01	0.03	0.04
14	-0.04	0.05	0.07	0.01	0.01	-0.02	0.01	0.02	0.03	0.10	-0.04	0.02	0.03	-0.01	-0.01	0.03
15	0.03	0.01	0.01	-0.02	0.01	-0.01	-0.01	0.00	-0.01	0.29	0.04	0.06	0.01	0.06	-0.01	0.02
16	0.01	0.02	0.05	0.02	0.02	-0.04	0.04	-0.02	0.01	-0.01	0.01	0.01	0.07	0.04	0.03	-0.01
17	0.04	0.07	0.00	0.04	0.00	0.04	0.04	0.00	0.00	0.03	0.02	0.10	0.00	-0.04	0.07	0.08
18	0.06	-0.03	0.03	0.02	0.03	0.02	0.01	0.08	0.07	0.00	0.02	0.04	-0.01	0.02	-0.01	0.03
19	0.04	0.07	-0.04	0.00	-0.03	0.03	0.02	-0.01	0.02	0.07	0.00	-0.02	-0.04	0.00	0.02	-0.03
20	0.06	-0.04	-0.02	-0.01	0.06	0.09	-0.02	-0.06	-0.05	0.03	-0.04	-0.04	0.07	0.05	-0.02	0.03
21	-0.02	-0.02	-0.06	-0.02	-0.04	0.00	-0.04	0.01	-0.05	0.04	-0.02	0.01	0.03	0.06	0.02	0.01
22	0.02	0.05	0.04	0.02	-0.02	0.08	-0.04	-0.02	0.02	-0.01	-0.01	0.02	-0.02	0.06	0.03	-0.02
23	-0.06	0.03	0.02	-0.03	0.01	0.08	-0.03	-0.03	-0.02	-0.02	-0.06	-0.02	-0.03	0.04	-0.06	0.00
24	0.03	-0.01	0.03	-0.06	0.07	0.03	-0.01	0.04	0.04	0.02	0.01	0.00	0.00	0.02	-0.02	0.01
25	-0.05	0.01	-0.05	0.07	0.01	0.06	0.04	-0.05	-0.01	0.00	0.02	-0.04	0.06	0.01	0.01	-0.03
26	0.04	-0.01	-0.02	0.05	0.02	0.00	0.03	-0.06	0.01	-0.01	-0.05	-0.03	0.05	0.05	-0.07	0.05
27	0.02	0.01	-0.03	-0.01	0.02	0.07	0.01	0.01	0.01	0.04	0.01	-0.07	0.01	0.03	0.03	0.01
28	-0.07	-0.01	0.06	0.02	0.03	0.00	0.00	0.00	0.08	0.00	-0.03	0.05	0.00	0.08	-0.01	-0.03
29	0.02	0.02	-0.01	-0.02	0.04	-0.01	0.04	0.07	0.02	-0.02	0.01	-0.02	0.04	-0.02	0.02	0.01
30	-0.02	-0.02	0.01	-0.02	0.02	0.02	-0.01	-0.02	0.04	0.01	0.00	-0.01	-0.03	0.04	0.04	0.00
31	-0.07	-0.02	0.02	0.02	0.04	0.05	0.07	0.02	0.01	-0.03	0.02	-0.03	0.00	0.03	0.01	0.03
32	-0.04	-0.01	-0.09	0.23	0.04	0.00	0.08	0.03	0.03	0.05	0.07	0.00	0.03	-0.01	0.08	-0.05
33	0.09	0.02	-0.03	0.03	0.06	0.02	0.00	0.01	-0.02	-0.08	0.10	-0.05	-0.02	0.13	-0.03	-0.02
34	'DHR'	-0.01	-0.01	0.02	0.04	-0.05	0.00	0.07	-0.01	0.04	-0.07	0.08	0.04	0.00	0.07	0.00
35	-0.01	'NVO'	0.06	0.01	0.00	0.02	0.02	0.03	0.00	-0.01	-0.03	0.04	-0.03	-0.02	0.01	-0.04
36	-0.01	0.04	'XOM'	-0.03	0.02	0.02	0.04	0.03	-0.01	0.08	0.02	0.00	-0.03	0.03	0.04	0.04
37	0.01	0.01	-0.02	'VZ'	0.02	-0.01	-0.05	0.04	0.05	0.40	0.01	0.00	0.04	0.04	0.00	0.00
38	0.03	0.00	0.02	0.01	'LLY'	0.08	0.05	0.06	-0.02	0.03	0.14	-0.05	-0.02	0.13	-0.06	0.07
39	-0.03	0.01	0.01	-0.01	0.05	'ABT'	0.06	-0.02	0.03	0.03	0.03	0.00	0.03	0.09	-0.01	0.06
40	0.00	0.01	0.03	-0.03	0.03	0.04	'INTC'	0.07	0.07	0.04	0.01	0.01	-0.01	0.02	-0.06	0.10
41	0.04	0.02	0.02	0.03	0.04	-0.01	0.05	'NTES'	0.03	0.00	0.01	0.05	0.03	-0.06	-0.02	-0.01
42	-0.01	0.00	-0.01	0.03	-0.01	0.02	0.05	0.02	'PEP'	0.07	0.02	0.00	0.08	0.05	0.00	0.04
43	0.03	0.00	0.06	0.27	0.02	0.02	0.03	0.00	0.05	'ACN'	-0.04	-0.02	-0.02	0.00	0.00	-0.02
44	-0.05	-0.02	0.01	0.01	0.09	0.02	0.01	0.00	0.01	-0.03	'COST'	0.06	0.04	0.01	-0.03	-0.03
45	0.05	0.03	0.00	0.00	-0.03	0.00	0.01	0.03	0.00	-0.02	0.04	'T'	-0.01	0.01	-0.02	0.02
46	0.03	-0.02	-0.02	0.02	-0.02	0.02	0.00	0.02	0.05	-0.01	0.03	0.00	'WFC'	-0.02	0.04	0.02
47	0.00	-0.02	0.02	0.02	0.09	0.06	0.01	-0.04	0.04	0.00	0.00	0.01	-0.01	'NVS'	0.00	0.09
48	0.05	0.01	0.03	0.00	-0.04	-0.01	-0.04	-0.01	0.00	0.00	-0.02	-0.01	0.02	0.00	'CVX'	-0.02
49	0.00	-0.03	0.03	0.00	0.04	0.04	0.07	-0.01	0.02	-0.02	-0.02	0.01	0.01	0.06	-0.01	'MRK'
50	0.00	-0.01	-0.01	0.07	-0.02	0.04	0.01	0.06	0.00	-0.02	0.02	0.03	0.00	-0.04	0.02	0.06
51	0.00	0.09	0.00	0.00	0.03	0.03	0.05	0.00	-0.01	0.04	0.04	0.02	0.01	0.00	0.02	-0.02
52	0.06	-0.04	0.02	0.01	0.00	0.00	0.01	0.00	0.01	0.02	0.02	-0.01	-0.01	0.11	0.01	0.03
53	0.08	0.05	0.02	0.01	0.00	0.02	0.01	-0.01	-0.01	0.03	0.01	0.01	0.00	0.01	0.03	0.07
54	0.01	0.01	0.01	-0.06	-0.01	0.08	0.05	0.06	0.07	0.00	0.01	0.01	-0.01	0.00	0.00	0.01
55	0.02	0.02	0.03	0.03	-0.02	0.00	0.03	0.04	0.04	0.01	0.01	0.05	0.01	0.05	-0.03	0.01
56	-0.05	0.04	0.03	-0.01	0.00	0.03	-0.02	-0.01	0.00	0.00	-0.01	-0.02	0.00	-0.01	0.02	0.06
57	0.04	-0.03	0.00	0.04	0.00	0.10	0.01	0.04	0.04	0.10	0.01	0.06	0.00	0.00	0.00	0.04
58	0.00	-0.03	0.00	0.00	0.02	0.02	0.01	-0.02	0.03	0.03	0.04	-0.01	0.01	0.03	0.04	0.00
59	0.06	-0.03	-0.03	-0.02	0.03	0.01	0.01	0.02	0.02	0.03	0.00	0.00	0.01	-0.01	0.03	0.00
60	0.03	0.00	-0.02	0.02	0.00	0.00	-0.01	0.03	0.01	0.01	-0.03	0.08	0.00	-0.01	0.00	-0.01
61	0.02	0.00	-0.02	0.00	0.12	0.05	0.01	-0.01	0.06	0.01	0.01	0.07	0.05	0.03	0.04	-0.01
62	0.05	0.01	-0.01	0.03	0.00	0.01	-0.01	0.01	0.10	-0.01	0.00	-0.02	0.00	0.02	0.00	0.00
63	0.03	-0.02	0.02	0.08	0.01	0.03	0.00	0.01	0.03	-0.01	0.04	0.02	0.01	0.03	-0.02	0.03
64	0.06	-0.01	0.04	0.03	-0.01	0.00	0.02	0.00	0.07	-0.02	0.01	0.01	0.00	0.02	0.06	0.02
65	0.02	-0.03	0.07	0.05	-0.01	0.02	0.00	0.02	0.09	0.04	0.01	0.01	-0.02	0.03	0.00	0.01
66	0.04	0.03	0.02	-0.01	-0.03	0.04	0.00	0.02	0.08	-0.02	0.04	-0.02	0.00	0.05	-0.01	0.00
67	-0.01	0.03	-0.03	0.00	0.15	0.08	0.01	0.00	0.06	0.04	0.05	0.00	-0.01	0.10	0.05	0.04
68	0.01	-0.01	-0.01	0.00	0.03	0.00	0.01	-0.01	0.05	-0.01	0.07	0.03	0.00	0.08	0.01	0.01
69	-0.02	0.07	0.00	0.00	0.08	-0.02	0.02	0.03	0.01	0.01	0.01	0.00	0.00	-0.03	0.05	0.09
70	-0.01	0.00	0.02	0.01	-0.04	0.00	0.01	-0.02	0.08	0.06	0.03	0.04	0.01	-0.01	0.01	0.01
71	-0.02	0.02	-0.01	0.04	0.04	0.01	0.03	0.04	0.06	0.01	0.03	-0.01	0.01	-0.01	0.01	-0.02
72	0.06	0.01	-0.02	0.03	-0.04	0.01	0.02	0.02	0.04	0.02	-0.01	0.00	0.53	0.05	-0.04	-0.01
73	0.06	0.01	0.09	0.02	0.05	0.01	0.05	0.01	-0.03	0.00	-0.04	-0.03	0.00	0.03	0.01	0.01
74	-0.04	0.05	0.06	-0.01	0.05	0.03	0.01	0.08	0.05	0.03	0.04	-0.01	0.06	0.25	-0.04	-0.01
75	0.03	0.07	-0.02	0.08	0.01	0.00	0.02	0.04	0.02	-0.02	0.03	0.00	0.03	-0.01	0.10	-0.02
76	0.00	0.04	0.04	0.01	-0.04	-0.01	0.00	0.00	0.02	-0.02	0.01	-0.02	-0.06	0.00	-0.01	0.14
77	0.00	0.05	0.06	0.01	-0.02	0.04	-0.01	0.07	0.08	0.04	0.00	0.03	0.00	0.00	-0.01	0.01
78	0.03	-0.01	0.07	0.07	0.03	0.14	0.02	0.02	0.00	0.06	0.03	-0.01	-0.03	0.05	0.03	0.00
79	0.05	0.07	0.08	-0.02	0.00	0.04	0.01	0.02	0.01	0.04	0.02	0.08	0.04	0.06	0.00	0.06
80	0.04	0.01	0.02	0.00	0.09	0.10	0.00	0.00	-0.01	-0.01	-0.03	0.10	0.00	-0.03	0.03	0.11
81	0.00	0.00	0.01	0.17	-0.01	0.03	0.02	0.03	0.01	0.07	0.06	0.07	-0.02	0.00	0.07	-0.02
82	0.02	0.01	0.01	-0.03	0.01	0.05	-0.04	-0.05	0.04	-0.01	0.06	0.00	-0.01	-0.03	0.01	0.06
83	-0.03	0.04	0.30	0.00	-0.03	0.01	0.02	-0.01	0.02	0.01	-0.03	0.03	0.02	0.09	-0.02	0.20
84	0.02	0.02	0.01	0.03	0.01	0.07	0.02	0.01	0.05	0.04	0.00	0.08	0.02	0.03	-0.01	0.00
85	0.02	0.06	0.07	0.05	0.02	0.01	0.03	0.07	0.02	-0.02	0.04	0.01	0.01	0.02	-0.01	0.07
86	0.02	0.04	0.12	-0.01	0.04	0.05	0.08	-0.01	0.02	0.06	-0.02	0.06	0.07	-0.01	0.04	0.03
87	0.03	0.08	0.09	0.01	0.08	0.03	-0.02	0.01	0.04	0.02	0.09	0.03	-0.01	-0.01	0.04	0.07
88	0.03	-0.01	0.08	0.01	0.02	-0.01	0.01	0.01	-0.06	0.02	-0.01	0.04	0.01	-0.01	0.13	0.05
89	0.05	0.05	0.13	0.08	0.04	-0.02	0.12	-0.02	0.06	0.05	0.01	0.05	0.09	0.08	0.07	0.01
90	0.02	0.03	0.01	0.04	0.05	0.10	0.11	0.04	0.01	0.08	0.01	0.02	-0.03	0.09	0.10	0.08
91	0.10	0.02	0.02	0.06	0.01	0.05	-0.02	0.06	0.03	0.10	0.05	0.11	0.02	0.16	0.06	0.02
92	0.01	0.02	0.10	0.03	0.04	0.07	0.05	0.02	0.04	0.06	0.09	0.17	0.03	0.07	0.13	0.06
93	0.10	0.02	0.02	0.14	0.00	0.14	0.06	0.08	0.09	0.07	0.11	0.08	0.12	0.09	0.07	0.04
94	0.09	0.00	0.02	0.08	0.04	0.01	0.07	0.04	0.12	0.15	0.02	0.06	0.02	0.08	0.03	0.03
95	0.11	0.05	0.13	0.10	0.06	0.01	0.02	0.04	0.07	0.09	0.08	0.09	0.02	0.03	0.02	0.02
96	0.06	0.08	0.02	0.07	0.06	0.14	0.07	0.05	0.07	0.26	0.15	0.11	0.14	0.05	0.12	0.24
97	0.09	0.02	0.10	0.10	0.02	0.06	0.11	0.06	0.02	0.02	0.29	0.17	0.06	0.03	0.13	0.18
98	0.20	0.14	0.06	0.15	0.07	0.06	0.13	0.08	0.02	0.16	0.13	0.09	0.12	0.14	0.12	0.11
99	0.23	0.19	0.23	0.05	0.21	0.31	0.19	0.21	0.20	0.23	0.21	0.20	0.31	0.09	0.51	0.13
100	0.39	0.35	0.40	0.31	0.29	0.31	0.41	0.26	0.27	0.40	0.24	0.29	0.25	0.32	0.30	0.29

Row\Column	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
1	0.01	-0.01	0.10	-0.05	-0.03	-0.02	-0.03	-0.02	0.07	0.05	0.01	0.07	0.03	0.04	0.05	0.03	-0.01
2	-0.01	-0.01	0.00	0.07	0.03	-0.02	-0.01	0.05	0.04	0.00	0.02	-0.03	0.03	0.00	-0.02	0.08	0.03
3	0.05	-0.04	0.00	0.08	0.10	0.05	0.08	0.05	0.03	0.00	0.13	-0.03	-0.02	0.19	0.06	0.03	-0.01
4	0.07	0.01	0.03	0.08	-0.01	0.03	0.03	-0.02	0.01	-0.01	-0.01	0.04	0.01	0.10	0.07	0.05	0.05
5	0.02	0.00	-0.02	0.10	-0.02	0.06	0.05	0.06	-0.02	0.06	-0.02	-0.05	-0.04	0.03	0.03	0.00	0.01
6	-0.06	-0.02	0.02	0.02	0.01	0.01	-0.01	0.06	0.01	0.02	0.08	0.03	0.01	-0.02	0.01	0.08	0.01
7	-0.01	-0.04	-0.03	0.06	0.00	0.01	-0.01	0.02	0.00	0.04	0.03	0.00	-0.02	0.15	0.01	0.07	0.01
8	0.07	-0.01	-0.01	0.00	-0.04	0.01	0.00	-0.02	-0.03	-0.06	0.00	-0.09	0.04	0.00	0.12	0.04	-0.01
9	0.04	0.03	0.02	0.09	-0.05	-0.01	0.01	0.02	-0.02	0.03	0.05	0.03	0.09	0.03	-0.03	0.04	-0.02
10	0.01	0.01	-0.06	-0.04	-0.04	-0.02	-0.03	0.02	-0.06	0.19	0.31	-0.02	0.02	0.00	-0.02	-0.02	-0.03
11	0.01	0.01	0.04	0.04	0.03	0.01	-0.01	0.05	-0.05	-0.01	-0.06	0.01	-0.03	-0.01	-0.02	0.13	0.11
12	0.03	0.02	0.01	0.03	0.03	-0.01	-0.01	0.00	0.00	0.00	0.08	0.07	0.21	0.02	0.02	0.15	0.04
13	-0.01	-0.02	-0.03	-0.01	0.01	0.02	-0.02	0.02	0.03	-0.01	0.02	0.00	0.05	-0.03	0.03	0.02	0.02
14	-0.01	-0.02	0.06	0.06	0.06	0.08	0.00	-0.01	0.02	0.04	0.04	0.12	0.02	0.05	0.01	0.11	0.04
15	0.01	-0.02	0.04	0.02	-0.01	-0.03	0.00	0.00	-0.02	0.05	-0.02	-0.03	0.01	-0.04	0.03	0.04	-0.01
16	-0.02	0.02	0.02	0.05	0.00	-0.05	0.03	0.00	0.00	-0.02	0.02	0.04	0.01	0.04	0.04	0.07	-0.03
17	-0.02	0.00	-0.11	-0.02	0.00	0.14	-0.04	0.02	0.11	0.01	0.02	0.04	0.03	-0.04	0.00	0.08	0.00
18	-0.01	0.02	0.05	0.00	0.01	0.08	0.01	0.03	0.04	0.01	-0.01	0.02	-0.02	0.02	0.03	-0.01	0.01
19	0.04	0.32	0.02	-0.06	-0.04	-0.02	-0.04	0.02	0.10	-0.05	0.08	0.01	0.00	-0.07	0.04	0.07	0.01
20	0.07	-0.02	0.00	0.00	-0.02	0.02	0.03	0.01	0.03	-0.05	0.01	0.02	0.00	0.07	0.03	0.06	0.02
21	0.04	-0.03	0.05	0.01	0.04	0.01	0.02	-0.04	0.03	0.05	-0.06	0.00	0.10	-0.05	-0.01	0.08	0.05
22	0.06	0.06	0.02	0.05	0.05	0.04	0.03	0.12	0.05	0.07	0.03	0.00	-0.03	-0.03	0.06	-0.01	0.00
23	0.03	0.03	0.06	-0.01	0.00	-0.03	0.06	0.00	0.08	0.03	0.00	-0.01	-0.08	-0.02	-0.02	0.00	-0.02
24	0.08	0.04	0.06	0.07	0.05	-0.01	0.05	0.11	0.01	0.02	0.01	0.06	0.15	0.03	-0.02	0.08	0.10
25	0.04	0.04	-0.02	0.04	0.03	0.03	-0.02	0.03	-0.01	0.04	0.00	0.01	0.03	-0.04	-0.02	-0.02	-0.03
26	0.01	0.04	0.16	-0.04	0.00	0.03	0.06	-0.03	0.04	0.11	0.09	0.04	0.01	0.02	0.07	0.03	0.01
27	0.04	-0.03	-0.04	0.00	-0.01	-0.01	0.01	0.00	0.05	0.01	0.02	-0.03	0.00	-0.01	-0.04	0.09	0.01
28	0.01	0.02	-0.02	0.00	0.02	-0.03	0.14	0.04	-0.03	0.01	0.03	0.02	0.05	-0.01	0.07	-0.06	-0.02
29	-0.01	0.00	0.05	0.02	0.02	0.01	0.02	0.00	0.03	0.02	0.01	0.00	0.06	0.04	0.00	0.07	0.04
30	0.03	0.02	0.01	0.00	0.03	0.04	-0.04	0.00	0.01	0.08	0.03	0.05	0.02	0.02	0.06	0.11	0.01
31	0.01	-0.02	0.04	-0.01	0.00	0.03	0.11	0.05	0.05	0.01	-0.01	0.07	0.00	0.03	0.00	0.06	-0.02
32	0.07	-0.04	0.03	-0.02	0.01	0.10	-0.04	0.03	0.01	0.07	-0.02	0.14	0.09	0.07	0.10	0.11	0.08
33	0.00	0.14	-0.01	-0.06	0.02	0.02	0.00	0.02	0.00	0.01	0.07	0.00	0.02	0.04	-0.01	0.00	0.03
34	-0.01	0.00	0.10	0.12	0.02	0.03	-0.08	0.06	0.00	0.09	0.04	0.03	0.08	0.05	0.10	0.03	0.06
35	-0.02	0.13	-0.06	0.08	0.02	0.03	0.06	-0.05	-0.04	-0.04	0.00	0.00	0.02	-0.02	-0.02	-0.05	0.05
36	-0.01	0.00	0.03	0.04	0.02	0.05	0.05	0.00	0.00	-0.05	-0.03	-0.03	-0.01	0.03	0.06	0.11	0.03
37	0.11	0.00	0.01	0.01	-0.08	0.04	-0.01	0.05	0.00	-0.04	0.03	-0.01	0.05	0.11	0.04	0.08	-0.01
38	-0.03	0.04	0.01	0.00	-0.01	-0.03	0.00	0.00	0.02	0.05	0.00	0.18	-0.01	0.01	-0.02	-0.02	-0.04
39	0.06	0.05	0.00	0.03	0.12	0.00	0.05	0.15	0.03	0.01	0.00	0.07	0.02	0.05	0.00	0.03	0.05
40	0.01	0.08	0.01	0.02	0.07	0.04	-0.03	0.01	0.02	0.01	-0.01	0.01	-0.01	-0.01	0.03	0.01	0.01
41	0.08	0.01	0.00	-0.01	0.10	0.06	-0.01	0.05	-0.03	0.03	0.05	-0.01	0.02	0.01	0.00	0.02	0.04
42	0.00	-0.02	0.02	-0.01	0.11	0.06	-0.01	0.05	0.04	0.04	0.01	0.08	0.15	0.05	0.10	0.13	0.13
43	-0.02	0.05	0.03	0.04	0.01	0.02	-0.01	0.14	0.05	0.05	0.01	0.02	-0.01	-0.01	-0.03	0.07	-0.03
44	0.04	0.06	0.03	0.02	0.01	0.02	-0.02	0.02	0.06	0.00	-0.04	0.01	0.00	0.06	0.02	0.02	0.06
45	0.04	0.03	-0.01	0.01	0.01	0.07	-0.03	0.09	-0.01	0.00	0.12	0.10	-0.04	0.03	0.02	0.02	-0.03
46	0.01	0.02	-0.01	0.00	-0.01	0.02	0.00	0.00	0.02	0.01	0.01	0.08	0.01	0.02	0.00	-0.02	0.00
47	-0.06	0.01	0.16	0.02	0.00	0.08	-0.02	0.01	0.05	-0.01	-0.02	0.05	0.03	0.04	0.03	0.04	0.07
48	0.04	0.03	0.01	0.05	0.00	-0.04	0.02	-0.01	0.07	0.05	0.00	0.07	0.00	-0.03	0.10	0.00	-0.01
49	0.09	-0.03	0.05	0.10	0.02	0.01	0.09	0.06	0.00	0.00	-0.02	-0.01	0.00	0.04	0.02	0.01	0.00
50	'AZN'	0.00	0.03	0.02	-0.01	-0.02	-0.01	0.05	0.01	0.00	-0.01	-0.02	0.06	0.05	0.00	0.03	-0.04
51	0.00	'MS'	-0.02	-0.05	0.03	0.06	-0.01	-0.05	0.02	0.02	0.02	0.00	-0.01	0.01	0.03	0.02	-0.02
52	0.02	-0.01	'MCD'	-0.04	0.05	0.07	-0.01	0.01	0.00	-0.01	0.00	0.00	0.02	0.01	0.00	0.02	0.04
53	0.02	-0.04	-0.02	'TXN'	0.00	0.03	-0.05	0.06	0.09	0.07	0.07	0.00	0.04	-0.03	0.10	0.05	0.01
54	0.00	0.02	0.03	0.00	'MDT'	0.03	0.01	0.03	0.04	-0.02	0.07	0.06	0.06	-0.02	0.05	0.01	0.04
55	-0.01	0.04	0.05	0.02	0.02	'UPS'	-0.01	0.07	0.01	0.03	0.05	0.06	-0.03	-0.03	0.05	0.02	0.00
56	0.00	0.00	-0.01	-0.03	0.01	-0.01	'SAP'	-0.01	0.02	0.07	0.13	0.01	0.02	0.10	0.04	0.04	0.01
57	0.03	-0.03	0.01	0.04	0.02	0.05	-0.01	'NEE'	0.12	0.03	-0.01	0.08	-0.06	0.07	0.04	0.00	0.08
58	0.00	0.01	0.00	0.06	0.02	0.01	0.01	0.08	'PM'	0.03	-0.03	0.03	0.00	0.04	-0.02	0.12	0.01
59	0.00	0.01	-0.01	0.04	-0.02	0.02	0.05	0.02	0.02	'TMUS'	-0.02	0.04	0.00	0.02	-0.01	0.07	0.06
60	0.00	0.01	0.00	0.04	0.05	0.04	0.09	-0.01	-0.02	-0.02	'LIN'	0.04	0.11	0.00	0.10	-0.01	0.01
61	-0.01	0.00	0.00	0.00	0.04	0.04	0.01	0.05	0.02	0.03	0.02	'INTU'	0.09	0.01	0.09	0.01	0.05
62	0.04	0.00	0.01	0.03	0.04	-0.02	0.01	-0.04	0.00	0.00	0.07	0.06	'QCOM'	0.03	0.01	0.01	0.03
63	0.03	0.01	0.00	-0.02	-0.01	-0.02	0.07	0.05	0.03	0.02	0.00	0.01	0.02	'HON'	0.08	0.04	-0.05
64	0.00	0.02	0.00	0.07	0.04	0.03	0.03	0.02	-0.01	0.00	0.07	0.06	0.01	0.05	'LOW'	0.01	-0.04
65	0.02	0.02	0.01	0.03	0.00	0.01	0.02	0.00	0.08	0.05	-0.01	0.00	0.01	0.03	0.01	'UL'	0.04
66	-0.02	-0.01	0.03	0.01	0.03	0.00	0.00	0.05	0.01	0.04	0.01	0.03	0.02	-0.03	-0.02	0.03	'RY'
67	-0.02	0.05	-0.02	-0.02	0.01	0.00	0.00	0.00	0.04	0.02	0.03	0.05	0.00	0.00	0.02	0.01	0.01
68	0.06	-0.05	0.02	-0.01	0.00	0.02	-0.02	-0.01	0.04	-0.02	0.04	-0.03	-0.01	0.02	0.04	0.01	0.05
69	-0.01	0.26	0.04	0.02	0.05	0.06	0.00	0.03	0.01	0.00	0.03	0.03	0.05	-0.01	-0.01	0.04	0.00
70	0.15	0.00	-0.02	0.04	0.07	0.02	0.00	0.00	0.03	0.00	0.06	0.02	0.01	0.00	0.01	-0.01	-0.01
71	-0.03	0.00	0.13	0.02	0.02	0.06	0.05	0.00	0.09	0.00	0.05	0.05	-0.01	0.05	0.02	-0.01	-0.01
72	-0.01	0.04	0.00	0.03	0.05	0.01	0.05	0.15	0.06	0.00	0.05	0.05	0.01	0.01	0.02	0.01	0.01
73	0.03	0.00	0.05	0.07	0.00	-0.03	0.04	-0.02	0.05	0.02	0.04	0.01	0.04	0.04	0.00	0.03	0.00
74	0.01	0.02	-0.03	0.06	0.06	0.05	0.05	-0.02	-0.02	0.02	0.03	0.01	0.01	0.00	0.02	0.04	0.04
75	0.01	0.05	-0.04	0.06	0.13	0.05	0.06	0.00	0.04	0.06	0.02	0.02	0.02	0.00	0.03	0.15	0.01
76	0.07	0.03	0.00	0.06	0.02	0.00	0.04	0.08	0.05	0.07	0.06	0.04	0.02	0.00	0.02	0.00	0.03
77	0.01	0.02	0.06	0.08	0.04	0.04	0.01	0.06	0.01	0.07	0.06	0.01	0.02	0.05	0.05	0.04	0.06
78	-0.02	-0.01	0.00	0.02	0.12	0.01	0.03	0.02	-0.01	0.02	0.04	0.11	0.01	0.05	0.01	0.00	0.01
79	-0.02	0.05	-0.01	0.06	-0.02	0.02	0.05	0.03	-0.02	0.01	0.04	0.03	0.04	0.00	0.05	-0.03	0.01
80	0.01	0.01	0.03	0.06	0.01	0.06	0.01	0.02	0.04	0.03	-0.01	0.04	0.00	0.17	-0.02	0.02	0.03
81	-0.02	0.03	-0.02	0.01	0.02	0.03	-0.02	0.05	0.09	0.01	0.01	0.06	-0.01	0.06	0.03	-0.01	0.41
82	0.01	0.03	0.02	0.04	-0.02	0.04	0.08	0.04	-0.04	0.04	0.04	0.06	0.01	0.02	0.04	0.09	-0.01
83	0.02	0.04	0.01	0.04	0.01	0.05	-0.03	0.08	0.02	0.03	0.06	0.04	0.01	0.10	0.02	-0.02	0.02
84	0.01	0.04	0.00	0.09	0.11	0.03	0.02	0.04	0.11	0.07	0.01	0.05	0.01	0.05	0.01	0.01	0.09
85	0.03	-0.01	0.03	0.02	0.02	0.06	0.04	-0.02	0.03	0.04	0.06	0.05	0.01	0.06	-0.01	0.00	0.01
86	-0.01	0.01	0.04	0.06	0.04	0.18	0.03	-0.03	0.03	0.07	0.05	0.02	0.02	0.02	0.08	0.07	0.13
87	0.04	0.37	0.05	0.06	0.06	0.03	0.10	0.07	0.03	0.05	0.02	-0.01	0.02	0.09	0.05	0.02	0.01
88	0.04	0.09	0.00	0.08	0.03	0.00	0.06	0.04	-0.02	0.02	0.05	0.03	0.00	0.03	0.02	0.02	-0.03
89	0.02	0.07	-0.01	0.05	0.01	0.08	0.05	0.02	0.03	0.07	0.03	0.02	0.09	0.10	0.03	0.08	0.06
90	0.01	0.09	0.07	0.06	0.01	0.11	0.07	-0.06	-0.01	0.09	0.08	0.03	-0.03	0.08	0.05	0.03	0.08
91	-0.01	0.05	0.06	0.26	0.09	0.03	0.06	0.08	0.03	0.05	0.06	0.02	0.00	0.16	0.13	0.06	0.15
92	0.03	0.02	0.06	0.07	0.03	0.07	0.00	-0.04	0.01	0.09	0.06	0.02	0.01	0.12	0.06	0.02	0.11
93	-0.01	0.05	0.06	0.08	0.06	0.04	0.06	0.06	0.08	0.09	0.04	0.10	0.05	0.15	0.08	0.05	0.08
94	0.00	0.07	0.05	0.11	0.02	0.06	0.10	0.00	0.07	0.09	0.12	0.16	0.03	0.23	0.13	0.01	0.08
95	0.04	-0.04	0.05	0.12	0.14	0.09	0.25	0.07	0.05	0.09	0.10	0.10	0.12	0.19	0.14	0.02	0.03
96	0.04	0.07	0.06	0.11	0.08	0.17	0.11	0.05	0.06	0.14	0.13	0.07	0.00	0.12	0.06	0.06	0.19
97	0.08	0.21	0.09	0.23	0.08	0.05	0.04	0.09	0.12	0.09	0.18	0.14	0.07	0.17	0.11	0.01	0.19
98	0.19	0.15	0.09	0.06	0.21	0.21	0.07	0.03	0.16	0.11	0.19	0.23	0.06	0.12	0.19	0.06	0.04
99	0.17	0.36	0.17	0.24	0.26	0.24	0.23	0.03	0.21	0.11	0.22	0.29	0.15	0.33	0.32	0.26	0.20
100	0.34	0.36	0.41	0.36	0.35	0.41	0.35	0.28	0.30	0.29	0.45	0.37	0.39	0.40	0.33	0.43	0.32

Row\Column	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
1	0.00	-0.02	0.09	0.10	0.01	0.07	-0.02	0.04	0.06	0.04	0.02	0.02	0.02	0.05	0.00	0.00	0.08
2	0.04	0.03	0.00	0.03	0.00	0.01	0.04	0.03	0.06	0.19	0.05	0.04	0.07	0.02	-0.04	0.03	0.03
3	-0.04	0.04	0.05	0.09	0.14	0.05	0.06	0.06	0.04	0.09	0.03	-0.03	-0.01	0.05	0.02	0.04	0.05
4	0.02	0.09	-0.01	0.02	0.06	0.00	0.10	0.03	0.02	0.07	0.05	0.08	0.00	0.10	0.03	0.02	0.08
5	-0.01	0.07	0.08	0.04	0.01	0.07	0.02	0.04	-0.03	0.01	0.02	0.06	0.10	0.16	0.10	0.19	0.05
6	0.01	0.04	0.16	-0.02	-0.01	0.01	-0.01	-0.02	0.00	0.03	0.08	0.01	0.20	0.09	-0.04	0.04	0.03
7	0.03	0.05	0.00	0.00	0.03	-0.02	0.05	0.01	0.05	0.06	0.02	0.13	0.03	-0.02	0.09	0.04	0.02
8	0.05	0.05	0.01	0.02	0.03	-0.05	0.02	0.07	-0.04	0.27	-0.01	0.02	0.02	0.01	-0.02	-0.06	0.03
9	0.05	-0.01	0.05	0.05	0.03	-0.03	-0.02	0.02	0.02	0.03	0.01	-0.03	0.02	-0.01	0.03	0.01	0.07
10	-0.05	0.25	-0.06	0.03	0.02	0.03	0.01	0.06	-0.01	0.14	0.08	-0.01	0.02	0.02	0.16	0.07	0.12
11	0.06	-0.03	-0.02	0.00	-0.03	-0.01	0.00	-0.03	0.05	0.03	0.04	0.03	-0.06	0.09	0.05	-0.02	0.06
12	0.02	0.02	0.10	0.08	0.00	0.05	0.03	-0.05	0.07	0.04	-0.01	0.03	0.04	0.09	0.01	0.02	0.03
13	0.09	-0.05	0.02	-0.03	0.01	0.05	0.07	0.03	0.02	0.07	0.05	0.13	0.06	0.01	0.05	0.08	-0.03
14	0.14	-0.02	0.07	0.02	-0.03	0.04	0.10	0.23	0.02	0.00	0.02	0.09	0.12	-0.02	0.05	0.09	0.13
15	0.23	0.00	0.00	0.12	0.07	0.00	0.01	0.06	-0.02	0.11	0.05	0.19	-0.03	0.07	0.36	0.03	0.13
16	0.02	0.01	-0.02	0.04	-0.04	-0.09	-0.03	0.01	0.05	-0.03	0.05	-0.06	0.06	0.03	-0.03	0.01	0.02
17	0.02	-0.07	-0.03	0.01	-0.09	0.01	-0.02	0.01	0.02	0.01	0.03	-0.03	0.11	0.12	0.00	0.08	0.06
18	0.04	0.03	-0.03	0.08	-0.03	-0.01	0.02	0.00	0.06	0.00	0.04	0.01	-0.03	-0.02	-0.01	0.01	0.12
19	0.33	-0.01	0.00	0.04	0.05	-0.01	-0.01	0.05	0.01	-0.02	-0.06	-0.05	-0.05	-0.01	0.30	0.04	-0.02
20	0.00	0.00	0.08	0.09	0.13	0.07	0.05	0.02	0.07	0.01	-0.01	0.11	0.07	0.09	0.02	0.11	0.12
21	0.06	0.06	0.01	0.04	-0.02	-0.03	-0.02	-0.04	0.06	0.04	0.04	0.03	0.03	-0.02	0.02	0.02	-0.04
22	-0.03	0.02	0.02	0.06	0.02	0.02	0.03	0.02	0.06	0.05	0.01	-0.02	0.06	0.08	-0.04	0.06	0.01
23	0.03	0.13	0.07	0.04	0.10	0.00	-0.02	-0.06	0.02	0.13	0.02	0.11	0.04	0.04	-0.07	-0.01	0.15
24	0.04	0.02	0.08	0.00	0.04	0.11	0.13	0.08	-0.02	0.03	0.02	0.13	0.07	0.05	0.05	0.03	0.11
25	0.00	0.00	0.02	0.09	-0.02	0.10	0.07	0.04	0.05	0.03	0.09	0.02	0.04	0.03	0.02	0.04	0.08
26	0.01	0.04	0.02	0.12	0.13	0.01	0.02	0.09	-0.02	0.07	0.00	0.14	0.10	0.03	0.00	0.09	-0.04
27	0.03	-0.01	0.04	-0.01	0.04	0.04	0.01	0.00	-0.03	0.05	0.04	0.05	0.17	0.09	0.05	0.05	0.00
28	-0.02	0.05	-0.03	0.04	0.01	0.05	0.05	0.02	0.04	-0.01	0.03	0.06	-0.01	0.11	0.05	-0.01	0.08
29	0.00	0.01	0.05	0.04	0.04	0.06	0.00	0.04	0.07	-0.01	-0.04	0.06	0.03	0.06	0.03	-0.01	0.00
30	0.06	0.05	0.04	0.03	0.01	0.02	0.01	0.02	0.04	0.06	0.00	0.02	0.06	0.01	0.03	-0.03	0.06
31	-0.07	0.01	0.05	0.03	0.06	0.04	0.01	0.07	0.08	0.04	0.02	0.16	0.05	0.09	0.21	-0.02	0.12
32	-0.02	-0.02	0.06	0.11	-0.01	-0.03	0.00	-0.03	0.02	-0.03	0.30	-0.01	0.03	0.08	-0.03	0.09	0.02
33	-0.05	0.06	0.05	-0.01	0.03	-0.01	0.15	0.25	0.03	0.03	0.04	0.06	0.01	-0.03	0.21	0.04	0.02
34	-0.01	0.02	-0.03	-0.01	-0.03	0.08	0.09	-0.06	0.05	0.01	0.00	0.04	0.07	0.05	0.00	0.03	-0.04
35	0.05	-0.01	0.11	-0.01	0.03	0.01	0.01	0.07	0.10	0.05	0.07	-0.02	0.10	0.02	0.00	0.02	0.06
36	-0.04	-0.02	0.01	0.04	-0.02	-0.03	0.13	0.09	-0.02	0.06	0.08	0.11	0.12	0.03	0.02	0.01	0.43
37	-0.01	-0.01	-0.01	0.01	0.06	0.05	0.04	-0.02	0.12	0.01	0.01	0.11	-0.03	0.00	0.26	-0.05	0.00
38	0.23	0.05	0.11	-0.06	0.06	-0.06	0.08	0.08	0.01	-0.06	-0.04	0.04	0.00	0.13	-0.02	0.02	-0.04
39	0.13	-0.01	-0.02	0.01	0.01	0.01	0.02	0.04	-0.01	-0.01	0.06	0.21	0.06	0.15	0.04	0.07	0.02
40	0.01	0.02	0.03	0.01	0.04	0.03	0.07	0.02	0.04	0.01	-0.01	0.03	0.01	0.00	0.03	-0.06	0.03
41	-0.01	-0.01	0.05	-0.03	0.05	0.03	0.01	0.13	0.07	-0.01	0.11	0.02	0.03	0.00	0.05	-0.08	-0.02
42	0.10	0.07	0.02	0.12	0.09	0.06	-0.04	0.08	0.03	0.03	0.11	0.00	0.02	-0.01	0.01	0.06	0.02
43	0.06	-0.02	0.01	0.10	0.01	0.03	0.00	0.04	-0.02	-0.03	0.07	0.09	0.06	-0.02	0.10	-0.01	0.02
44	0.08	0.11	0.01	0.05	0.04	-0.02	-0.06	0.05	0.05	0.01	-0.01	0.04	0.02	-0.04	0.09	0.09	-0.04
45	0.01	0.05	0.00	0.05	-0.01	0.00	-0.05	-0.01	0.01	-0.03	0.05	-0.01	0.13	0.15	0.10	0.00	0.04
46	-0.01	0.00	0.01	0.01	0.01	0.71	0.00	0.08	0.04	-0.08	0.01	-0.04	0.05	0.00	-0.04	-0.02	0.03
47	0.15	0.13	-0.05	-0.02	-0.01	0.08	0.05	0.37	-0.01	0.00	0.00	0.08	0.08	-0.05	-0.01	-0.04	0.14
48	0.08	0.01	0.07	0.02	0.02	-0.06	0.01	-0.06	0.15	-0.01	-0.02	0.05	0.00	0.05	0.10	0.02	-0.03
49	0.05	0.02	0.14	0.01	-0.03	-0.01	0.02	-0.02	-0.04	0.20	0.02	0.00	0.09	0.16	-0.04	0.10	0.29
50	-0.02	0.09	-0.01	0.22	-0.05	-0.02	0.04	0.01	0.02	0.10	0.01	-0.02	-0.02	0.01	-0.03	0.01	0.03
51	0.07	-0.08	0.38	-0.01	0.01	0.05	0.00	0.03	0.08	0.05	0.04	-0.01	0.07	0.01	0.04	0.04	0.07
52	-0.03	0.02	0.07	-0.03	0.19	0.01	0.07	-0.04	-0.07	0.00	0.08	0.00	-0.01	0.05	-0.02	0.02	0.01
53	-0.03	-0.02	0.03	0.06	0.03	0.05	0.10	0.08	0.09	0.09	0.12	0.03	0.09	0.10	0.02	0.06	0.06
54	0.01	0.00	0.07	0.10	0.03	0.08	0.01	0.08	0.19	0.03	0.06	0.18	-0.02	0.02	0.03	-0.03	0.02
55	0.01	0.02	0.09	0.02	0.08	0.01	-0.04	0.08	0.08	0.01	0.07	0.02	0.03	0.09	0.05	0.06	0.07
56	0.00	-0.03	0.00	-0.01	0.08	0.08	0.06	0.07	0.09	0.06	0.02	0.04	0.08	0.02	-0.03	0.12	-0.04
57	0.01	-0.01	0.05	0.00	0.00	0.22	-0.04	-0.03	0.01	0.12	0.09	0.03	0.05	0.04	0.07	0.07	0.13
58	0.07	0.06	0.02	0.05	0.14	0.09	0.08	-0.04	0.06	0.07	0.02	-0.02	-0.03	0.06	0.13	-0.06	0.04
59	0.02	-0.03	-0.01	0.00	0.00	0.00	0.03	0.04	0.09	0.10	0.10	0.02	0.02	0.04	0.02	0.06	0.04
60	0.05	0.07	0.05	0.09	0.07	0.08	0.06	0.05	0.04	0.08	0.10	0.07	0.06	-0.01	0.01	0.07	0.09
61	0.07	-0.05	0.04	0.02	0.07	0.07	0.01	0.01	0.02	0.06	0.01	0.16	0.05	0.07	0.08	0.09	0.06
62	0.00	-0.01	0.07	0.01	-0.01	0.02	0.06	0.02	0.02	0.03	0.04	0.01	0.06	0.00	-0.01	0.01	0.01
63	0.00	0.03	-0.01	-0.01	0.08	0.01	0.05	-0.01	-0.01	0.01	0.07	0.08	0.00	0.25	0.09	0.03	0.15
64	0.02	0.05	-0.01	0.01	0.03	0.03	0.00	0.04	0.04	0.03	0.08	0.01	0.08	-0.03	0.05	0.06	0.03
65	0.02	0.01	0.06	-0.01	-0.01	0.02	0.04	0.06	0.23	0.00	0.06	0.00	-0.04	0.03	-0.01	0.13	-0.03
66	0.02	0.08	0.00	-0.01	-0.01	0.02	0.00	0.06	0.02	0.04	0.10	0.01	0.02	0.04	0.58	-0.02	0.03
67	'ILMN'	-0.04	0.03	0.06	0.07	-0.02	0.06	0.18	0.00	0.09	0.05	-0.02	0.01	0.05	0.09	0.07	0.06
68	-0.03	'BHP'	0.03	0.00	0.02	0.02	0.03	0.04	-0.02	0.04	0.01	-0.01	0.01	0.00	0.06	0.03	-0.04
69	0.02	0.02	'C'	0.05	0.06	0.01	0.06	0.04	-0.02	0.19	-0.10	0.37	0.09	0.05	-0.01	0.15	0.03
70	0.04	0.00	0.03	'SONY'	0.03	0.05	0.02	0.03	0.02	0.05	0.06	0.04	0.13	0.10	0.00	0.02	0.10
71	0.05	0.01	0.04	0.02	'HDB'	-0.03	0.09	0.07	-0.02	0.02	0.04	0.00	0.02	0.07	0.00	0.02	0.12
72	-0.01	0.01	0.00	0.03	-0.02	'BMY'	-0.02	0.03	0.04	0.02	0.07	0.06	0.07	0.03	-0.01	0.07	0.00
73	0.04	0.02	0.04	0.01	0.06	-0.01	'AMT'	0.02	0.00	0.05	0.01	-0.02	0.11	0.01	0.03	0.06	0.05
74	0.12	0.03	0.02	0.02	0.04	0.02	0.01	'SBUX'	-0.01	0.01	0.04	-0.03	0.10	-0.02	0.10	0.06	0.11
75	0.00	-0.01	-0.01	0.01	-0.01	0.03	0.00	-0.01	'BLK'	-0.01	-0.01	0.05	-0.02	0.05	0.01	0.00	0.02
76	0.06	0.02	0.13	0.03	0.01	0.01	0.04	0.00	-0.01	'SCHW'	0.01	0.00	0.01	0.01	0.00	0.10	0.06
77	0.03	0.01	-0.07	0.04	0.03	0.05	0.01	0.02	-0.01	0.01	'UNP'	0.05	0.06	0.01	0.02	-0.03	-0.03
78	-0.01	-0.01	0.25	0.03	0.00	0.04	-0.02	-0.02	0.03	0.00	0.03	'BHP'	0.03	0.04	0.03	0.04	-0.01
79	0.01	0.00	0.06	0.09	0.01	0.05	0.07	0.07	-0.01	0.01	0.04	0.02	'AXP'	0.03	0.08	-0.01	0.02
80	0.04	0.00	0.04	0.07	0.05	0.02	0.00	-0.01	0.04	0.01	0.01	0.02	0.02	'GS'	0.06	0.07	0.07
81	0.06	0.04	-0.01	0.00	0.00	-0.01	0.02	0.07	0.01	0.00	0.01	0.02	0.06	0.04	'RTX'	0.00	0.04
82	0.05	0.02	0.10	0.01	0.01	0.04	0.04	0.04	0.00	0.07	-0.02	0.02	-0.01	0.05	0.00	'AMD'	0.02
83	0.04	-0.03	0.02	0.07	0.08	0.00	0.03	0.07	0.01	0.04	-0.02	-0.01	0.02	0.04	0.03	0.01	'BA'
84	0.00	0.05	0.02	0.02	0.09	0.04	0.06	0.00	0.04	0.08	0.01	0.04	-0.03	0.01	0.02	0.00	0.04
85	0.08	0.07	0.07	0.00	0.04	0.02	0.04	0.01	0.05	0.03	0.00	0.00	0.17	0.04	0.03	0.07	0.04
86	-0.02	-0.02	0.01	0.04	0.02	0.00	0.08	0.00	0.14	-0.01	0.04	0.06	0.06	0.05	0.22	0.02	0.01
87	0.06	-0.01	0.08	0.07	0.01	0.02	0.02	0.05	0.02	0.04	0.02	0.02	0.04	0.02	0.03	0.03	0.03
88	0.05	0.02	0.07	0.08	0.07	0.02	0.01	0.02	0.02	-0.01	0.07	0.07	0.03	0.09	0.08	0.17	0.07
89	0.05	0.02	0.02	0.04	0.03	0.05	0.00	0.04	0.11	0.02	0.02	0.03	0.06	0.03	0.08	0.10	0.05
90	0.02	-0.03	0.04	0.03	0.06	0.05	0.04	0.05	0.06	0.06	0.03	0.56	0.02	0.06	0.04	0.04	0.03
91	0.00	0.03	0.02	0.04	0.00	0.04	0.09	0.05	0.10	0.11	0.17	0.03	0.04	0.11	0.04	0.07	0.03
92	0.04	0.03	0.01	0.08	0.07	0.05	0.02	0.07	0.08	0.04	0.01	0.04	0.14	0.09	0.13	0.09	0.14
93	0.05	-0.02	0.13	0.02	0.05	0.02	0.08	0.03	0.17	0.02	-0.01	0.07	0.02	0.18	0.07	0.12	0.04
94	0.09	-0.01	0.11	0.04	0.01	0.07	0.07	0.13	0.07	0.18	0.04	0.03	0.10	0.05	0.08	0.07	0.16
95	0.08	0.01	0.11	0.06	0.10	0.12	0.10	0.08	0.03	0.02	0.01	0.09	0.10	0.11	0.22	0.08	0.12
96	0.16	0.01	0.07	0.08	0.02	0.06	0.06	0.10	0.14	0.12	0.13	0.13	0.13	0.02	0.08	0.06	0.14
97	0.06	0.04	0.10	0.11	0.12	0.08	0.06	0.17	0.05	0.10	0.04	0.12	0.14	0.17	0.23	0.12	0.03
98	0.11	0.79	0.24	0.02	0.14	0.07	0.17	0.15	0.15	0.19	0.11	0.02	0.24	0.13	0.20	0.07	0.12
99	0.10	0.25	0.18	0.21	0.22	0.03	0.04	0.31	0.20	0.25	0.22	0.29	0.11	0.25	0.28	0.06	0.29
100	0.31	0.50	0.44	0.33	0.35	0.41	0.26	0.32	0.57	0.37	0.45	0.46	0.45	0.07	0.40	0.34	0.33

Row\Column	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	-0.03	0.04	0.04	0.10	0.02	-0.02	0.06	0.19	0.05	0.06	0.09	0.17	0.13	0.18	0.04	0.12	0.59
2	0.08	0.04	0.08	0.08	-0.01	0.08	0.05	0.15	0.24	0.03	0.08	0.13	0.15	0.20	0.33	0.52	0.46
3	-0.01	-0.03	0.08	0.05	0.01	-0.03	0.00	0.01	0.06	0.12	0.17	0.27	0.03	0.15	0.11	0.07	0.47
4	0.08	0.01	0.01	0.16	0.15	0.02	0.00	0.11	0.05	0.07	0.12	0.12	0.09	0.18	0.26	0.29	0.66
5	0.20	0.05	0.10	0.09	0.10	0.02	0.02	0.08	0.09	0.02	0.09	0.07	0.05	0.24	0.03	0.30	0.50
6	-0.03	0.06	0.12	0.05	0.02	0.04	0.11	0.09	0.05	0.09	0.09	0.02	0.04	0.02	0.67	0.22	0.54
7	0.05	0.06	0.04	0.05	0.08	0.05	0.01	0.08	0.06	0.05	0.15	0.15	0.02	0.17	0.26	0.51	0.35
8	0.00	0.03	-0.01	0.35	0.06	0.01	0.17	0.01	0.01	0.09	0.03	0.11	0.10	0.20	0.19	0.56	0.32
9	-0.02	0.06	0.07	0.09	0.06	0.07	0.08	0.06	0.06	0.01	0.01	0.09	0.08	0.02	0.15	0.81	0.42
10	0.02	0.03	0.06	0.25	0.05	0.02	0.02	0.09	0.45	-0.02	0.02	0.03	0.30	0.12	0.27	0.25	0.60
11	0.03	0.06	-0.11	0.11	0.05	0.02	0.15	0.08	0.07	0.12	0.07	0.17	0.09	0.02	0.27	0.28	0.41
12	0.04	0.04	0.05	0.04	0.10	0.07	0.02	0.00	0.08	0.03	0.06	0.13	0.02	0.30	0.22	0.35	0.60
13	-0.02	0.00	0.08	0.04	0.03	0.02	0.06	0.10	0.06	0.06	0.16	0.39	0.11	0.25	0.09	0.17	0.41
14	0.03	0.16	0.06	0.07	0.00	0.13	-0.03	0.08	0.10	0.10	0.08	0.02	0.14	0.15	0.04	0.29	0.39
15	-0.06	0.02	0.09	0.09	0.02	0.09	-0.02	0.05	0.16	0.12	0.01	0.00	0.22	0.28	0.20	0.06	0.58
16	0.50	0.04	0.01	0.18	0.11	0.02	0.07	-0.03	0.05	0.02	0.14	0.11	0.21	0.20	0.04	0.40	0.51
17	0.04	-0.01	-0.01	-0.05	0.06	-0.02	-0.07	0.07	0.00	-0.04	0.02	-0.01	0.09	0.08	0.34	0.16	0.74
18	0.04	0.03	0.08	0.03	0.07	0.04	0.10	0.15	0.01	0.08	0.06	0.07	0.09	0.00	0.24	0.33	0.47
19	0.04	-0.03	0.01	0.02	0.03	0.15	-0.04	0.09	0.29	0.04	0.20	0.10	0.17	0.16	0.44	0.20	0.12
20	0.03	0.00	0.02	0.08	0.10	0.04	0.05	0.11	0.11	0.10	0.15	0.15	0.09	0.28	0.29	0.45	0.43
21	0.06	0.06	-0.02	0.07	0.22	0.18	0.06	0.04	0.16	0.55	0.09	0.06	0.09	0.16	0.24	0.08	0.46
22	0.06	0.02	0.11	0.02	-0.01	0.07	0.05	0.12	0.10	-0.03	0.07	0.05	0.18	0.20	0.24	0.33	0.71
23	-0.04	0.04	0.03	0.07	0.02	0.14	0.07	0.09	0.02	0.09	0.02	0.09	0.07	0.16	0.31	0.43	0.61
24	0.09	0.09	0.11	0.09	0.10	0.07	0.11	0.08	0.02	0.13	0.10	0.10	0.16	0.07	0.42	0.28	0.45
25	0.12	0.07	0.07	0.08	0.00	0.03	0.07	0.09	0.01	0.02	0.18	0.06	0.07	0.09	0.05	0.38	0.29
26	0.06	0.18	0.08	0.01	0.02	-0.01	0.08	0.02	0.20	0.01	0.03	0.36	0.06	0.10	0.17	0.04	0.39
27	0.01	0.06	0.03	0.08	0.02	0.09	0.07	0.05	0.02	0.02	0.20	0.09	0.08	0.11	0.15	0.26	0.66
28	0.08	0.02	0.05	-0.01	0.13	0.07	0.15	0.08	0.08	0.05	0.18	0.04	0.11	0.17	0.27	0.46	0.42
29	0.06	0.02	0.08	0.00	0.03	0.07	0.01	0.07	-0.01	0.10	0.15	0.03	0.03	0.12	0.32	0.40	0.45
30	0.09	0.09	0.11	0.09	0.15	0.07	0.12	0.11	0.03	0.11	0.19	0.09	0.06	0.23	0.15	0.47	0.57
31	0.06	0.01	0.03	0.09	0.07	0.03	0.08	0.08	0.04	0.13	0.12	0.15	0.23	0.24	0.25	0.41	0.46
32	0.02	0.03	-0.05	0.01	0.01	0.13	0.03	0.02	0.06	0.06	0.04	0.04	0.05	0.18	0.18	0.36	0.51
33	0.04	0.09	0.02	0.02	0.11	-0.02	0.12	0.11	0.11	0.16	0.07	0.18	0.02	0.10	0.18	0.29	0.60
34	0.03	0.04	0.04	0.04	0.05	0.08	0.03	0.15	0.01	0.15	0.14	0.17	0.10	0.14	0.29	0.34	0.55
35	0.03	0.08	0.05	0.12	-0.02	0.07	0.04	0.02	0.03	0.03	0.00	0.08	0.11	0.03	0.21	0.29	0.49
36	0.01	0.11	0.17	0.13	0.13	0.19	0.02	0.03	0.15	0.03	0.02	0.20	0.03	0.15	0.09	0.33	0.57
37	0.05	0.08	-0.01	0.02	0.01	0.11	0.06	0.09	0.04	0.20	0.13	0.15	0.11	0.16	0.22	0.07	0.45
38	0.01	0.03	0.06	0.12	0.03	0.07	0.08	0.02	0.06	0.00	0.07	0.08	0.09	0.03	0.11	0.30	0.42
39	0.10	0.02	0.07	0.05	-0.02	-0.02	0.15	0.07	0.10	0.21	0.02	0.02	0.21	0.09	0.09	0.45	0.45
40	0.03	0.04	0.13	-0.03	0.02	0.19	0.17	-0.02	0.08	0.09	0.11	0.03	0.10	0.17	0.20	0.28	0.58
41	0.01	0.11	-0.02	0.01	0.01	-0.03	0.06	0.09	0.02	0.12	0.06	0.07	0.08	0.09	0.11	0.30	0.37
42	0.08	0.03	0.02	0.07	-0.09	0.10	0.02	0.05	0.07	0.14	0.17	0.11	0.11	0.03	0.04	0.30	0.39
43	0.07	-0.02	0.09	0.03	0.03	0.08	0.12	0.15	0.09	0.11	0.22	0.13	0.38	0.03	0.24	0.34	0.56
44	0.01	0.06	-0.03	0.13	-0.02	0.02	0.02	0.07	0.13	0.16	0.02	0.12	0.22	0.43	0.20	0.31	0.35
45	0.12	0.02	0.09	0.05	0.06	0.07	0.02	0.16	0.25	0.13	0.09	0.13	0.17	0.25	0.13	0.29	0.42
46	0.02	0.02	0.10	-0.01	0.01	0.13	-0.05	0.03	0.04	0.18	0.02	0.03	0.20	0.09	0.19	0.45	0.36
47	0.05	0.03	-0.01	-0.01	-0.02	0.12	0.13	0.23	0.10	0.13	0.12	0.04	0.08	0.04	0.21	0.14	0.47
48	-0.01	-0.01	0.06	0.06	0.19	0.11	0.14	0.09	0.19	0.10	0.04	0.03	0.18	0.19	0.19	0.70	0.44
49	0.00	0.11	0.04	0.10	0.07	0.01	0.13	0.02	0.09	0.05	0.05	0.03	0.36	0.27	0.16	0.20	0.42
50	0.01	0.04	-0.01	0.07	0.06	0.04	0.02	-0.01	0.04	-0.01	0.00	0.06	0.06	0.11	0.28	0.25	0.47
51	0.07	-0.02	0.01	0.53	0.13	0.10	0.14	0.08	0.03	0.08	0.10	-0.06	0.11	0.30	0.22	0.52	0.52
52	0.00	0.04	0.06	0.08	0.01	-0.02	0.10	0.09	0.09	0.09	0.08	0.08	0.10	0.14	0.14	0.26	0.57
53	0.13	0.03	0.09	0.08	0.13	0.07	0.09	0.39	0.10	0.12	0.16	0.17	0.16	0.33	0.08	0.36	0.51
54	0.17	0.03	0.06	0.09	0.05	0.02	0.02	0.13	0.05	0.08	0.03	0.21	0.12	0.13	0.32	0.38	0.50
55	0.04	0.09	0.27	0.05	0.00	0.12	0.16	0.04	0.10	0.06	0.09	0.13	0.26	0.07	0.31	0.36	0.56
56	0.03	0.06	0.04	0.14	0.09	0.08	0.11	0.09	0.00	0.09	0.14	0.37	0.16	0.06	0.11	0.34	0.50
57	0.06	-0.03	-0.05	0.10	0.07	0.03	-0.09	0.11	-0.06	0.09	0.01	0.11	0.08	0.14	0.05	0.04	0.40
58	0.16	0.05	0.05	0.05	-0.02	0.05	-0.01	0.05	0.01	0.13	0.10	0.07	0.08	0.17	0.24	0.31	0.43
59	0.10	0.06	0.10	0.07	0.02	0.11	0.13	0.07	0.14	0.13	0.14	0.14	0.20	0.14	0.17	0.17	0.43
60	0.02	0.09	0.07	0.03	0.08	0.05	0.12	0.09	0.09	0.06	0.18	0.14	0.19	0.26	0.27	0.32	0.63
61	0.07	0.07	0.04	-0.01	0.05	0.03	0.04	0.02	0.04	0.14	0.24	0.16	0.11	0.22	0.33	0.42	0.52
62	0.01	0.01	0.03	0.04	0.00	0.14	-0.04	0.01	0.01	0.08	0.04	0.18	0.01	0.11	0.10	0.22	0.54
63	0.08	0.09	0.03	0.14	0.05	0.15	0.12	0.23	0.18	0.22	0.35	0.28	0.17	0.25	0.18	0.48	0.55
64	0.01	-0.01	0.12	0.07	0.03	0.04	0.07	0.19	0.10	0.12	0.20	0.21	0.10	0.17	0.28	0.46	0.47
65	0.01	-0.01	0.10	0.04	0.03	0.12	0.04	0.10	0.03	0.08	0.01	0.02	0.09	0.02	0.09	0.38	0.60
66	0.13	0.02	0.20	0.02	-0.05	0.09	0.12	0.22	0.16	0.12	0.13	0.05	0.28	0.27	0.06	0.30	0.47
67	-0.01	0.12	-0.03	0.09	0.07	0.08	0.04	0.00	0.05	0.08	0.14	0.11	0.23	0.09	0.16	0.15	0.45
68	0.08	0.11	-0.02	-0.01	0.02	0.03	-0.05	0.05	0.05	-0.03	-0.02	0.02	0.01	0.06	0.94	0.37	0.68
69	0.03	0.10	0.01	0.12	0.11	0.03	0.05	0.03	0.02	0.20	0.16	0.16	0.10	0.15	0.35	0.26	0.61
70	0.04	0.00	0.06	0.10	0.12	0.06	0.04	0.06	0.12	0.02	0.05	0.09	0.12	0.16	0.04	0.31	0.48
71	0.13	0.06	0.03	0.02	0.10	0.04	0.08	-0.01	0.11	0.07	0.01	0.15	0.03	0.17	0.21	0.33	0.50
72	0.06	0.03	0.00	0.03	0.03	0.08	0.08	0.06	0.08	0.04	0.10	0.18	0.10	0.12	0.10	0.05	0.57
73	0.09	0.06	0.12	0.03	0.02	0.01	0.07	0.14	0.03	0.13	0.11	0.14	0.09	0.09	0.25	0.06	0.37
74	0.01	0.02	0.00	0.07	0.03	0.05	0.07	0.07	0.10	0.05	0.19	0.13	0.15	0.26	0.22	0.45	0.46
75	0.05	0.07	0.21	0.03	0.03	0.16	0.09	0.15	0.11	0.26	0.11	0.04	0.22	0.08	0.22	0.30	0.76
76	0.13	0.04	-0.02	0.06	-0.02	0.03	0.09	0.17	0.06	0.03	0.27	0.03	0.17	0.16	0.28	0.36	0.51
77	0.02	-0.01	0.05	0.03	0.10	0.03	0.05	0.25	0.01	-0.02	0.06	0.01	0.19	0.06	0.16	0.32	0.62
78	0.05	0.00	0.09	0.03	0.10	0.05	0.75	0.05	0.07	0.10	0.05	0.13	0.19	0.18	0.03	0.43	0.64
79	-0.04	0.26	0.08	0.06	0.04	0.08	0.02	0.06	0.20	0.03	0.15	0.15	0.20	0.20	0.35	0.16	0.62
80	0.02	0.07	0.08	0.03	0.13	0.05	0.09	0.16	0.14	0.26	0.07	0.16	0.03	0.26	0.19	0.36	0.10
81	0.03	0.05	0.32	0.04	0.12	0.12	0.06	0.06	0.19	0.10	0.12	0.33	0.12	0.33	0.30	0.41	0.56
82	0.00	0.11	0.04	0.04	0.26	0.15	0.06	0.10	0.13	0.18	0.10	0.12	0.09	0.18	0.11	0.10	0.48
83	0.06	0.06	0.02	0.04	0.11	0.08	0.05	0.05	0.21	0.06	0.24	0.18	0.20	0.05	0.17	0.42	0.47
84	'AMAT'	0.05	0.10	0.08	0.07	0.05	-0.01	0.13	0.16	0.06	0.18	0.11	0.17	0.21	0.25	0.26	0.50
85	0.03	'AMGN'	0.19	0.02	0.02	0.04	0.01	0.03	0.06	0.04	0.05	0.02	0.29	0.13	0.06	0.26	0.44
86	0.06	0.13	'ISRG'	0.10	0.10	0.00	0.08	0.07	0.06	0.02	0.09	0.06	0.19	0.28	0.20	0.27	0.46
87	0.05	0.01	0.06	'IBM'	0.11	0.09	0.12	0.09	0.11	0.05	0.03	0.22	0.25	0.25	0.31	0.40	0.44
88	0.04	0.02	0.07	0.07	'SNY'	0.02	0.16	0.01	0.09	0.02	0.13	0.28	0.14	0.22	0.20	0.37	0.40
89	0.03	0.03	0.00	0.06	0.01	'TGT'	0.05	0.01	0.06	0.10	0.19	0.04	0.09	0.14	0.18	0.33	0.43
90	-0.01	0.01	0.05	0.08	0.11	0.03	'TD'	0.12	0.02	0.19	0.02	0.13	0.03	0.20	0.33	0.30	0.63
91	0.09	0.02	0.04	0.06	0.01	0.01	0.08	'TTE'	-0.01	0.10	0.23	0.06	0.07	0.02	0.26	0.06	0.53
92	0.11	0.04	0.04	0.07	0.06	0.04	0.01	-0.01	'EL'	0.05	0.07	0.15	0.13	0.03	0.25	0.35	0.48
93	0.04	0.02	0.01	0.03	0.01	0.06	0.13	0.07	0.03	'CVS'	0.07	0.07	0.14	0.04	0.14	0.36	0.38
94	0.12	0.03	0.06	0.02	0.09	0.13	0.02	0.15	0.05	0.04	'GE'	0.23	0.18	0.18	0.25	0.38	0.46
95	0.07	0.01	0.04	0.15	0.19	0.02	0.09	0.04	0.10	0.05	0.16	'DEO'	0.03	0.18	0.24	0.19	0.66
96	0.11	0.20	0.13	0.17	0.10	0.06	0.02	0.05	0.09	0.09	0.12	0.02	'SPGI'	0.12	0.04	0.35	0.52
97	0.14	0.09	0.19	0.17	0.15	0.09	0.14	0.02	0.02	0.02	0.12	0.12	0.08	'RIO'	0.37	0.47	0.53
98	0.17	0.04	0.14	0.21	0.14	0.12	0.22	0.18	0.17	0.10	0.17	0.16	0.02	0.25	'JBHT'	0.29	0.58
99	0.17	0.18	0.18	0.28	0.25	0.22	0.21	0.04	0.24	0.24	0.26	0.13	0.24	0.33	0.20	'DE'	0.78
100	0.35	0.31	0.31	0.31	0.27	0.29	0.46	0.37	0.33	0.26	0.32	0.48	0.37	0.38	0.41	0.59	'CAT'

The result of the 100-dimensional stock index experiment, $\hat{\mathcal{M}}^{Mixed-R-vine}_{100-Dim}$ and $\hat{\mathcal{T}}^{Mixed-R-vine}_{100-Dim}$ of VBDA1 in the R-vine-ICAPM-EGARCH(1,1,1)-Mixture model is presented in the lower and upper triangular matrix, respectively, as follows.

Row/Column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	100	'Galambos'	'Clayton'	'Frank'	'Frank'	'Gaussian'	'Galambos'	'Galambos'	'Clayton'	'Gaussian'	't'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'
2	96	99	'Galambos'	'Frank'	'Frank'	'Galambos'	'Gaussian'	'Frank'	'Gumbel'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'
3	95	96	98	'BB10'	'Frank'	'Frank'	'Gaussian'	't'	'B5'	'Galambos'	'Gaussian'	'Frank'	'Gaussian'	'Galambos'	'Gaussian'	'Frank'	'Frank'
4	97	95	96	97	't'	't'	'BB10'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gumbel'	'Gaussian'	'Gaussian'	'Gaussian'	'Gumbel'	'B5'
5	91	97	95	96	96	'Gaussian'	'Gaussian'	't'	'Frank'	'Gaussian'	'BB10'	'Frank'	'Gaussian'	'Galambos'	'Gaussian'	't'	'Frank'
6	90	91	97	95	94	95	'Galambos'	'Frank'	'BB10'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Clayton'	'Gaussian'	'Frank'	'Gumbel'
7	89	90	91	94	93	94	94	'Gaussian'	'Clayton'	'Galambos'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Frank'
8	88	89	90	93	92	93	91	93	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Clayton'	'Frank'	'Gaussian'	'Gaussian'
9	85	88	89	92	87	92	90	91	92	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Galambos'	'Clayton'	'Galambos'
10	84	85	88	87	91	87	89	90	91	91	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Galambos'	'Frank'
11	74	84	85	86	90	91	88	89	90	87	90	'B5'	'Gaussian'	'Gaussian'	'Galambos'	'Gaussian'	'Frank'
12	73	74	84	88	89	90	85	88	89	86	87	89	'Gumbel'	'Gaussian'	'Frank'	'Gaussian'	'Frank'
13	82	73	74	85	86	89	84	85	88	88	86	87	88	'Galambos'	'Frank'	'Frank'	'Gaussian'
14	80	82	73	84	83	86	74	84	85	85	88	86	87	87	'Frank'	'Frank'	'Frank'
15	79	80	82	74	82	83	73	74	84	84	85	88	86	85	86	'Frank'	'Clayton'
16	78	79	80	73	81	82	82	73	74	74	84	85	83	84	85	85	'Clayton'
17	72	78	79	83	80	81	80	82	73	73	74	84	82	74	84	83	84
18	76	72	78	81	79	80	79	80	82	83	73	74	81	73	74	82	83
19	75	76	72	77	78	79	78	79	80	81	83	73	80	82	73	81	82
20	71	75	76	89	77	78	72	78	79	77	81	83	79	80	82	80	81
21	70	71	75	72	76	77	76	72	78	89	77	81	78	79	80	79	80
22	69	70	71	67	88	76	75	76	72	72	89	77	77	78	79	78	79
23	68	69	70	70	85	88	71	75	76	67	72	72	76	72	78	77	78
24	66	68	69	63	74	85	70	71	75	70	67	67	84	76	72	76	77
25	65	66	68	61	73	74	69	70	71	63	70	70	75	75	76	84	76
26	64	65	66	60	72	73	68	69	70	61	63	63	71	71	75	75	74
27	62	64	65	59	67	72	66	68	69	60	61	61	70	70	71	71	73
28	56	62	64	64	75	67	65	66	68	59	60	60	67	69	70	70	72
29	53	56	62	58	71	75	64	65	66	64	59	82	63	68	69	67	67
30	58	53	56	57	69	71	62	64	65	58	64	80	69	66	68	63	75
31	57	58	53	55	68	69	56	62	64	57	58	79	68	65	66	69	71
32	52	57	58	65	63	68	53	56	62	55	57	78	66	64	65	68	69
33	50	52	57	62	61	63	58	53	56	65	55	76	61	62	64	66	68
34	49	50	52	56	60	61	57	58	53	62	65	75	60	56	62	61	63
35	51	49	50	53	59	60	52	57	58	56	62	71	59	53	56	60	61
36	48	51	49	52	64	59	50	52	57	53	56	69	65	58	53	59	60
37	47	48	51	50	58	64	49	50	52	52	53	68	62	57	58	65	59
38	45	47	48	49	57	58	51	49	50	50	52	66	56	52	57	62	64
39	44	45	47	54	66	57	48	51	49	49	50	65	53	50	52	56	58
40	42	44	45	47	55	66	47	48	51	54	49	59	55	49	50	53	57
41	40	42	44	45	54	55	45	47	48	47	54	62	54	51	49	55	66
42	38	40	42	44	51	54	44	45	47	45	47	56	51	48	51	54	55
43	37	38	40	42	48	51	42	44	45	44	45	53	48	47	48	51	54
44	35	37	38	40	46	48	40	42	44	42	44	55	64	45	47	48	51
45	32	35	37	38	43	46	38	40	42	40	42	54	46	44	45	64	48
46	46	32	35	37	41	43	37	38	40	38	40	51	43	42	44	46	46
47	43	46	32	35	39	41	35	37	38	37	38	48	41	40	42	43	43
48	41	43	46	32	36	39	32	35	37	35	37	46	39	38	40	41	41
49	29	41	43	29	52	36	46	32	35	32	35	43	36	37	38	39	39
50	36	29	41	39	50	52	43	46	32	29	32	41	34	35	37	36	36
51	28	36	29	28	49	50	41	43	46	39	29	39	33	32	35	34	52
52	27	28	36	27	47	49	29	41	43	28	39	36	40	46	32	33	50
53	20	27	28	20	45	47	36	29	41	27	28	34	31	43	46	40	49
54	18	20	27	18	44	45	28	36	29	20	27	33	30	41	43	31	47
55	16	18	20	16	42	44	27	28	36	18	20	40	26	29	41	30	45
56	14	16	18	14	34	42	20	27	28	16	18	31	25	36	29	26	44
57	13	14	16	13	33	34	18	20	27	14	16	30	24	28	36	25	42
58	21	13	14	82	38	33	16	18	20	13	14	26	23	27	28	24	34
59	12	21	13	21	37	38	14	16	18	82	13	25	22	20	27	23	33
60	17	12	21	12	35	37	13	14	16	21	82	58	74	18	20	22	38
61	15	17	12	41	32	35	21	13	14	12	21	57	73	16	18	74	37
62	9	15	17	17	29	32	12	21	13	41	12	52	28	14	16	73	35
63	25	9	15	15	31	29	17	12	21	17	41	50	58	13	14	28	32
64	1	25	9	34	30	31	15	17	12	15	17	49	57	21	13	58	29
65	4	1	25	33	26	30	9	15	17	34	15	47	52	12	21	57	31
66	5	4	1	9	25	26	25	9	15	33	34	45	21	17	12	52	30
67	3	5	4	25	24	25	1	25	9	9	33	24	44	15	17	21	26
68	8	3	5	43	84	24	4	1	25	25	9	23	19	9	15	44	25
69	87	8	3	1	62	84	5	4	1	43	25	22	17	25	9	19	24
70	24	87	8	4	23	62	3	5	4	1	43	28	15	1	25	17	70
71	22	24	87	5	22	23	8	3	5	4	1	21	11	4	1	15	65
72	19	22	24	3	95	22	87	8	3	5	4	19	20	5	4	11	23
73	10	19	22	31	70	27	24	87	8	3	5	17	29	3	5	20	22
74	60	10	19	30	65	20	22	24	87	31	3	42	10	8	3	29	27
75	61	60	10	26	56	18	19	22	24	30	31	38	1	24	8	10	20
76	59	61	60	8	53	16	10	19	22	26	30	37	45	22	24	1	18
77	7	59	61	23	40	14	60	10	19	8	26	35	4	19	22	45	16
78	6	7	59	79	28	13	61	60	10	23	8	15	8	10	19	4	14
79	31	6	7	19	21	19	59	61	60	79	23	11	16	60	10	8	13
80	30	31	6	11	19	40	7	59	61	19	79	44	7	86	60	16	19
81	26	30	31	46	17	53	6	7	59	11	19	32	42	83	61	7	56
82	23	26	30	2	15	12	31	6	7	46	11	20	6	81	59	42	40
83	86	23	26	69	11	11	30	31	6	2	46	10	37	77	7	6	28
84	67	86	23	90	27	10	26	30	31	69	2	1	32	67	6	37	12
85	54	67	86	80	18	1	23	26	30	90	69	4	2	63	31	32	11
86	39	54	67	7	16	21	86	23	26	80	7	27	5	55	30	2	10
87	2	39	54	76	14	4	67	86	23	7	78	18	3	54	26	5	1
88	11	2	39	75	13	8	54	67	86	76	71	8	18	39	23	3	4
89	92	11	2	66	12	7	39	54	67	75	68	7	14	34	67	18	8
90	83	92	11	24	10	9	2	39	54	66	24	6	13	33	54	14	7
91	98	83	92	22	1	5	11	2	39	24	22	2	85	7	83	13	9
92	33	98	83	51	4	3	92	11	2	22	48	14	72	6	81	50	5
93	55	33	94	6	8	2	83	92	11	51	6	13	9	11	77	72	3
94	34	55	93	91	7	6	33	83	33	6	36	12	49	30	63	49	62
95	94	34	81	10	6	70	55	33	55	78	80	5	12	61	55	47	2
96	77	94	77	36	2	15	34	55	34	10	76	3	38	59	39	38	53
97	93	77	63	71	5	28	77	34	83	36	75	29	27	26	34	35	6
98	81	93	33	48	3	17	81	77	77	71	51	9	35	23	33	9	17
99	63	81	34	68	9	56	93	81	81	68	10	64	50	2	2	12	15
100	99	63	55	78	20	65	63	63	63	48	66	16	47	31	11	27	21

Row/Column	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
1	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Frank'	't'	'Galambos'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Frank'
2	'Frank'	't'	'Frank'	'Gaussian'	'Frank'	'B5'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Frank'
3	'Frank'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Clayton'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	't'	'Frank'
4	'Frank'	'Frank'	'Gaussian'	'Clayton'	'Gumbel'	'Frank'	'Gaussian'	't'	'Gaussian'	'Gaussian'	'Frank'	't'	'Galambos'	'Gaussian'	'Clayton'	'Frank'
5	'Frank'	'Frank'	'Frank'	't'	'Gumbel'	'Frank'	'Frank'	'Gumbel'	'Clayton'	'Frank'	'Frank'	'Frank'	't'	'Gumbel'	't'	'Gaussian'
6	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Galambos'	'Clayton'	'Gaussian'	'Gaussian'	'Frank'
7	'Gaussian'	'Galambos'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Gumbel'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Frank'
8	'Frank'	't'	'Gaussian'	'Frank'	'Galambos'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'
9	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Clayton'	'Clayton'	'BB10'	'Gumbel'	'Galambos'	'Gumbel'	'Gaussian'	'Galambos'	'Frank'	'Gaussian'	'Clayton'	'Frank'
10	'Gaussian'	'Gaussian'	'Gaussian'	'Gumbel'	'Frank'	't'	'Gaussian'	'Frank'	'B5'	'Gaussian'	'Frank'	'Gaussian'	't'	'Gaussian'	'Frank'	'Gaussian'
11	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Clayton'	'Frank'	'Frank'	'B5'	'Gaussian'	'Frank'	'Frank'	'Galambos'	'Gaussian'	'Gaussian'	'Gaussian'	'B5'
12	'Frank'	'Gaussian'	'Frank'	'Clayton'	'Gaussian'	'Frank'	'Gumbel'	'Clayton'	'Frank'	'Gumbel'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Gaussian'
13	'Frank'	'Gaussian'	't'	'Frank'	'Frank'	'Galambos'	't'	'Gaussian'	't'	't'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Gumbel'	'Gaussian'
14	'Frank'	'Frank'	'Frank'	'Frank'	'Frank'	'Gaussian'	't'	'Gaussian'	'Frank'	't'	't'	't'	'Gaussian'	'Gaussian'	'Gaussian'	'Galambos'
15	'Gaussian'	'Gaussian'	'BB10'	'Galambos'	't'	'Frank'	'Gaussian'	'Frank'	'Galambos'	't'	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Frank'	'B5'
16	'Gaussian'	'Galambos'	'Frank'	'Clayton'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Galambos'	'Frank'	'Frank'	'Gaussian'
17	'Clayton'	't'	'Gaussian'	'Gaussian'	't'	'Clayton'	'Gaussian'	'Gaussian'	'Gaussian'	'Galambos'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'
18	83	'Gaussian'	'Gaussian'	'Gaussian'	'Gumbel'	'Galambos'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'
19	74	82	'Clayton'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Clayton'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'
20	73	74	81	'Gaussian'	't'	'B5'	't'	'Clayton'	't'	'Gaussian'	'Frank'	'BB1'	't'	't'	'Gaussian'	'Gaussian'
21	82	73	74	80	'Gaussian'	'Gaussian'	'Gumbel'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'
22	80	81	73	74	79	'Gumbel'	't'	't'	'Frank'	'B5'	't'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Frank'
23	79	77	80	73	74	78	'Frank'	'Gumbel'	'BB10'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'BB10'
24	78	72	79	77	73	74	77	'Frank'	'Frank'	'BB1'	'Frank'	'Frank'	'Frank'	'Frank'	'Frank'	'Gaussian'
25	72	67	78	72	77	73	74	76	'Frank'	'Frank'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Frank'
26	76	70	72	67	72	77	73	74	75	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'
27	75	63	76	70	67	72	72	73	74	74	'Frank'	'Gaussian'	'Galambos'	'Frank'	'Gaussian'	'Frank'
28	71	61	75	63	70	67	76	72	73	71	73	't'	'Clayton'	't'	'Gaussian'	'Gaussian'
29	70	60	71	61	63	70	75	67	72	70	71	72	'Clayton'	'Gaussian'	'Frank'	'Gumbel'
30	69	59	70	60	61	63	71	70	67	67	70	71	71	'Gaussian'	'B5'	'Frank'
31	68	64	69	59	60	61	70	63	70	63	67	70	67	70	'Gaussian'	'Gaussian'
32	66	58	68	64	59	60	69	61	63	69	63	67	70	67	69	'Frank'
33	65	57	66	58	64	59	68	60	61	68	69	63	63	69	67	68
34	64	55	65	57	58	64	66	59	60	66	68	69	61	68	63	67
35	62	65	64	55	57	58	65	64	59	61	66	68	60	63	61	63
36	56	62	62	65	55	57	64	58	64	60	61	66	59	61	60	61
37	53	56	56	62	65	55	62	57	58	59	60	61	64	60	59	60
38	58	53	53	56	62	65	56	55	57	65	59	60	58	59	64	59
39	57	52	58	53	56	62	53	65	55	62	65	59	57	64	58	64
40	52	50	57	52	53	56	58	62	65	56	62	65	55	58	57	58
41	50	49	52	50	52	53	57	56	62	53	56	62	65	57	55	57
42	49	54	50	49	50	52	52	53	56	55	53	56	62	66	65	55
43	51	47	49	54	49	50	50	52	53	54	55	53	56	55	62	65
44	48	45	51	47	54	49	49	50	52	51	54	55	53	54	56	62
45	47	44	48	45	47	54	51	49	50	48	51	54	52	51	53	56
46	45	42	47	44	45	47	48	54	49	64	48	51	50	48	52	53
47	44	40	45	42	44	45	47	47	54	46	64	48	49	46	50	52
48	42	38	44	40	42	44	45	45	47	43	46	64	54	43	49	50
49	40	37	42	38	40	42	44	44	45	41	43	46	47	41	54	49
50	38	35	40	37	38	40	42	42	44	39	41	43	45	39	47	54
51	37	32	38	35	37	38	40	40	42	36	39	41	44	36	45	47
52	35	29	37	32	35	37	38	38	40	34	36	39	42	52	44	45
53	32	39	35	29	32	35	37	37	38	33	34	36	40	50	42	44
54	46	28	32	39	29	32	35	35	37	40	33	34	38	49	40	42
55	43	27	46	28	39	29	32	32	35	31	40	33	37	47	38	40
56	41	20	43	27	28	39	46	29	32	30	31	40	35	45	37	38
57	29	18	41	20	27	28	43	39	29	26	30	31	32	44	35	37
58	36	16	29	18	20	27	41	28	39	25	26	30	29	42	32	35
59	28	14	36	16	18	20	29	27	28	24	25	26	39	34	29	32
60	27	13	28	14	16	18	36	20	27	23	24	25	28	33	39	29
61	20	80	27	13	14	16	28	18	20	22	23	24	27	38	28	39
62	18	79	20	21	13	14	27	16	18	28	22	23	20	37	27	28
63	16	78	18	12	21	13	20	14	16	72	28	22	18	35	20	27
64	14	76	16	41	12	21	18	13	14	21	72	28	16	32	18	20
65	13	75	14	17	41	12	16	21	13	44	21	58	14	29	16	18
66	21	71	13	15	17	41	14	12	21	19	44	57	13	31	14	16
67	12	21	21	34	15	17	13	41	12	17	19	52	21	30	13	14
68	17	12	12	33	34	15	21	17	41	15	17	21	12	26	21	13
69	15	41	17	9	33	34	12	15	17	50	15	44	41	25	12	21
70	9	17	15	25	9	33	17	34	15	11	50	19	17	24	41	12
71	25	15	9	43	25	9	15	33	34	20	11	17	15	62	17	41
72	1	34	25	1	43	25	9	9	33	29	20	15	34	23	15	17
73	4	33	1	4	1	43	25	25	9	10	29	11	33	22	34	15
74	5	69	4	5	4	1	1	43	25	1	10	20	9	27	33	34
75	3	68	5	3	5	4	4	1	43	49	1	29	25	20	9	33
76	8	66	3	31	3	5	5	4	1	47	49	10	43	18	25	9
77	24	51	8	30	31	3	3	5	4	4	47	1	1	16	43	25
78	22	48	24	26	30	31	8	3	5	8	4	45	4	14	1	43
79	19	46	22	8	26	30	24	31	3	16	8	4	5	13	4	1
80	10	9	19	23	8	26	22	30	31	7	16	8	3	19	5	4
81	60	25	10	79	23	8	19	26	30	38	7	16	31	40	3	5
82	61	1	60	19	78	23	10	8	26	6	38	7	30	53	31	3
83	59	36	61	11	76	19	60	23	8	35	6	42	26	12	30	31
84	7	31	59	46	75	11	61	19	23	27	35	6	8	11	26	30
85	6	43	7	2	71	46	59	11	19	2	27	37	23	10	8	26
86	31	4	6	69	69	2	7	46	11	5	2	32	19	1	23	8
87	30	5	31	7	68	69	6	2	46	3	5	2	11	21	19	23
88	26	30	30	78	66	7	31	69	2	73	3	5	46	4	11	19
89	23	26	26	71	19	76	30	7	69	57	58	3	2	8	46	11
90	67	24	23	68	11	75	26	71	7	45	45	18	69	7	2	46
91	54	23	67	24	51	66	23	68	71	42	52	14	7	9	68	2
92	39	22	54	22	48	24	67	24	68	37	9	13	66	5	66	7
93	2	19	39	48	36	22	54	22	24	9	37	50	24	3	51	66
94	11	11	2	6	2	51	39	48	22	13	13	9	22	2	48	24
95	33	10	11	36	7	6	2	6	48	14	14	12	51	6	7	22
96	55	3	33	66	6	10	11	36	6	12	12	47	6	65	24	51
97	34	2	55	76	24	36	33	66	36	18	57	35	10	56	22	6
98	77	7	34	10	46	71	55	10	66	58	18	38	36	15	6	10
99	63	6	77	75	22	48	34	75	10	32	32	27	48	17	36	36
100	81	8	63	51	10	68	63	51	51	52	42	49	68	28	10	48

Row/Column	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
1	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Galambos'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Clayton'	'Gumbel'	'Galambos'
2	't'	'Clayton'	'Frank'	'Gaussian'	'Gaussian'	'Galambos'	'Gumbel'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Clayton'	'Galambos'	'Frank'
3	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Clayton'	'Gaussian'	'Gaussian'	'Frank'	'Galambos'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'
4	'Frank'	'B5'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Clayton'	'Frank'	'Frank'	'Frank'	'Galambos'	't'	'Frank'	'Frank'	'Frank'
5	'B5'	'Frank'	'Gaussian'	'BB1'	'Frank'	't'	'Frank'	'Frank'	'Gaussian'	'BB10'	'Clayton'	'Gaussian'	'Gaussian'	't'	'Clayton'	'Gaussian'
6	'Gaussian'	'Frank'	'Frank'	'Frank'	'Frank'	'Gaussian'	't'	'Frank'	'Frank'	't'	't'	'Galambos'	'Gaussian'	'Gaussian'	'Frank'	'Clayton'
7	'Gaussian'	'Gaussian'	'B5'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Clayton'	'Frank'	'Gaussian'
8	'Gaussian'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Clayton'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Gumbel'	'Gaussian'	'B5'	'Gaussian'	'Clayton'
9	'Frank'	'BB1'	'Frank'	'Frank'	't'	'Frank'	'Clayton'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	't'	'Frank'	'Gaussian'	'Gaussian'	'Frank'
10	't'	'Frank'	'Frank'	'Gaussian'	't'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Galambos'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Galambos'	't'
11	'Gaussian'	'Clayton'	'Frank'	'Galambos'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Frank'	't'	't'	't'
12	'Gumbel'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Galambos'	'Frank'	'Frank'	'Galambos'	'BB10'	'Gaussian'	'BB10'	'Gaussian'
13	'Frank'	't'	'Frank'	'Frank'	'Clayton'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Clayton'	'Gumbel'	'Gaussian'	'Clayton'	'Frank'
14	'Gaussian'	'B5'	'Clayton'	'Frank'	't'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	't'	'Gaussian'	'Galambos'	'Frank'	'Gaussian'	'Gaussian'	'Frank'
15	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	't'	't'	'Frank'	'Frank'	'Frank'	't'	'Gaussian'
16	'Gaussian'	'Galambos'	'Frank'	'Frank'	'Frank'	't'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Clayton'	'Frank'	'Frank'	'Frank'
17	'Frank'	'Gaussian'	'Gaussian'	'Frank'	't'	'Frank'	'Frank'	'Gaussian'	'Galambos'	'Gumbel'	'Gaussian'	'Frank'	't'	'Gaussian'	'Frank'	't'
18	'Frank'	'Gaussian'	'Clayton'	'B5'	'Frank'	'Gumbel'	'Gumbel'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Clayton'	'Gaussian'	'BB10'	't'	'Clayton'
19	't'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	't'	'Frank'	'Gaussian'	'Clayton'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Galambos'	'Gaussian'
20	'Clayton'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Galambos'	'Gaussian'	'Gaussian'	'Clayton'	'Gaussian'	't'	'B5'
21	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'B5'	'Gaussian'	'Gaussian'	'Gumbel'	'Frank'	't'	'Gaussian'
22	'Frank'	'Clayton'	'Clayton'	'Gumbel'	'Gaussian'	'Clayton'	'Gaussian'	't'	'Gaussian'	'Gaussian'	'Gaussian'	'Galambos'	'Gaussian'	'Clayton'	'B5'	'Gaussian'
23	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Clayton'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	't'
24	'Gumbel'	'Gaussian'	'Gaussian'	'Gaussian'	't'	't'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	't'	'Frank'	'Gaussian'	'Gaussian'
25	'Gaussian'	'Gumbel'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Galambos'	'Frank'	'Frank'	'Frank'
26	'Gaussian'	't'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Clayton'	'Frank'	'Gaussian'	'Gaussian'
27	'Gaussian'	'Gumbel'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Frank'	't'	'Galambos'	'Gaussian'	'BB10'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'
28	'Gaussian'	'Frank'	'Frank'	'Gumbel'	'Gumbel'	'Gaussian'	'Gaussian'	'Frank'	'Clayton'	'Galambos'	'Gaussian'	'Clayton'	't'	'Frank'	'Gaussian'	'Frank'
29	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Clayton'	'Frank'	'Gaussian'	'Galambos'	't'	'Clayton'	'Gaussian'	'Gaussian'	'Gaussian'
30	'Gaussian'	'Frank'	't'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Galambos'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'
31	'Gaussian'	'Gaussian'	't'	'Gaussian'	't'	'Frank'	't'	'Gaussian'	'Frank'	'Gaussian'	'Galambos'	'Gaussian'	'Gaussian'	'Gumbel'	'Galambos'	'Frank'
32	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	't'
33	'Galambos'	't'	'Gaussian'	'Frank'	'Clayton'	'Galambos'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Clayton'	'Gaussian'	't'
34	67	't'	'Gaussian'	'Frank'	't'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Clayton'	'Gaussian'	'Frank'	'Frank'	'Gaussian'
35	66	66	'Clayton'	'Gumbel'	'Gaussian'	'Galambos'	'Frank'	'B5'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	't'	'Gaussian'
36	65	63	65	't'	'Gumbel'	'Gaussian'	'Gaussian'	'Galambos'	'Gaussian'	'Clayton'	'Clayton'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Frank'
37	64	61	63	64	'BB10'	'Gaussian'	'Gaussian'	'Galambos'	'Frank'	't'	'BB10'	'Frank'	'Gumbel'	'Frank'	'Gaussian'	't'
38	62	60	61	63	63	'Frank'	'Frank'	'Clayton'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'
39	56	59	60	61	62	62	'Frank'	'Frank'	'Frank'	'Gaussian'	'Galambos'	'Frank'	'Gumbel'	'Frank'	't'	't'
40	53	64	59	60	56	61	61	'Frank'	't'	'Frank'	'Galambos'	'Galambos'	'Gaussian'	'B5'	'Gaussian'	'Frank'
41	58	58	64	59	53	60	56	60	'Frank'	'Frank'	'Gaussian'	'Clayton'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'
42	57	57	58	62	58	59	53	56	59	'BB1'	'Galambos'	'Gaussian'	't'	'Frank'	't'	'Gaussian'
43	52	55	57	56	57	58	58	53	56	58	'Gaussian'	't'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'
44	50	65	55	53	52	57	57	58	53	56	57	'Frank'	'Clayton'	'Gumbel'	'Gaussian'	'Gaussian'
45	49	62	54	55	50	55	52	57	58	53	56	56	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'
46	51	56	51	54	49	54	50	52	57	55	53	55	55	'Frank'	'Gaussian'	'Gaussian'
47	48	53	48	51	51	51	49	50	52	54	55	54	53	54	'Gaussian'	'Frank'
48	47	52	46	48	48	48	51	49	50	51	54	51	52	53	53	'Gaussian'
49	45	50	43	46	47	46	48	51	49	48	51	48	50	52	51	52
50	44	49	41	43	45	43	47	48	51	46	48	46	49	50	48	51
51	42	54	39	41	44	41	45	47	48	43	46	43	51	49	46	48
52	40	47	36	39	42	39	44	45	47	41	43	41	48	51	43	46
53	38	45	52	36	40	36	42	44	45	39	41	39	47	48	41	43
54	37	44	50	34	38	52	40	42	44	36	39	36	45	47	39	41
55	35	42	49	33	37	50	38	40	42	34	36	52	44	45	36	39
56	32	40	47	40	35	49	37	38	40	33	34	50	42	44	52	36
57	46	38	45	31	32	47	35	37	38	40	33	49	40	42	50	34
58	43	37	44	30	46	45	32	35	37	31	40	47	38	40	49	33
59	41	35	42	26	43	44	46	32	35	30	31	45	37	38	47	40
60	29	32	34	25	41	42	43	46	32	26	30	44	35	37	45	31
61	36	29	33	58	29	34	41	43	46	25	26	42	32	35	44	30
62	28	39	38	57	36	33	29	41	43	24	25	34	46	32	42	26
63	27	28	37	52	28	38	36	29	41	23	24	33	43	46	34	25
64	20	27	35	50	27	37	28	36	29	22	23	38	41	43	33	24
65	18	20	32	49	20	35	27	28	36	28	22	37	29	41	38	23
66	16	18	29	47	18	32	20	27	28	21	28	35	36	29	37	22
67	14	16	31	45	16	29	18	20	27	44	21	32	28	36	35	28
68	13	14	30	24	14	31	16	18	20	19	44	29	27	28	32	21
69	21	13	26	23	13	30	14	16	18	17	19	31	20	27	29	44
70	12	21	25	22	21	26	13	14	16	15	17	30	18	20	31	19
71	17	12	24	28	12	25	21	13	14	50	15	26	16	18	30	17
72	15	41	62	21	17	24	12	21	13	11	50	25	14	16	26	15
73	9	17	23	19	15	23	17	12	21	20	11	24	13	14	25	50
74	25	15	22	17	9	22	15	17	12	29	20	23	21	13	24	11
75	1	34	27	42	25	27	9	15	17	10	29	22	12	21	23	20
76	4	33	20	38	1	20	25	9	15	1	10	27	17	12	22	29
77	5	9	18	37	4	18	1	25	9	49	1	20	15	17	27	10
78	3	25	16	35	5	16	4	1	25	47	49	18	9	15	20	1
79	8	43	14	15	3	14	5	4	1	4	47	16	25	9	18	49
80	24	1	13	11	8	13	3	5	4	8	4	14	1	25	16	47
81	22	4	19	44	24	19	8	3	5	16	8	13	4	1	14	4
82	19	5	40	32	22	56	24	8	3	7	16	19	5	4	13	8
83	10	3	53	20	19	40	22	24	8	38	7	40	3	5	19	16
84	60	31	12	10	10	28	19	22	24	6	38	53	8	3	40	7
85	61	30	11	1	60	12	10	19	22	35	6	12	24	8	28	38
86	59	26	10	4	61	11	60	10	19	27	35	11	22	24	12	6
87	7	8	1	27	59	10	55	59	10	2	27	10	19	22	11	35
88	6	23	21	18	7	1	54	7	55	5	2	1	10	19	10	27
89	31	19	4	8	6	4	39	55	54	3	5	21	7	10	1	2
90	30	11	8	7	31	8	34	6	39	57	3	4	6	7	4	5
91	26	46	7	6	30	7	33	31	34	45	45	8	31	6	8	3
92	23	2	9	2	26	9	7	30	33	42	52	7	30	31	7	45
93	63	7	5	14	23	5	6	26	7	37	9	9	26	30	9	42
94	55	24	3	13	54	3	11	23	6	9	37	5	23	26	5	37
95	2	22	2	12	39	2	30	2	11	13	13	3	54	23	3	9
96	11	48	6	5	2	6	59	11	30	14	14	2	39	2	2	13
97	54	6	15	3	11	17	31	54	31	12	12	6	2	11	6	14
98	39	36	28	29	33	21	2	39	2	18	18	15	11	34	17	12
99	34	51	17	9	55	15	26	33	26	52	32	28	33	33	21	18
100	33	10	56	16	34	53	23	34	23	32	42	17	34	39	15	32

Row/Column	50	51	52	53	54	55	56	57
1	'Frank'	'Frank'	'Frank'	't'	't'	't'	'Gaussian'	'Frank'
2	'Frank'	'Gaussian'	'Gaussian'	'B5'	'Frank'	'Gaussian'	'Gaussian'	'Frank'
3	'Frank'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Clayton'	'Clayton'	'Frank'
4	'Frank'	'Gaussian'	't'	'BB1'	'Gaussian'	'Frank'	'Gumbel'	'Gaussian'
5	'Frank'	't'	'Gaussian'	'Frank'	'Frank'	'Frank'	't'	'Frank'
6	'Gaussian'	'Gaussian'	'Frank'	'Galambos'	'Frank'	'Frank'	'Frank'	'Frank'
7	'Gaussian'	'Gaussian'	'Frank'	'Gumbel'	'Gaussian'	'Frank'	'Frank'	'Gaussian'
8	'Frank'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'BB10'	'Gaussian'	'Frank'
9	'Gaussian'	'Galambos'	'Frank'	'Clayton'	'Gaussian'	't'	'Gaussian'	'Gaussian'
10	't'	'Galambos'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'BB10'
11	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'BB10'	't'	'Frank'
12	'BB1'	'Gumbel'	'BB10'	'B5'	'Frank'	'Gaussian'	'Gaussian'	'Frank'
13	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Frank'	'Gaussian'
14	'Gaussian'	't'	'Frank'	'Frank'	'Clayton'	'Frank'	'Gaussian'	'Gaussian'
15	'Gaussian'	't'	'Frank'	't'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'
16	'Gaussian'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Clayton'	'Gaussian'
17	'Gaussian'	'Frank'	'Gaussian'	't'	'Frank'	'Frank'	'Gaussian'	'Galambos'
18	'Gaussian'	'Galambos'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Clayton'
19	'B5'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	't'
20	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'BB10'
21	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'	't'	'Gaussian'
22	'Clayton'	'Frank'	't'	'Clayton'	'Frank'	'Frank'	'Gumbel'	'Frank'
23	'BB10'	'Galambos'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Gaussian'
24	'BB1'	'Gaussian'	'Clayton'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Clayton'
25	'Frank'	'Gaussian'	'Frank'	'Frank'	'Galambos'	'Galambos'	'Frank'	'Frank'
26	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Gaussian'	'Gaussian'	'Clayton'	'Gaussian'
27	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Frank'
28	't'	'Gumbel'	'Gaussian'	'Frank'	'Gumbel'	'Gaussian'	't'	'BB1'
29	'Frank'	'Frank'	'Frank'	'Galambos'	'Frank'	'Frank'	'BB10'	'Gaussian'
30	'Gumbel'	'Frank'	'BB10'	'BB10'	'Frank'	'Galambos'	'Gaussian'	'Gaussian'
31	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	't'	't'	't'	'Frank'
32	'Frank'	'Gaussian'	'BB10'	'Gaussian'	'Gaussian'	'Clayton'	'Gaussian'	'Galambos'
33	'Gaussian'	'Frank'	't'	't'	'Gumbel'	'Gaussian'	'Gaussian'	'Frank'
34	't'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	't'
35	'Frank'	'Frank'	'Gaussian'	'Frank'	'Galambos'	'Gumbel'	'B5'	'Gaussian'
36	'Gaussian'	'Gaussian'	'Frank'	'B5'	'Gaussian'	'Clayton'	'BB1'	'Frank'
37	'Frank'	't'	'Gaussian'	'BB10'	'Gaussian'	'Frank'	'Gaussian'	'Frank'
38	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Frank'
39	'Frank'	'Frank'	'Gaussian'	'Galambos'	'Gaussian'	't'	't'	't'
40	'Galambos'	'Frank'	'Galambos'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'
41	'Frank'	'Frank'	'Gaussian'	'Frank'	'Clayton'	'B5'	'Gaussian'	'Clayton'
42	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'
43	'Gaussian'	'Clayton'	't'	'Frank'	'Gaussian'	'Galambos'	'Gaussian'	'Frank'
44	'Gumbel'	'Frank'	'Frank'	't'	'BB10'	't'	'Gaussian'	'Frank'
45	't'	'BB10'	'Gaussian'	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Frank'
46	'Frank'	'Frank'	'Frank'	't'	'Gaussian'	'Frank'	'Gaussian'	'Frank'
47	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	't'	'Gaussian'
48	'Galambos'	'Galambos'	'BB10'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'
49	'Frank'	't'	'Gumbel'	'BB7'	'Gaussian'	'Gaussian'	'Clayton'	'Clayton'
50	51	'Gaussian'	'Frank'	'Frank'	'Gaussian'	't'	't'	'Frank'
51	50	50	'Gaussian'	'Gaussian'	'Gumbel'	'Clayton'	'Gaussian'	'Gaussian'
52	49	48	49	'Gaussian'	'Clayton'	'Frank'	't'	'Gaussian'
53	47	46	48	48	'Gaussian'	'Frank'	'Gaussian'	'Clayton'
54	45	43	46	47	47	'Frank'	'BB10'	'Gaussian'
55	44	41	43	45	46	46	'Gaussian'	'Frank'
56	42	39	41	44	43	45	45	'Gaussian'
57	40	36	39	42	41	44	43	44
58	38	34	36	40	39	42	41	43
59	37	33	34	38	36	40	39	41
60	35	40	33	37	34	38	36	39
61	32	31	40	35	33	37	34	36
62	29	30	31	32	40	35	33	34
63	39	26	30	29	31	32	40	33
64	28	25	26	39	30	29	31	40
65	27	24	25	28	26	39	30	31
66	20	23	24	27	25	28	26	30
67	18	22	23	20	24	27	25	26
68	16	28	22	18	23	20	24	25
69	14	21	28	16	22	18	23	24
70	13	44	21	14	28	16	22	23
71	21	19	44	13	21	14	28	22
72	12	17	19	21	44	13	21	28
73	41	15	17	12	19	21	44	21
74	17	11	15	41	17	12	19	19
75	15	20	11	17	15	41	17	17
76	34	29	20	15	11	17	15	42
77	33	10	29	34	20	15	11	38
78	9	1	10	33	29	34	20	37
79	25	45	1	9	10	33	29	35
80	43	4	45	25	1	9	10	15
81	1	8	4	43	45	25	1	11
82	4	16	8	1	4	43	4	20
83	5	7	16	4	8	1	8	29
84	3	42	7	5	16	4	16	10
85	31	6	42	3	7	5	7	1
86	30	37	6	31	42	3	38	4
87	26	32	37	30	6	31	6	8
88	8	2	32	26	37	30	35	16
89	23	5	2	8	32	26	27	7
90	19	3	5	23	2	8	2	32
91	11	18	3	19	5	23	5	6
92	46	14	18	11	3	19	3	2
93	2	13	14	46	18	11	42	5
94	7	9	13	2	14	36	9	3
95	24	49	9	7	13	2	13	18
96	22	12	12	24	9	7	32	14
97	48	38	47	22	12	6	14	27
98	6	27	35	6	38	24	12	9
99	36	35	38	10	27	22	18	12
100	10	47	27	36	35	10	37	13

Row/Column	58	59	60	61	62	63	64	65	66
1	'Frank'	'Frank'	't'	'Frank'	'Gumbel'	'Galambos'	'Galambos'	'Gaussian'	'Gaussian'
2	'Gumbel'	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Frank'	'Gaussian'	'Clayton'	'Galambos'
3	'Gaussian'	'Frank'	'Frank'	't'	't'	'Frank'	'B5'	'Gumbel'	'Gaussian'
4	'Frank'	'Gaussian'	'Frank'	'Frank'	'Gumbel'	'Clayton'	'Frank'	'Frank'	'Galambos'
5	'Gaussian'	'Frank'	't'	'Gaussian'	'Gaussian'	't'	'Gumbel'	'BB10'	'Gumbel'
6	'Gaussian'	'Gaussian'	'Clayton'	'Galambos'	'Frank'	'Gaussian'	'Frank'	'Clayton'	'Gumbel'
7	'Frank'	'Galambos'	'Gumbel'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Clayton'	'Gaussian'
8	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	't'	'Frank'	'Gaussian'
9	'Gaussian'	'Frank'	'B5'	't'	'Clayton'	'Frank'	'Gaussian'	'Frank'	'Gaussian'
10	'Gaussian'	'B5'	't'	't'	'Frank'	'Galambos'	't'	'Frank'	'Gaussian'
11	'Gaussian'	'Frank'	'Frank'	'Galambos'	'Frank'	't'	'Gaussian'	'Frank'	'Frank'
12	'Gaussian'	'Frank'	'Frank'	'Frank'	'Frank'	'Frank'	'BB10'	'Frank'	'Gaussian'
13	'Galambos'	't'	'Galambos'	'Gaussian'	'Gumbel'	't'	'BB10'	'Gaussian'	'Frank'
14	'Gaussian'	'Galambos'	'Frank'	't'	't'	'Frank'	'Gaussian'	'Clayton'	'Frank'
15	'Frank'	'Gumbel'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	't'
16	'Gaussian'	'Gaussian'	't'	'Frank'	'Frank'	'Frank'	'Frank'	'Frank'	'Gaussian'
17	'Frank'	'Galambos'	'Galambos'	'Clayton'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Gaussian'
18	'Gaussian'	't'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'
19	'Frank'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Gumbel'	'Frank'	'Gaussian'
20	'Gumbel'	'Gaussian'	't'	'BB10'	'Gaussian'	'Clayton'	't'	'Frank'	'Galambos'
21	'Gumbel'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	't'	'Gumbel'
22	'Frank'	'BB1'	'Frank'	'Gaussian'	't'	'Gaussian'	'Frank'	't'	'Gaussian'
23	'Frank'	'Gaussian'	'Frank'	'Gaussian'	't'	'Gaussian'	'Gaussian'	'Gaussian'	'Gaussian'
24	'BB10'	't'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'B5'
25	'Frank'	'Frank'	'Frank'	'Frank'	'Frank'	't'	'Gaussian'	'Gaussian'	'Gaussian'
26	'Frank'	'Clayton'	'Frank'	'B5'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Gaussian'
27	'Frank'	'Gaussian'	'Frank'	'Gaussian'	't'	'Gaussian'	'Frank'	'Frank'	'Galambos'
28	'Gaussian'	'Galambos'	'Gumbel'	'Gaussian'	'Frank'	't'	'Clayton'	'Gaussian'	't'
29	'Frank'	'Gaussian'	'Galambos'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Frank'
30	'Frank'	'Frank'	'Frank'	'Frank'	'Galambos'	'B5'	'Frank'	'Frank'	'Gumbel'
31	'B5'	'Frank'	'Gaussian'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Galambos'	'Frank'
32	'Frank'	'Frank'	'Gaussian'	'Frank'	'Clayton'	'Frank'	'Clayton'	'Clayton'	't'
33	'Gaussian'	't'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	't'	't'	'Gaussian'
34	'Gaussian'	'Frank'	'Frank'	't'	'Clayton'	'Frank'	'Frank'	'Frank'	'Frank'
35	'Gaussian'	'Gaussian'	'Gaussian'	't'	'Gumbel'	't'	't'	'Gaussian'	'B5'
36	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Gaussian'	'Clayton'	'Gaussian'
37	'Gaussian'	'Frank'	'Clayton'	'Gaussian'	'Gaussian'	'Gaussian'	'Frank'	't'	'Gaussian'
38	't'	'Frank'	'Gaussian'	't'	'Gaussian'	't'	'Gaussian'	't'	'Gaussian'
39	'Gaussian'	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Frank'	't'	'Galambos'	'B5'
40	'Gaussian'	'Galambos'	'Gaussian'	'Galambos'	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Gaussian'
41	'Gaussian'	'Gaussian'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Galambos'	'Gaussian'
42	'Frank'	'Gumbel'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Clayton'	'Frank'	'Frank'
43	'Frank'	'Galambos'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gumbel'	'Gaussian'
44	'Frank'	'Frank'	'Gaussian'	'Galambos'	'Gaussian'	'Frank'	'Galambos'	'Frank'	'Frank'
45	'Gaussian'	'Gaussian'	't'	'Gaussian'	't'	'Gaussian'	't'	't'	'Gaussian'
46	'Frank'	'B5'	'B5'	'Gaussian'	'B5'	'Gaussian'	'Gaussian'	'Frank'	'Galambos'
47	'Frank'	'Frank'	'Gaussian'	'Galambos'	'Frank'	'Frank'	'Gaussian'	'Galambos'	'Gaussian'
48	'Clayton'	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Frank'	'Gaussian'	'Gaussian'
49	'Gaussian'	'Gaussian'	't'	'Gaussian'	'Gumbel'	'Frank'	'Frank'	'Gaussian'	'Gaussian'
50	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	't'	'Gaussian'	't'	'Frank'	'Gaussian'
51	'Frank'	'Frank'	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	't'	't'	'Gaussian'
52	'Gaussian'	'Frank'	't'	'Gaussian'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Clayton'
53	'Frank'	'Frank'	'Clayton'	'Gaussian'	'Gaussian'	'Gaussian'	't'	'Clayton'	'Gaussian'
54	'Frank'	't'	'Clayton'	'Frank'	'Frank'	'Gaussian'	'Frank'	'Frank'	'Frank'
55	'BB10'	'Gaussian'	'Gaussian'	'Clayton'	'Gaussian'	'Gaussian'	'B5'	'Gumbel'	'Gaussian'
56	'Gaussian'	'Frank'	'Frank'	't'	'Gumbel'	't'	'Gumbel'	't'	'Galambos'
57	'Frank'	't'	'Frank'	'Gaussian'	'Frank'	'Clayton'	't'	'Gaussian'	'Frank'
58	43	'Gumbel'	't'	'Galambos'	't'	'Gumbel'	't'	'Clayton'	'Gaussian'
59	42	42	'Gaussian'	'Frank'	't'	't'	'Gaussian'	'Gaussian'	'Frank'
60	40	41	41	'Galambos'	'Frank'	'Gaussian'	'Clayton'	'Gaussian'	't'
61	38	39	40	40	'Frank'	'Gaussian'	'Gaussian'	'Gaussian'	'Gumbel'
62	37	36	38	39	39	'Gaussian'	'Gaussian'	'Frank'	'Frank'
63	35	34	37	36	38	38	'Frank'	'Galambos'	'Gaussian'
64	32	33	35	34	37	36	37	'Gaussian'	'Gaussian'
65	29	40	32	33	35	34	36	36	'Gaussian'
66	39	31	29	38	32	33	34	35	35
67	28	30	39	37	29	31	33	32	34
68	27	26	28	35	36	30	31	29	33
69	20	25	27	32	28	26	30	28	31
70	18	24	20	29	27	25	26	27	30
71	16	23	18	31	20	24	25	20	26
72	14	22	16	30	18	23	24	18	25
73	13	28	14	26	16	22	23	16	24
74	21	21	13	25	14	28	22	14	23
75	12	19	21	24	13	21	28	13	22
76	41	17	12	23	21	19	21	21	28
77	17	15	36	22	12	17	19	12	21
78	15	11	34	27	17	15	17	17	19
79	34	20	33	20	15	11	15	15	17
80	33	29	31	18	9	20	11	34	15
81	9	10	17	16	25	29	20	33	11
82	25	1	30	14	1	10	29	9	20
83	1	4	15	13	4	1	10	25	29
84	36	8	9	19	5	4	1	1	10
85	31	16	26	12	3	8	4	4	1
86	30	7	24	11	8	16	8	5	4
87	26	38	23	10	24	7	16	3	8
88	24	6	22	1	22	6	7	31	16
89	23	35	19	21	19	37	6	30	7
90	22	27	11	4	10	32	35	26	6
91	19	2	10	8	7	2	27	8	32
92	4	5	8	7	6	5	2	23	2
93	11	3	25	9	31	3	5	19	5
94	5	9	7	5	30	18	3	11	3
95	3	37	6	3	26	14	9	2	18
96	10	13	1	2	23	13	13	7	14
97	8	14	5	28	2	9	32	24	13
98	7	12	2	17	11	12	14	22	9
99	6	18	3	15	34	35	12	6	12
100	2	32	4	6	33	27	18	10	27

RowColumn	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83
1	’Gaussian’	’Gaussian’	’Clayton’	’Clayton’	’B5’	’Gaussian’	’Gaussian’	’Gumbel’	’Frank’	’Frank’	’Galambos’	’Frank’	’Gaussian’	’B5’	’Gaussian’	’Gaussian’	’Clayton’
2	’Frank’	’Galambos’	’Frank’	’Galambos’	’Gaussian’	’B5’	’Frank’	’Galambos’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Gumbel’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’
3	’Gaussian’	’Frank’	’Galambos’	’Frank’	’Clayton’	’Galambos’	’Frank’	’t’	’Gumbel’	’Frank’	’Gumbel’	’Gaussian’	’t’	’Gaussian’	’Gumbel’	’Gumbel’	’Frank’
4	’BB10’	’Clayton’	’Gaussian’	’Galambos’	’Frank’	’Gaussian’	’BB7’	’Frank’	’BB10’	’Frank’	’Galambos’	’Frank’	’t’	’Clayton’	’Frank’	’Frank’	’Frank’
5	’Gaussian’	’Frank’	’Frank’	’Galambos’	’Gaussian’	’Gaussian’	’BB10’	’Frank’	’t’	’t’	’BB10’	’Gaussian’	’Frank’	’Frank’	’t’	’Frank’	’Frank’
6	’BB10’	’Clayton’	’Frank’	’t’	’Gaussian’	’BB10’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Galambos’
7	’Frank’	’Frank’	’Gaussian’	’Frank’	’Galambos’	’Gaussian’	’Gumbel’	’B5’	’t’	’B5’	’BB10’	’t’	’B5’	’Gaussian’	’Clayton’	’Frank’	’BB10’
8	’Galambos’	’Frank’	’BB10’	’Clayton’	’Gaussian’	’Gaussian’	’Frank’	’Clayton’	’Frank’	’Frank’	’Gaussian’	’BB10’	’BB10’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’
9	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’t’	’t’	’Frank’	’Gaussian’
10	’Gaussian’	’t’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’BB10’	’Clayton’	’Gaussian’	’t’	’Frank’	’t’	’BB10’	’Frank’	’Gaussian’	’Frank’	’Frank’
11	’Clayton’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’t’	’t’	’Gaussian’	’Frank’	’Gaussian’	’BB1’	’Gaussian’	’Frank’	’BB7’	’Frank’	’Frank’
12	’Frank’	’Gaussian’	’Clayton’	’Frank’	’Gaussian’	’Clayton’	’Frank’	’t’	’Gaussian’	’Frank’	’Gaussian’	’t’	’Frank’	’Frank’	’Frank’	’Gaussian’	’B5’
13	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Galambos’	’Galambos’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Clayton’	’Gaussian’
14	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’BB10’	’Gaussian’	’Gaussian’	’Frank’	’Clayton’	’t’	’Frank’	’Frank’	’Gumbel’
15	’Gaussian’	’Gaussian’	’Gaussian’	’Clayton’	’Gumbel’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Clayton’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Clayton’
16	’BB10’	’t’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’BB10’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’t’	’Galambos’	’Frank’	’Frank’	’Gaussian’
17	’B5’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’t’	’Frank’	’Galambos’	’Gaussian’	’Galambos’	’t’	’Frank’	’Frank’	’Frank’	’Frank’	’B5’
18	’Frank’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’BB10’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’t’
19	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’t’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’t’	’Frank’	’Frank’
20	’Gaussian’	’Gaussian’	’Clayton’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Gaussian’	’t’	’Clayton’	’Gaussian’	’Frank’	’Frank’	’BB7’	’Frank’
21	’B5’	’Frank’	’t’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Clayton’	’t’	’Clayton’	’Gaussian’	’Frank’	’t’	’Frank’	’BB10’	’Gaussian’
22	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’B5’	’Gaussian’	’Frank’	’Clayton’	’Gaussian’	’Frank’	’B5’	’Frank’	’Gaussian’	’Frank’	’BB10’
23	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’Gaussian’	’Galambos’	’Frank’	’Galambos’	’Frank’	’Clayton’	’t’	’Gaussian’	’Gaussian’	’Frank’
24	’Frank’	’t’	’Frank’	’Gaussian’	’Frank’	’BB7’	’Gumbel’	’Frank’	’Gaussian’	’BB4’	’BB10’	’Frank’	’t’	’Frank’	’Galambos’	’Gaussian’	’Clayton’
25	’Gaussian’	’Galambos’	’t’	’Frank’	’Gaussian’	’Clayton’	’Frank’	’Galambos’	’t’	’Galambos’	’Clayton’	’t’	’Gumbel’	’Frank’	’BB10’	’Frank’	’Frank’
26	’Gaussian’	’Frank’	’Clayton’	’t’	’Gaussian’	’Gaussian’	’BB10’	’B5’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Clayton’	’BB10’	’Frank’	’Frank’	’Gaussian’
27	’Frank’	’t’	’Frank’	’Gaussian’	’Galambos’	’BB10’	’t’	’Gaussian’	’t’	’Gaussian’	’Galambos’	’t’	’t’	’Galambos’	’Frank’	’Frank’	’t’
28	’Gaussian’	’Gaussian’	’Gaussian’	’BB1’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Clayton’	’Gaussian’	’Gaussian’	’Frank’
29	’Gaussian’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Gaussian’
30	’Frank’	’Frank’	’Frank’	’Frank’	’BB10’	’Frank’	’Frank’	’t’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Galambos’	’BB1’	’Gaussian’	’Frank’
31	’Gaussian’	’Gumbel’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’Clayton’	’Gaussian’	’Frank’	’BB10’	’Gaussian’	’t’	’Clayton’	’Frank’	’Frank’	’Clayton’
32	’Gaussian’	’t’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’BB10’	’t’	’Frank’	’Gaussian’	’Gumbel’	’Frank’	’Gaussian’	’Clayton’	’t’
33	’Gaussian’	’Frank’	’Clayton’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’t’	’Frank’	’t’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Gaussian’
34	’t’	’BB10’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Frank’	’Gaussian’	’BB10’	’Frank’	’Frank’	’t’	’t’	’t’	’Gaussian’
35	’Gaussian’	’Frank’	’t’	’Gaussian’	’t’	’Gaussian’	’Gumbel’	’Gaussian’	’Frank’	’Galambos’	’Gumbel’	’Gaussian’	’Clayton’	’Frank’	’Gaussian’	’BB10’	’B5’
36	’t’	’Gaussian’	’Gaussian’	’Clayton’	’Frank’	’Gaussian’	’Frank’	’Clayton’	’Frank’	’Frank’	’Frank’	’Clayton’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’t’
37	’t’	’t’	’Gaussian’	’t’	’t’	’Frank’	’Frank’	’Gaussian’	’t’	’Gaussian’	’Gaussian’	’Clayton’	’Frank’	’t’	’t’	’Gaussian’	’t’
38	’Frank’	’Frank’	’BB4’	’Gaussian’	’Gaussian’	’Gaussian’	’Clayton’	’Frank’	’Frank’	’t’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’t’	’Gaussian’	’t’
39	’Frank’	’Gaussian’	’t’	’Galambos’	’Gumbel’	’Gaussian’	’Frank’	’Galambos’	’t’	’t’	’Frank’	’Frank’	’t’	’Frank’	’t’	’Gaussian’	’BB10’
40	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Clayton’	’Clayton’	’t’	’Gaussian’	’Clayton’	’Galambos’	’Gaussian’	’Clayton’	’BB10’	’Gumbel’	’Frank’	’Gaussian’	’BB1’
41	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Clayton’	’Clayton’	’Gaussian’	’BB7’	’Frank’	’t’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’
42	’Frank’	’Frank’	’BB10’	’Gaussian’	’Frank’	’Galambos’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Galambos’	’Gaussian’	’BB10’	’Gaussian’	’BB10’
43	’Frank’	’Gaussian’	’BB10’	’Frank’	’Frank’	’Galambos’	’Gaussian’	’t’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Clayton’	’Gaussian’	’BB10’
44	’Clayton’	’Frank’	’t’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Galambos’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Galambos’	’Gaussian’	’Frank’	’Frank’	’Gaussian’
45	’Gaussian’	’Gaussian’	’Gaussian’	’Gumbel’	’Frank’	’Frank’	’t’	’Gaussian’	’t’	’t’	’Gumbel’	’t’	’Clayton’	’Gumbel’	’Clayton’	’Frank’	’Frank’
46	’Gaussian’	’B5’	’Galambos’	’B5’	’B5’	’BB7’	’Frank’	’Frank’	’Frank’	’t’	’Gumbel’	’Gaussian’	’Gaussian’	’B5’	’Gaussian’	’Frank’	’Frank’
47	’Frank’	’Clayton’	’Gaussian’	’t’	’Gaussian’	’t’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’t’	’Gaussian’	’Gaussian’
48	’Clayton’	’BB10’	’Frank’	’Gaussian’	’t’	’Gaussian’	’BB10’	’t’	’Frank’	’Gaussian’	’t’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Gaussian’
49	’Gaussian’	’Frank’	’Clayton’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’t’	’t’	’Gaussian’	’t’	’Gaussian’	’Frank’	’Gaussian’	’t’	’Frank’
50	’Gaussian’	’t’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Galambos’	’Gaussian’	’t’	’Frank’	’Gaussian’	’Gaussian’	’BB10’	’Gaussian’	’t’	’Gaussian’
51	’t’	’Gaussian’	’t’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Clayton’	’Gaussian’	’Frank’	’BB10’	’Galambos’	’Gaussian’	’Gumbel’
52	’t’	’BB10’	’t’	’Gaussian’	’Frank’	’Frank’	’Clayton’	’t’	’Gaussian’	’t’	’Frank’	’Gaussian’	’t’	’t’	’Gaussian’	’Gaussian’	’t’
53	’t’	’Gaussian’	’t’	’t’	’Frank’	’Gaussian’	’Clayton’	’Gumbel’	’Gaussian’	’Frank’	’t’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Gumbel’	’B5’
54	’Gaussian’	’Gaussian’	’Clayton’	’Clayton’	’Gaussian’	’Frank’	’Frank’	’Clayton’	’Frank’	’Frank’	’Clayton’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’BB10’
55	’Gaussian’	’t’	’Clayton’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’BB10’	’Frank’	’BB10’	’Frank’	’Clayton’	’Frank’	’Gaussian’	’Gaussian’
56	’Gaussian’	’Gaussian’	’t’	’t’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’t’	’t’	’Frank’	’Clayton’	’BB10’	’Gaussian’	’Frank’	’Gaussian’
57	’Gumbel’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’t’	’Clayton’	’Frank’	’Frank’
58	’Gaussian’	’Gumbel’	’Gaussian’	’Gaussian’	’BB7’	’t’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’t’	’Gaussian’	’Frank’	’Frank’	’t’	’Frank’
59	’t’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’t’	’Frank’	’t’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’t’
60	’Frank’	’Frank’	’t’	’Clayton’	’Gaussian’	’Clayton’	’Frank’	’BB1’	’Frank’	’Clayton’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’
61	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Gumbel’	’Galambos’	’B5’	’Frank’	’Gaussian’	’Frank’	’Gumbel’	’Gaussian’	’Frank’	’Frank’	’Frank’
62	’Gaussian’	’Gaussian’	’Frank’	’BB1’	’Gaussian’	’Gumbel’	’Frank’	’Gaussian’	’Gumbel’	’Frank’	’Frank’	’BB10’	’Frank’	’BB10’	’Gaussian’	’Gaussian’	’Gumbel’
63	’Gaussian’	’Galambos’	’Frank’	’Gaussian’	’t’	’BB10’	’t’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’Gaussian’	’t’
64	’Frank’	’Clayton’	’t’	’Gaussian’	’Frank’	’Galambos’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Gumbel’	’Frank’	’Gumbel’
65	’Frank’	’t’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Gumbel’	’Frank’	’Frank’	’Galambos’	’Gaussian’	’Gaussian’	’Galambos’	’Gaussian’	’Frank’	’Frank’
66	’Gaussian’	’Frank’	’t’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Clayton’	’Gaussian’	’Gaussian’	’Frank’	’t’	’Gaussian’	’Gaussian’
67	34	’Gaussian’	’Frank’	’t’	’Gumbel’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Gumbel’	’BB10’	’Frank’	’Frank’	’Frank’
68	32	33	’Gaussian’	’Gaussian’	’Galambos’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Galambos’	’Frank’	’Gumbel’	’Gaussian’
69	29	32	32	’Gumbel’	’Gaussian’	’Frank’	’t’	’B5’	’Gaussian’	’Frank’	’Frank’	’t’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’BB10’
70	28	29	31	31	’Frank’	’Gaussian’	’BB10’	’t’	’BB10’	’Frank’	’Frank’	’Gaussian’	’Clayton’	’Frank’	’Gaussian’	’BB10’	’t’
71	27	28	30	29	30	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Galambos’	’Gaussian’	’BB10’	’Frank’	’Frank’	’Gaussian’	’Clayton’
72	20	27	26	28	29	29	’Gaussian’	’Frank’	’Clayton’	’BB10’	’Frank’	’t’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Frank’
73	18	20	25	27	28	26	28	’Frank’	’Frank’	’Clayton’	’t’	’Frank’	’t’	’t’	’Frank’	’B5’	’Frank’
74	16	18	24	20	27	25	26	27	’Gaussian’	’Gumbel’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’t’
75	14	16	23	18	20	24	25	26	26	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’t’
76	13	14	22	16	18	23	24	25	20	25	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Gumbel’	’Gaussian’	’Frank’
77	21	13	28	14	16	22	23	24	18	20	24	’Frank’	’Frank’	’BB10’	’Gaussian’	’Gaussian’	’Gaussian’
78	12	21	21	13	14	28	22	23	16	18	20	23	’Frank’	’Frank’	’Frank’	’Gaussian’	’Gaussian’
79	17	12	19	21	13	21	27	22	14	16	18	20	22	’BB10’	’Gaussian’	’Gaussian’	’BB10’
80	15	17	17	12	21	19	20	21	13	14	16	18	20	21	’Frank’	’Gaussian’	’Frank’
81	9	15	15	17	12	17	18	19	21	13	14	16	18	20	20	’Frank’	’t’
82	25	9	11	15	17	15	16	17	12	21	13	14	16	18	19	19	’BB10’
83	1	25	20	9	15	11	14	15	17	12	21	13	14	16	17	18	18
84	4	1	29	25	9	20	13	11	15	17	12	21	13	14	15	16	17
85	5	4	10	1	25	10	19	20	9	15	17	12	21	13	11	14	15
86	3	5	1	4	1	1	12	10	25	9	15	17	12	19	18	13	11
87	8	3	4	5	4	4	11	1	1	24	9	15	17	12	16	12	10
88	24	8	8	3	5	27	10	4	4	23	1	9	15	11	14	17	1
89	22	24	16	8	3	18	1	8	5	22	4	1	9	10	13	15	4
90	19	22	7	24	8	8	21	16	3	19	5	4	1	1	12	9	8
91	10	19	6	22	24	7	4	7	8	11	3	5	4	4	10	1	16
92	7	10	27	19	22	6	8	6	24	10	8	3	5	8	1	4	7
93	6	7	2	10	19	2	7	2	22	8	23	8	3	7	4	5	6
94	31	6	5	7	10	14	9	5	19	7	19	22	8	9	8	3	2
95	30	31	3	6	7	13	5	3	10	6	11	19	19	5	7	8	5
96	26	30	9	11	6	12	3	18	7	1	2	10	11	3	6	11	3
97	23	26	13	30	11	5	2	14	6	4	7	7	2	2	2	2	9
98	2	23	14	26	26	3	6	13	11	2	6	6	7	6	5	7	13
99	11	2	12	23	23	9	15	9	2	3	10	11	6	17	3	6	14
100	33	11	18	2	2	16	17	12	23	5	22	2	10	15	9	10	12

Row/Column	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	’Gaussian’	’Gaussian’	’t’	’Gaussian’	’BB10’	’t’	’Galambos’	’t’	’Frank’	’Frank’	’t’	’BB1’	’Frank’	’Frank’	’BB10’	’t’	’t’
2	’Frank’	’Gaussian’	’Galambos’	’Frank’	’Gaussian’	’Frank’	’t’	’Frank’	’t’	’Gaussian’	’Gaussian’	’Frank’	’t’	’t’	’t’	’t’	’t’
3	’Gaussian’	’t’	’Frank’	’Gumbel’	’BB10’	’Gaussian’	’Gaussian’	’BB10’	’Gumbel’	’Frank’	’Frank’	’Frank’	’BB10’	’Frank’	’Gaussian’	’BB10’	’t’
4	’Frank’	’t’	’t’	’Gaussian’	’Frank’	’BB10’	’Gaussian’	’Frank’	’Clayton’	’Frank’	’t’	’Frank’	’Frank’	’Frank’	’t’	’t’	’t’
5	’t’	’t’	’t’	’Gumbel’	’t’	’BB10’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Clayton’	’Frank’	’Frank’	’Frank’	’BB10’	’Frank’	’BB7’
6	’Gaussian’	’Frank’	’Frank’	’t’	’Gaussian’	’Gaussian’	’Clayton’	’Frank’	’Gaussian’	’Gaussian’	’Clayton’	’BB10’	’BB10’	’BB10’	’t’	’t’	’t’
7	’Frank’	’Frank’	’BB10’	’Frank’	’Galambos’	’t’	’Gaussian’	’Gaussian’	’t’	’Gaussian’	’Frank’	’Gumbel’	’BB10’	’Gaussian’	’t’	’BB1’	’t’
8	’Gaussian’	’BB10’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Frank’	’BB10’	’Gaussian’	’Frank’	’t’	’Frank’	’t’	’t’	’Gaussian’	’BB1’	’t’
9	’Gaussian’	’Frank’	’Frank’	’Clayton’	’Frank’	’Clayton’	’Gaussian’	’Frank’	’Clayton’	’Gaussian’	’Gaussian’	’Gaussian’	’t’	’BB10’	’t’	’t’	’t’
10	’Gaussian’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’BB10’	’t’	’t’	’Gaussian’	’Gaussian’	’t’	’t’	’t’	’Frank’	’t’	’BB7’
11	’Frank’	’Clayton’	’Frank’	’Frank’	’Frank’	’BB10’	’Gaussian’	’Frank’	’t’	’Gaussian’	’Frank’	’t’	’Frank’	’BB10’	’Frank’	’t’	’t’
12	’Galambos’	’Frank’	’Frank’	’t’	’Gaussian’	’Frank’	’BB10’	’t’	’t’	’BB10’	’Frank’	’Frank’	’BB10’	’Frank’	’Frank’	’Frank’	’BB7’
13	’Gaussian’	’Frank’	’Clayton’	’BB1’	’Frank’	’BB10’	’Gumbel’	’t’	’Frank’	’Gaussian’	’Frank’	’t’	’t’	’Frank’	’Gaussian’	’Frank’	’t’
14	’t’	’Frank’	’Frank’	’Frank’	’Gaussian’	’t’	’Gaussian’	’Gaussian’	’Gaussian’	’t’	’Gaussian’	’BB10’	’Frank’	’t’	’BB10’	’t’	’t’
15	’Gaussian’	’Gaussian’	’Clayton’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’t’	’Frank’	’t’	’Gaussian’	’Frank’	’Frank’	’Frank’	’BB10’	’BB7’
16	’Gaussian’	’Gumbel’	’Gaussian’	’Clayton’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Galambos’	’BB10’	’Frank’	’t’	’Frank’	’Frank’	’BB10’	’BB10’	’BB10’
17	’Frank’	’Frank’	’Frank’	’Gaussian’	’BB1’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’t’	’Gaussian’	’Gaussian’	’t’	’t’	’t’	’BB3’
18	’Frank’	’Gaussian’	’BB1’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’t’	’Frank’	’Frank’	’Gaussian’	’t’	’t’	’BB7’
19	’Gaussian’	’Frank’	’t’	’Gaussian’	’t’	’t’	’Gaussian’	’Frank’	’Frank’	’t’	’BB1’	’Frank’	’t’	’t’	’t’	’t’	’BB10’
20	’Galambos’	’t’	’BB10’	’t’	’Frank’	’BB10’	’t’	’t’	’Frank’	’Frank’	’Frank’	’t’	’Gaussian’	’Frank’	’BB10’	’t’	’t’
21	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’t’	’t’	’Gaussian’	’Gumbel’	’Frank’	’t’	’BB10’	’BB7’
22	’Frank’	’BB10’	’Frank’	’BB10’	’t’	’Gumbel’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’t’	’Frank’	’Frank’	’Frank’	’t’	’BB3’
23	’Gaussian’	’Gaussian’	’Galambos’	’Gaussian’	’Frank’	’Frank’	’BB10’	’Frank’	’Gaussian’	’Frank’	’BB1’	’t’	’Frank’	’t’	’Frank’	’t’	’t’
24	’Gaussian’	’Gaussian’	’Gaussian’	’BB1’	’Frank’	’t’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’t’	’Gumbel’	’Gaussian’	’t’	’t’	’t’
25	’Frank’	’Frank’	’BB1’	’Clayton’	’Frank’	’Frank’	’Frank’	’t’	’t’	’t’	’Frank’	’Frank’	’B5’	’Gumbel’	’Frank’	’BB10’	’t’
26	’Gaussian’	’Frank’	’Galambos’	’Gaussian’	’t’	’Gaussian’	’Frank’	’BB10’	’t’	’Gaussian’	’BB10’	’t’	’Frank’	’t’	’Frank’	’BB10’	’t’
27	’Gaussian’	’Galambos’	’t’	’Clayton’	’BB10’	’Frank’	’Clayton’	’t’	’Gaussian’	’t’	’Gaussian’	’Gaussian’	’t’	’Frank’	’Frank’	’BB1’	’t’
28	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Clayton’	’Frank’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’BB10’	’Frank’	’t’	’Frank’	’t’	’t’
29	’Frank’	’BB10’	’Clayton’	’t’	’BB10’	’Gaussian’	’BB10’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’BB10’	’BB10’	’Gaussian’	’t’	’Frank’	’BB10’
30	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’BB10’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’t’	’t’	’t’
31	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Gumbel’	’t’	’Frank’	’t’	’BB10’	’Frank’	’t’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’t’
32	’BB10’	’BB10’	’Gaussian’	’Frank’	’Gaussian’	’t’	’Frank’	’BB10’	’Gumbel’	’Frank’	’t’	’BB10’	’t’	’t’	’Frank’	’BB1’	’BB7’
33	’Gaussian’	’Frank’	’BB10’	’BB10’	’Frank’	’Gaussian’	’t’	’Frank’	’Clayton’	’t’	’t’	’Frank’	’BB10’	’Frank’	’t’	’t’	’t’
34	’Gaussian’	’Frank’	’t’	’Frank’	’t’	’Clayton’	’Gaussian’	’Frank’	’Frank’	’t’	’t’	’Frank’	’Gumbel’	’Gaussian’	’Frank’	’Frank’	’BB7’
35	’t’	’Frank’	’Gumbel’	’Clayton’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’BB10’	’Gaussian’	’Gumbel’	’Gaussian’	’Gumbel’	’Frank’	’t’	’BB7’
36	’Gaussian’	’Clayton’	’t’	’BB1’	’Frank’	’Gaussian’	’BB10’	’BB10’	’Frank’	’t’	’BB10’	’Frank’	’BB10’	’Frank’	’t’	’BB1’	’t’
37	’Gumbel’	’Clayton’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’t’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Gumbel’	’Gaussian’	’Frank’	’BB10’	’t’
38	’Gaussian’	’Frank’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’BB10’	’Gaussian’	’Gaussian’	’BB10’	’Frank’	’Frank’	’BB10’	’Frank’	’t’	’t’
39	’Frank’	’Gaussian’	’Clayton’	’Clayton’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Frank’	’BB10’	’Gaussian’	’t’	’t’	’t’	’t’	’t’
40	’BB10’	’Gaussian’	’Gaussian’	’Gaussian’	’t’	’Frank’	’Frank’	’Gaussian’	’t’	’Clayton’	’Frank’	’BB10’	’Gumbel’	’Frank’	’Frank’	’t’	’BB7’
41	’Gaussian’	’Gumbel’	’t’	’t’	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’BB10’	’BB7’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’t’	’t’
42	’Frank’	’B5’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Gumbel’	’Frank’	’Frank’	’Frank’	’t’	’Clayton’	’Gaussian’	’Gaussian’	’t’	’Frank’	’t’
43	’Frank’	’Gaussian’	’Frank’	’t’	’BB10’	’Frank’	’Frank’	’t’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’BB10’	’Frank’	’t’	’t’
44	’Gaussian’	’Frank’	’t’	’Frank’	’Gaussian’	’BB10’	’BB10’	’Frank’	’t’	’t’	’BB10’	’Frank’	’Frank’	’t’	’Frank’	’t’	’t’
45	’Frank’	’BB10’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Gumbel’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’t’	’t’
46	’BB10’	’t’	’Gaussian’	’Gaussian’	’t’	’Frank’	’Gaussian’	’Gumbel’	’Clayton’	’t’	’Gumbel’	’Gaussian’	’t’	’Gaussian’	’Gaussian’	’t’	’t’
47	’Frank’	’BB10’	’t’	’t’	’t’	’Frank’	’Clayton’	’Frank’	’t’	’t’	’Frank’	’Frank’	’Frank’	’BB10’	’Frank’	’t’	’t’
48	’Frank’	’Gaussian’	’Galambos’	’t’	’t’	’t’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Gumbel’	’Gaussian’	’Frank’	’t’	’t’	’BB1’	’t’
49	’t’	’Clayton’	’Gaussian’	’Galambos’	’Gaussian’	’Gaussian’	’t’	’t’	’Clayton’	’t’	’Gaussian’	’BB10’	’t’	’Frank’	’Frank’	’t’	’t’
50	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’t’	’Gaussian’	’Frank’	’Clayton’	’Frank’	’Frank’	’Frank’	’t’	’t’
51	’Frank’	’Gaussian’	’t’	’t’	’Gaussian’	’Gaussian’	’Frank’	’t’	’BB10’	’Gumbel’	’Clayton’	’Gaussian’	’Frank’	’t’	’t’	’t’	’t’
52	’t’	’Frank’	’Frank’	’Clayton’	’Frank’	’Gaussian’	’Frank’	’Clayton’	’Clayton’	’Frank’	’Frank’	’t’	’t’	’Frank’	’Gaussian’	’t’	’BB1’
53	’Gaussian’	’Gaussian’	’Frank’	’BB1’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Gumbel’	’Frank’	’t’	’t’	’t’	’t’	’t’	’BB7’
54	’t’	’t’	’t’	’Gaussian’	’t’	’BB10’	’Frank’	’Frank’	’Gaussian’	’Gumbel’	’BB10’	’t’	’t’	’Frank’	’Frank’	’t’	’t’
55	’Frank’	’Frank’	’Frank’	’BB1’	’Gaussian’	’Gaussian’	’Frank’	’t’	’t’	’Gaussian’	’t’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’BB7’
56	’Gaussian’	’t’	’Frank’	’t’	’Frank’	’Gaussian’	’BB1’	’Gaussian’	’t’	’Frank’	’Frank’	’Frank’	’Frank’	’BB10’	’Frank’	’t’	’BB7’
57	’Frank’	’Gaussian’	’Gaussian’	’BB1’	’t’	’Gaussian’	’t’	’t’	’Gaussian’	’t’	’Gaussian’	’Frank’	’t’	’Frank’	’BB10’	’BB10’	’t’
58	’t’	’Galambos’	’Frank’	’Gaussian’	’t’	’Frank’	’t’	’Galambos’	’t’	’BB4’	’Gumbel’	’Gaussian’	’Gaussian’	’Frank’	’t’	’t’	’t’
59	’Clayton’	’Frank’	’Clayton’	’Clayton’	’Gaussian’	’Clayton’	’Frank’	’Frank’	’Frank’	’Clayton’	’Frank’	’t’	’t’	’Frank’	’Frank’	’Frank’	’t’
60	’BB10’	’BB1’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’Frank’	’t’	’t’	’t’	’t’	’BB7’
61	’BB1’	’t’	’Frank’	’Gaussian’	’Galambos’	’t’	’Gumbel’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’t’	’Frank’	’BB7’
62	’Gaussian’	’Gaussian’	’Frank’	’t’	’t’	’Frank’	’t’	’Gaussian’	’t’	’Frank’	’Gaussian’	’t’	’Frank’	’Frank’	’t’	’Frank’	’BB7’
63	’Frank’	’Gaussian’	’Gaussian’	’t’	’Frank’	’BB7’	’t’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’t’
64	’t’	’t’	’Frank’	’BB1’	’t’	’Frank’	’Gumbel’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’t’	’t’
65	’Frank’	’Frank’	’Gaussian’	’BB10’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’B5’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’t’	’t’	’BB7’
66	’Gaussian’	’BB10’	’Gaussian’	’Gaussian’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’t’	’Frank’	’t’	’BB10’	’Frank’	’t’	’BB10’	’t’	’t’
67	’t’	’Frank’	’Gaussian’	’Clayton’	’Frank’	’B5’	’BB10’	’Gaussian’	’Frank’	’t’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’BB7’
68	’Gaussian’	’Clayton’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’Gaussian’	’Gumbel’	’Clayton’	’t’	’Frank’	’BB10’	’t’	’Clayton’	’t’	’t’	’BB7’
69	’Frank’	’Gaussian’	’BB10’	’t’	’Clayton’	’t’	’BB10’	’Frank’	’BB10’	’Frank’	’Frank’	’t’	’Frank’	’t’	’Frank’	’t’	’BB7’
70	’Frank’	’Gaussian’	’Clayton’	’Frank’	’Clayton’	’Frank’	’Frank’	’Frank’	’BB1’	’Frank’	’t’	’Gaussian’	’t’	’Frank’	’BB10’	’Frank’	’t’
71	’Frank’	’Frank’	’Frank’	’Gaussian’	’Galambos’	’Gaussian’	’Clayton’	’Frank’	’Frank’	’t’	’t’	’Clayton’	’BB10’	’t’	’t’	’t’	’BB7’
72	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’Gaussian’	’t’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Frank’	’t’	’Frank’	’t’	’BB10’	’t’
73	’Gaussian’	’Frank’	’Clayton’	’Galambos’	’Gumbel’	’Gaussian’	’t’	’Clayton’	’t’	’Clayton’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’BB1’	’BB10’	’t’
74	’t’	’BB10’	’Gaussian’	’Clayton’	’Galambos’	’Frank’	’t’	’Gaussian’	’Frank’	’Frank’	’t’	’Frank’	’Frank’	’Frank’	’t’	’t’	’t’
75	’t’	’Gaussian’	’t’	’Gaussian’	’Frank’	’t’	’BB1’	’Frank’	’Frank’	’Frank’	’Gaussian’	’BB1’	’Frank’	’t’	’t’	’t’	’BB3’
76	’BB1’	’Gaussian’	’Gaussian’	’BB7’	’Gaussian’	’BB10’	’Frank’	’Frank’	’Gaussian’	’BB10’	’Frank’	’t’	’t’	’t’	’t’	’BB1’	’t’
77	’Galambos’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Gaussian’	’Frank’	’Gaussian’	’t’	’t’	’Gaussian’	’t’	’t’	’t’	’BB1’	’t’
78	’Gaussian’	’Gaussian’	’B5’	’t’	’Clayton’	’Gaussian’	’t’	’Gaussian’	’t’	’B5’	’Frank’	’Frank’	’t’	’t’	’BB10’	’t’	’BB7’
79	’Gaussian’	’t’	’Frank’	’t’	’t’	’Frank’	’BB10’	’Galambos’	’Frank’	’BB10’	’Frank’	’Frank’	’Frank’	’t’	’t’	’t’	’BB7’
80	’t’	’Galambos’	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’Frank’	’BB10’	’Gaussian’	’Frank’	’t’	’BB10’
81	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’t’	’Clayton’	’Frank’	’Frank’	’t’	’Gumbel’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’t’	’BB7’
82	’Gaussian’	’Clayton’	’Frank’	’Gaussian’	’Frank’	’t’	’Gaussian’	’Clayton’	’Frank’	’Gaussian’	’Frank’	’Clayton’	’Frank’	’Frank’	’t’	’t’	’BB7’
83	’Gaussian’	’t’	’Gaussian’	’Gaussian’	’t’	’t’	’Gumbel’	’Gaussian’	’t’	’t’	’Frank’	’t’	’Gaussian’	’BB10’	’t’	’t’	’BB7’
84	17	’Frank’	’Frank’	’Gumbel’	’Frank’	’t’	’Gaussian’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’t’	’t’	’Frank’	’BB10’
85	16	16	’Frank’	’t’	’Frank’	’Gaussian’	’Gaussian’	’BB10’	’t’	’Gaussian’	’B5’	’BB10’	’t’	’Frank’	’BB10’	’t’	’t’
86	14	15	15	’t’	’Frank’	’t’	’Clayton’	’BB7’	’Gaussian’	’BB10’	’Frank’	’t’	’Frank’	’t’	’Frank’	’Frank’	’t’
87	13	11	14	14	’Frank’	’Frank’	’Frank’	’Gaussian’	’Frank’	’BB10’	’BB10’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’t’
88	12	10	13	11	13	’BB10’	’t’	’t’	’Gaussian’	’BB10’	’Frank’	’Frank’	’Frank’	’t’	’t’	’t’	’t’
89	11	1	12	10	11	12	’t’	’Gaussian’	’Frank’	’Clayton’	’Frank’	’BB10’	’t’	’Frank’	’Frank’	’t’	’t’
90	10	4	11	1	10	11	11	’Clayton’	’BB10’	’Gaussian’	’BB10’	’t’	’BB10’	’t’	’Frank’	’t’	’BB7’
91	1	8	10	4	1	10	9	10	’Gaussian’	’t’	’Gaussian’	’t’	’Clayton’	’Gaussian’	’t’	’BB10’	’Gumbel’
92	4	7	1	8	4	1	1	9	9	’Gaussian’	’Frank’	’Frank’	’t’	’BB10’	’t’	’BB10’	’t’
93	8	6	4	7	8	4	4	1	1	8	’Frank’	’Galambos’	’Frank’	’BB10’	’Frank’	’BB10’	’BB10’
94	7	2	8	6	7	8	5	4	4	1	7	’t’	’t’	’Frank’	’Frank’	’t’	’t’
95	9	14	7	2	6	7	3	5	8	4	1	6	’Gaussian’	’Frank’	’t’	’t’	’BB3’
96	5	13	9	5	2	6	8	3	7	5	4	1	5	’Frank’	’BB10’	’t’	’t’
97	3	12	5	3	5	2	10	8	6	3	5	4	1	4	’t’	’t’	’BB7’
98	2	5	3	9	3	5	7	2	2	2	3	5	4	1	3	’t’	’t’
99	6	3	2	13	9	3	6	7	5	7	2	3	2	2	1	2	’t’
100	15	9	6	12	12	9	2	6	3	6	6	2	3	3	2	1	1

This is the full 30-dimensional empirical result of Kendall’s $\hat{\tau}$ and Spearman $\hat{\rho}_s$ in VBDA2 in the ICAPM-EGARCH(1,1,1)-Mixture-R-vine model. The result is represented in matrix form in which the diagonal matrix presents the stock name, the upper triangular matrix presents Spearman $\hat{\rho}_s$ and the lower triangular matrix presents Kendall’s $\hat{\tau}$ as follows.

Row/Column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	’HDB’	0.01	0.04	-0.01	0.02	0.04	0.06	0.00	0.03	0.02	0.00	0.03	0.04	0.07	0.04
2	0.01	’BMY’	0.03	0.04	0.01	0.02	0.04	0.03	0.05	-0.01	0.30	0.13	0.02	0.01	0.00
3	0.03	0.02	’AMT’	0.18	0.06	0.02	-0.02	0.03	-0.02	0.02	0.04	0.10	0.05	0.06	0.02
4	-0.01	0.02	0.12	’SBUX’	0.02	0.02	0.03	0.02	0.06	0.01	0.02	0.01	0.10	0.09	0.10
5	0.01	0.00	0.04	0.01	’BLK’	0.04	0.01	0.01	-0.02	0.03	0.00	0.22	0.05	0.01	0.00
6	0.03	0.01	0.02	0.01	0.03	’SCHW’	0.01	-0.01	0.14	0.34	0.05	0.12	0.02	0.02	0.07
7	0.04	0.02	-0.01	0.02	0.01	0.01	’UNP’	0.00	0.00	0.01	-0.01	-0.01	0.01	0.06	0.00
8	0.00	0.02	0.02	0.01	0.01	-0.01	0.00	’BHP’	-0.02	0.03	0.01	0.01	0.00	0.03	0.03
9	0.02	0.04	-0.01	0.04	-0.01	0.09	0.00	-0.01	’AXP’	0.03	0.06	0.00	-0.02	0.28	0.15
10	0.02	-0.01	0.01	0.01	0.02	0.23	0.01	0.02	0.02	’GS’	0.02	0.06	-0.02	-0.01	0.02
11	0.00	0.20	0.03	0.02	0.00	0.03	-0.01	0.00	0.04	0.02	’RTX’	-0.03	0.31	0.01	0.03
12	0.02	0.08	0.06	0.01	0.15	0.08	-0.01	0.01	0.00	0.04	-0.02	’AMD’	0.30	0.04	0.03
13	0.02	0.01	0.03	0.06	0.03	0.02	0.01	0.00	-0.01	-0.02	0.21	0.20	’BA’	0.03	0.04
14	0.05	0.01	0.04	0.06	0.01	0.01	0.04	0.02	0.19	-0.01	0.01	0.03	0.02	’AMAT’	0.09
15	0.03	0.01	0.01	0.07	0.00	0.05	0.00	0.02	0.10	0.02	0.02	0.02	0.02	0.06	’AMGN’
16	0.04	0.02	-0.06	0.07	0.05	0.09	0.74	0.01	0.05	0.03	0.04	0.05	0.01	0.00	0.08
17	0.03	0.05	0.05	0.00	0.03	0.06	0.01	0.01	0.06	0.04	0.03	0.02	0.01	0.03	0.08
18	0.05	0.06	-0.01	0.09	0.04	-0.01	0.03	0.00	0.06	0.12	0.09	0.04	0.04	0.05	0.05
19	0.05	0.04	0.01	-0.03	0.01	0.04	0.01	0.01	0.05	0.03	0.04	0.04	0.02	0.01	0.08
20	0.02	0.07	0.03	0.08	0.08	0.05	0.06	0.00	0.04	0.09	0.03	0.01	0.05	0.03	0.14
21	0.00	0.03	0.12	-0.01	0.05	0.13	0.01	0.11	0.00	0.02	0.07	0.03	0.04	0.03	0.03
22	0.03	0.01	0.01	0.17	0.08	0.07	0.14	0.03	0.05	0.05	0.07	0.08	0.13	0.01	0.08
23	0.01	0.02	0.01	0.02	0.10	0.00	0.20	0.01	0.10	0.00	0.05	0.05	0.07	0.08	0.01
24	0.09	0.05	0.03	0.02	0.08	0.09	0.10	0.27	0.08	0.10	0.07	0.04	0.06	0.09	0.01
25	0.07	0.03	0.07	0.01	0.08	0.06	0.11	0.09	0.10	0.09	0.18	0.04	0.10	0.09	0.06
26	0.08	0.11	0.09	0.07	0.15	0.13	0.09	0.07	0.07	0.15	0.13	0.05	0.07	0.11	0.03
27	0.06	0.10	0.01	0.11	0.13	0.14	0.15	0.13	0.16	0.22	0.15	0.06	0.10	0.14	0.05
28	0.12	0.04	0.15	0.17	0.18	0.21	0.21	0.13	0.22	0.19	0.30	0.07	0.18	0.20	0.10
29	0.51	0.21	0.24	0.30	0.34	0.17	0.24	0.70	0.31	0.32	0.31	0.20	0.32	0.19	0.23
30	0.26	0.31	0.28	0.30	0.47	0.42	0.45	0.35	0.43	0.36	0.34	0.33	0.31	0.39	0.34

Row/Column	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0.06	0.05	0.07	0.08	0.02	0.00	0.04	0.01	0.14	0.10	0.12	0.09	0.18	0.70	0.39
2	0.02	0.07	0.08	0.06	0.11	0.04	0.02	0.02	0.07	0.05	0.17	0.15	0.06	0.32	0.45
3	-0.09	0.07	-0.02	0.01	0.04	0.18	0.01	0.02	0.04	0.11	0.13	0.02	0.23	0.35	0.41
4	0.11	0.00	0.14	-0.04	0.12	-0.02	0.25	0.03	0.04	0.02	0.10	0.16	0.26	0.44	0.44
5	0.07	0.05	0.06	0.01	0.12	0.07	0.11	0.15	0.12	0.12	0.22	0.20	0.27	0.49	0.64
6	0.13	0.08	-0.02	0.07	0.07	0.19	0.11	0.00	0.13	0.09	0.19	0.21	0.30	0.26	0.59
7	0.90	0.02	0.05	0.02	0.09	0.01	0.20	0.30	0.15	0.16	0.13	0.22	0.30	0.36	0.61
8	0.01	0.01	-0.01	0.01	0.01	0.16	0.04	0.02	0.39	0.14	0.10	0.19	0.20	0.88	0.51
9	0.07	0.08	0.08	0.07	0.06	0.01	0.07	0.15	0.12	0.15	0.10	0.25	0.32	0.46	0.60
10	0.05	0.07	0.18	0.05	0.13	0.03	0.07	0.00	0.15	0.13	0.22	0.33	0.28	0.46	0.51
11	0.07	0.04	0.14	0.05	0.05	0.10	0.10	0.07	0.10	0.27	0.20	0.23	0.44	0.45	0.48
12	0.07	0.02	0.07	0.06	0.02	0.05	0.12	0.07	0.06	0.06	0.08	0.08	0.10	0.30	0.48
13	0.01	0.01	0.05	0.03	0.07	0.07	0.19	0.10	0.09	0.15	0.10	0.15	0.27	0.46	0.44
14	0.00	0.05	0.07	0.02	0.05	0.05	0.02	0.12	0.14	0.14	0.16	0.20	0.30	0.28	0.55
15	0.12	0.12	0.07	0.12	0.21	0.05	0.11	0.02	0.02	0.09	0.04	0.08	0.15	0.33	0.49
16	’ISRG’	0.01	0.05	0.02	0.01	0.07	0.07	0.02	0.58	0.13	0.05	0.15	0.06	0.23	0.46
17	0.01	’IBM’	0.02	0.09	0.09	0.03	0.10	0.14	0.05	0.14	0.11	0.03	0.30	0.44	0.56
18	0.03	0.01	’SNY’	0.02	0.04	0.06	0.15	0.17	0.03	0.02	0.11	0.62	0.24	0.24	0.55
19	0.01	0.06	0.01	’TGT’	0.06	-0.01	0.06	0.04	0.06	0.15	0.10	0.18	0.21	0.28	0.46
20	0.00	0.06	0.03	0.04	’TD’	0.06	0.06	0.22	0.18	0.08	0.18	0.12	0.38	0.32	0.57
21	0.05	0.02	0.04	0.00	0.04	’TTE’	0.01	0.08	0.19	0.05	0.24	0.11	0.48	0.27	0.57
22	0.04	0.06	0.10	0.04	0.04	0.01	’EL’	0.20	0.08	0.10	0.15	0.18	0.20	0.32	0.55
23	0.01	0.09	0.11	0.03	0.15	0.05	0.13	’CVS’	0.14	0.05	0.08	0.14	0.11	0.35	0.47
24	0.41	0.03	0.02	0.04	0.12	0.13	0.06	0.09	’GE’	0.12	0.14	0.30	0.23	0.39	0.51
25	0.09	0.10	0.01	0.10	0.05	0.03	0.07	0.04	0.08	’DEO’	0.06	0.07	0.10	0.31	0.54
26	0.03	0.08	0.07	0.07	0.12	0.16	0.10	0.05	0.09	0.04	’SPGI’	0.11	0.19	0.07	0.52
27	0.10	0.02	0.45	0.12	0.08	0.07	0.12	0.10	0.21	0.05	0.07	’RIO’	0.15	0.55	0.52
28	0.04	0.21	0.16	0.14	0.26	0.33	0.13	0.07	0.16	0.07	0.13	0.10	’JBHT’	0.20	0.61
29	0.15	0.30	0.16	0.19	0.22	0.18	0.21	0.24	0.27	0.21	0.05	0.39	0.14	’DE’	0.77
30	0.32	0.39	0.39	0.32	0.42	0.41	0.39	0.32	0.36	0.38	0.37	0.37	0.44	0.58	’CAT’

The result of the 30-dimensional stock index experiment, $\hat{\mathcal{M}}_{30-Dim}^{Mixed-R-vine}$ and $\hat{\mathcal{T}}_{30-Dim}^{Mixed-R-vine}$ of VBDA2 in the ICAPM-EGARCH(1,1,1)-Mixture-R-vine model is presented in the lower and upper triangular matrix, respectively, as follows.

Row\Column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	30	’Frank’	’Gaussian’	’t’	’t’	’Gaussian’	’Frank’	’t’	’Clayton’	’Gaussian’	’t’	’Frank’	’Gaussian’	’Frank’	’Frank’
2	27	29	’Clayton’	’Frank’	’B5’	’Frank’	’Clayton’	’Frank’	’Frank’	’Gaussian’	’Frank’	’Frank’	’BB10’	’B5’	’BB10’
3	26	27	28	’Frank’	’Gaussian’	’Gaussian’	’Gaussian’	’Clayton’	’Gaussian’	’Gaussian’	’Clayton’	’BB7’	’BB10’	’Frank’	’Galambos’
4	24	26	27	27	’B5’	’Frank’	’BB10’	’Gumbel’	’Gaussian’	’Clayton’	’Frank’	’BB10’	’Frank’	’t’	’BB1’
5	23	24	26	25	26	’BB10’	’B5’	’Frank’	’Frank’	’BB7’	’Gaussian’	’Frank’	’Frank’	’BB10’	’Galambos’
6	22	23	24	23	25	25	’Clayton’	’Gaussian’	’BB10’	’t’	’Gaussian’	’t’	’Gaussian’	’Frank’	’Frank’
7	21	22	23	19	23	24	24	’Gaussian’	’B5’	’Frank’	’t’	’t’	’Clayton’	’Frank’	’Gaussian’
8	20	21	22	18	19	23	23	23	’Gaussian’	’Clayton’	’Frank’	’Gaussian’	’Galambos’	’Clayton’	’Frank’
9	17	20	21	15	18	22	19	22	22	’Gumbel’	’t’	’BB10’	’Gaussian’	’t’	’Frank’
10	16	17	20	13	15	21	18	21	19	21	’Gaussian’	’Frank’	’Frank’	’Frank’	’BB10’
11	13	16	17	26	13	20	15	20	18	19	20	’Gaussian’	’t’	’Galambos’	’Clayton’
12	14	13	16	24	21	17	13	19	15	18	19	19	’Frank’	’t’	’Gaussian’
13	11	14	13	22	20	16	21	17	13	15	18	17	18	’t’	’t’
14	25	11	14	17	12	13	20	16	21	13	15	16	17	17	’t’
15	19	25	11	16	9	14	12	18	20	17	13	13	16	15	16
16	18	19	25	14	8	11	9	15	12	16	17	14	13	13	15
17	15	18	19	11	10	12	8	14	9	14	16	11	14	12	13
18	10	15	18	12	6	9	10	12	8	11	14	12	11	9	12
19	6	10	15	10	2	19	6	9	10	12	11	9	12	8	9
20	2	6	10	6	16	8	2	8	6	10	12	8	9	10	8
21	1	2	6	2	1	7	16	10	2	6	10	18	8	6	10
22	3	1	2	9	3	10	1	6	16	2	6	15	7	2	6
23	12	3	1	1	17	6	3	2	1	9	2	10	10	16	2
24	4	12	3	3	4	5	17	13	3	1	9	6	6	1	1
25	28	4	12	4	14	4	4	1	17	3	1	2	5	3	3
26	9	28	4	21	24	3	14	3	4	4	3	1	4	14	4
27	8	9	8	7	11	1	22	11	14	7	4	3	3	4	11
28	5	8	9	20	7	2	5	7	11	5	7	7	1	5	7
29	7	5	5	5	22	15	7	4	7	20	5	4	2	7	5
30	29	7	7	8	5	18	11	5	5	8	8	5	15	11	14

Row\Column	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	’BB1’	’t’	’BB1’	’Gaussian’	’Gaussian’	’BB10’	’Gaussian’	’Galambos’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’Gaussian’	’BB3’	’t’
2	’Clayton’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’B5’	’t’	’t’	’Gumbel’	’t’	’Frank’	’BB10’	’t’	’t’
3	’t’	’Gaussian’	’t’	’BB10’	’B5’	’t’	’Frank’	’t’	’Gaussian’	’BB4’	’t’	’BB10’	’Frank’	’t’	’t’
4	’BB10’	’t’	’Frank’	’Gaussian’	’Frank’	’Gaussian’	’Frank’	’Clayton’	’BB10’	’BB10’	’Frank’	’BB10’	’t’	’t’	’t’
5	’Frank’	’Gumbel’	’Gumbel’	’BB10’	’Frank’	’t’	’Frank’	’t’	’t’	’t’	’Frank’	’t’	’t’	’t’	’BB7’
6	’t’	’Gumbel’	’Frank’	’Frank’	’Clayton’	’Frank’	’Frank’	’t’	’t’	’Frank’	’Frank’	’t’	’t’	’Frank’	’BB7’
7	’BB1’	’t’	’Clayton’	’Frank’	’Frank’	’BB10’	’Frank’	’t’	’Frank’	’BB10’	’t’	’t’	’Frank’	’t’	’BB7’
8	’t’	’Gaussian’	’Gaussian’	’Galambos’	’Gaussian’	’Gaussian’	’Gaussian’	’Clayton’	’BB3’	’Gaussian’	’Frank’	’t’	’t’	’t’	’t’
9	’Frank’	’Frank’	’Gaussian’	’t’	’t’	’BB10’	’t’	’BB10’	’BB10’	’Frank’	’BB1’	’Frank’	’t’	’t’	’t’
10	’Frank’	’t’	’Frank’	’Frank’	’BB1’	’B5’	’Gaussian’	’Frank’	’Frank’	’t’	’t’	’Frank’	’t’	’t’	’BB7’
11	’t’	’t’	’t’	’t’	’t’	’Frank’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’t’	’t’	’t’	’t’
12	’Clayton’	’Frank’	’Gumbel’	’t’	’t’	’Gaussian’	’BB10’	’t’	’Clayton’	’Frank’	’t’	’Frank’	’t’	’t’	’BB7’
13	’Frank’	’BB10’	’t’	’Gaussian’	’t’	’Gumbel’	’t’	’Frank’	’t’	’t’	’BB1’	’t’	’t’	’t’	’t’
14	’BB10’	’Frank’	’t’	’Gumbel’	’t’	’Clayton’	’t’	’Frank’	’BB10’	’Frank’	’Frank’	’Frank’	’t’	’Frank’	’BB7’
15	’Frank’	’Gaussian’	’Gaussian’	’BB10’	’Frank’	’Gaussian’	’t’	’Galambos’	’t’	’Frank’	’t’	’t’	’t’	’t’	’t’
16	15	’t’	’t’	’Clayton’	’Frank’	’Frank’	’Frank’	’Gumbel’	’BB3’	’Frank’	’BB10’	’Frank’	’BB10’	’t’	’t’
17	13	14	’Clayton’	’Frank’	’Frank’	’BB10’	’Frank’	’Frank’	’Frank’	’Frank’	’Frank’	’BB10’	’Frank’	’t’	’t’
18	14	13	13	’Galambos’	’Gaussian’	’Frank’	’t’	’Gaussian’	’Frank’	’Clayton’	’BB10’	’BB1’	’t’	’t’	’BB7’
19	11	12	12	12	’t’	’Gaussian’	’t’	’Clayton’	’Frank’	’Frank’	’t’	’Frank’	’t’	’t’	’t’
20	12	9	9	11	11	’Clayton’	’t’	’t’	’Frank’	’t’	’Frank’	’t’	’t’	’t’	’t’
21	9	8	8	10	9	10	’BB10’	’Gaussian’	’t’	’t’	’t’	’Frank’	’t’	’t’	’BB7’
22	8	10	10	6	8	9	9	’BB10’	’t’	’Gaussian’	’Frank’	’t’	’t’	’t’	’t’
23	7	6	6	2	10	8	6	8	’t’	’Frank’	’t’	’t’	’Gumbel’	’t’	’t’
24	10	2	2	1	6	7	2	6	7	’Gumbel’	’Frank’	’t’	’t’	’t’	’t’
25	6	1	1	3	2	2	1	2	6	6	’Gaussian’	’t’	’t’	’t’	’BB7’
26	5	3	3	4	1	1	3	1	2	2	5	’t’	’Frank’	’BB10’	’BB7’
27	4	4	4	9	3	3	4	3	1	1	2	4	’BB1’	’t’	’BB7’
28	3	11	5	7	4	4	8	4	3	3	1	2	3	’t’	’BB7’
29	1	7	7	5	5	5	7	7	4	4	3	1	2	2	’t’
30	2	5	11	8	7	6	5	5	5	5	4	3	1	1	1

B.3 Parametric Copula Families and Some Properties

This appendix presents the dependence structure of all copula families used in this study including one- and two-parameter(s) copula families. The elliptical copulae are the Gaussian copula and Student-*t* copula. The important tail dependence of copula families in this study are shown in Table B.3.

Table B.3: Tail Dependence of Parametric Copula Families

Copula	K_L or λ_L	K_U **or** λ_U
One-parameter		
Gaussian	$K_L = 2/(1+\rho)$	$K_U = K_L$
Gumbel (extreme value)	$K_L = 2^{1/\theta}$	$K_U = 2 - 2^{1/\theta}$
Clayton	$K_L = 2^{-1/\theta}$	$K_U = 2$
Frank	$K_L = 2$	$K_U = 2$
Joe/B5	$K_L = 2$	$\lambda_U = 2 - 2^{1/\theta}$
Galambos (extreme value)	$K_L = 2 - 2^{-1/\theta}$	$\lambda_U = 2^{-1/\theta}$
Two-parameters		
Student-*t*	$\lambda_L = \lambda_U$	$\lambda_U = 2T_{\nu+1}(-\sqrt{\frac{(\nu+1)(1-\rho)}{1+\rho}})$
BB1	$\lambda_L = 2^{-1/\theta\delta}$	$\lambda_U = 2 - 2^{1/\delta}$
BB3	$\lambda_L = 1$	$\lambda_U = 2 - 2^{1/\theta}$
BB4	$\lambda_L = (2 - 2^{-1/\delta})^{-1/\theta}$	$\lambda_U = 2^{-1/\delta}$
BB5 (extreme value)	$K_L = 2 - \lambda_U$	$\lambda_U = 2 - (2 - 2^{-1/\delta})^{1/\theta}$
BB7	$\lambda_L = 2^{-1/\delta}$	$\lambda = 2 - 2^{1/\theta}$
BB10	$K_L = 2$	$K_U = 2$

A rank-based measurement of the dependence measure of the copula families can be computed either from a closed-form solution (if available) or numerical approximation via a Monte-Carlo simulation through two-variable integrals as in equation B.3.1 for Kendall's τ and equation B.3.2 for Spearman ρ_s, respectively.

$$\hat{\tau}(C) = 1 - 4\int_0^1 \int_0^1 C_{1|2}(u|v)C_{2|1}(v|u)dudv \qquad \text{(B.3.1)}$$

$$\hat{\rho_s}(C) = 3 - 12 \int_0^1 \int_0^1 u C_{2|1}(v|u) du dv \tag{B.3.2}$$

Note that a closed-form Kendall's $\hat{\tau}$ and Spearman $\hat{\rho_s}$ of Gaussian copula, *t*-copula, Gumbel copula, Frank copula and Clayton copula are used. Otherwise, the study used Monte-Carlo approximation.

B.4 Independence of the Standardised Residuals

This appendix discusses the Ljung-Box independence testing of the marginal model residuals of this study including ICAPM-EGARCH with mixture innovation and the competing ARMA-GARCH with Student-*t* innovation with the estimation method of rw-MH, variational Bayes and QMLE. For the 100 stocks, standardised residuals of all the proposed models indicated independence where $\alpha = 0.01$ meant failure to reject the null hypothesis that the residuals are independently distributed, except for the stock abbreviations 'V', 'MA', 'TXN', 'QCOM', 'SCHW' and 'DEO' in ICAPM-EGARCH(1,1,1)-Mixture-rw-MH model. For the ICAPM-EGARCH(1,1,1)-Mixture-VBDA0 model, the stock abbreviations are 'V', 'TXN', 'QCOM' and 'DEO'. For the ICAPM-EGARCH(1,1,1)-Mixture-VBDA1 model, the stock abbreviations are 'V', 'MA', 'TXN', 'QCOM' and 'DEO'. For ICAPM-EGARCH(1,1,1)-Mixture-VBDA2 model, the stock abbreviations are 'V', 'TXN', 'QCOM' and 'DEO'. For ICAPM-EGARCH(1,1,1)-Mixture-QMLE model, the stock abbreviations are 'V', 'MA', 'TXN', 'QCOM', 'SCHW' and 'DEO'. For ARMA(1,1)-GARCH(1,1)-Student-*t*-QMLE model, the stock abbreviations are 'QCOM'and 'HON'.

B.5 Top 100 Stock Indexes Using in Empirical Study from the Nasdaq Market

No	Symbol	Name	Sector
1	AAPL	Apple Inc. Common Stock	Technology
2	MSFT	Microsoft Corporation Common Stock	Technology
3	GOOG	Alphabet Inc. Class C Capital Stock	Technology
4	GOOGL	Alphabet Inc. Class A Common Stock	Technology
5	AMZN	Amazon.com Inc. Common Stock	Consumer Services
6	MCHP	Microchip Technology Incorporated Common Stock	Technology
7	TSM	Taiwan Semiconductor Manufacturing Company Ltd.	Technology
8	NVDA	NVIDIA Corporation Common Stock	Technology
9	V	Visa Inc.	Miscellaneous
10	JPM	JP Morgan Chase & Co. Common Stock	Finance
11	IDXX	IDEXX Laboratories Inc. Common Stock	Health Care
12	JNJ	Johnson & Johnson Common Stock	Health Care
13	WMT	Walmart Inc. Common Stock	Consumer Services
14	UNH	United Health Group Incorporated Common Stock (DE)	Health Care
15	ADI	Analog Devices Inc. Common Stock	Technology
16	HD	Home Depot Inc. (The) Common Stock	Consumer Services
17	ASML	ASML Holding N.V. New York Registry Shares	Technology
18	PG	Procter & Gamble Company (The) Common Stock	Consumer Non-Durables
19	BAC	Bank of America Corporation Common Stock	Finance
20	MA	Mastercard Incorporated Common Stock	Miscellaneous
21	KLAC	KLA Corporation Common Stock	Capital Goods
22	DIS	Walt Disney Company (The) Common Stock	Consumer Services
23	ADBE	Adobe Inc. Common Stock	Technology
24	CMCSA	Comcast Corporation Class A Common Stock	Consumer Services
25	NFLX	Netflix Inc. Common Stock	Consumer Services

26	CRM	Salesforce.com Inc Common Stock	Technology
27	PFE	Pfizer Inc. Common Stock	Health Care
28	TM	Toyota Motor Corporation Common Stock	Capital Goods
29	NKE	Nike Inc. Common Stock	Consumer Non-Durables
30	CSCO	Cisco Systems Inc. Common Stock (DE)	Technology
31	ORCL	Oracle Corporation Common Stock	Technology
32	KO	Coca-Cola Company (The) Common Stock	Consumer Non-Durables
33	TMO	Thermo Fisher Scientific Inc Common Stock	Capital Goods
34	DHR	Danaher Corporation Common Stock	Health Care
35	NVO	Novo Nordisk A/S Common Stock	Health Care
36	XOM	Exxon Mobil Corporation Common Stock	Energy
37	VZ	Verizon Communications Inc. Common Stock	Public Utilities
38	LLY	Eli Lilly and Company Common Stock	Health Care
39	ABT	Abbott Laboratories Common Stock	Health Care
40	INTC	Intel Corporation Common Stock	Technology
41	NTES	NetEase Inc. American Depositary Shares	Miscellaneous
42	PEP	PepsiCo Inc. Common Stock	Consumer Non-Durables
43	ACN	Accenture plc Class A Ordinary Shares (Ireland)	Technology
44	COST	Costco Wholesale Corporation Common Stock	Consumer Services
45	T	AT&T Inc.	Consumer Services
46	WFC	Wells Fargo & Company Common Stock	Finance
47	NVS	Novartis AG Common Stock	Health Care
48	CVX	Chevron Corporation Common Stock	Energy
49	MRK	Merck & Company Inc. Common Stock (new)	Health Care
50	AZN	AstraZeneca PLC American Depositary Shares	Health Care
51	MS	Morgan Stanley Common Stock	Finance
52	MCD	McDonald's Corporation Common Stock	Consumer Services
53	TXN	Texas Instruments Incorporated Common Stock	Technology
54	MDT	Medtronic plc. Ordinary Shares	Health Care
55	UPS	United Parcel Service Inc. Common Stock	Transportation

56	SAP	SAP SE ADS	Technology
57	NEE	NextEra Energy Inc. Common Stock	Public Utilities
58	PM	Philip Morris International Inc Common Stock	Consumer Non-Durables
59	TMUS	T-Mobile US Inc. Common Stock	Public Utilities
60	LIN	Linde plc Ordinary Share	Basic Industries
61	INTU	Intuit Inc. Common Stock	Technology
62	QCOM	QUALCOMM Incorporated Common Stock	Technology
63	HON	Honeywell International Inc. Common Stock	
64	LOW	Lowe’s Companies Inc. Common Stock	Consumer Services
65	UL	Unilever PLC Common Stock	Consumer Non-Durables
66	RY	Royal Bank Of Canada Common Stock	
67	ILMN	Illumina Inc. Common Stock	Health Care
68	BHP	BHP Group Limited American Depositary Shares (Each representing two Ordinary Shares)	Basic Industries
69	C	Citigroup Inc. Common Stock	Finance
70	SONY	Sony Group Corporation American Depositary Shares	Consumer Non-Durables
71	HDB	HDFC Bank Limited Common Stock	Finance
72	BMY	Bristol-Myers Squibb Company Common Stock	Health Care
73	AMT	American Tower Corporation (REIT) Common Stock	Finance
74	SBUX	Starbucks Corporation Common Stock	
75	BLK	BlackRock Inc. Common Stock	Finance
76	SCHW	Charles Schwab Corporation (The) Common Stock	Finance
77	UNP	Union Pacific Corporation Common Stock	Transportation
78	BBL	BHP Group PlcSponsored ADR	Energy
79	AXP	American Express Company Common Stock	Finance
80	GS	Goldman Sachs Group Inc. (The) Common Stock	Finance
81	RTX	Raytheon Technologies Corporation Common Stock	Capital Goods
82	AMD	Advanced Micro Devices Inc. Common Stock	Technology
83	BA	Boeing Company (The) Common Stock	Capital Goods
84	AMAT	Applied Materials Inc. Common Stock	Capital Goods
85	AMGN	Amgen Inc. Common Stock	Health Care

86	ISRG	Intuitive Surgical Inc. Common Stock	Health Care
87	IBM	International Business Machines Corporation Common Stock	Technology
88	SNY	Sanofi ADS	Health Care
89	TGT	Target Corporation Common Stock	Consumer Services
90	TD	Toronto Dominion Bank (The) Common Stock	
91	TTE	TotalEnergies SE	Energy
92	EL	Estee Lauder Companies Inc. (The) Common Stock	Consumer Non-Durables
93	CVS	CVS Health Corporation Common Stock	Health Care
94	GE	General Electric Company Common Stock	Consumer Durables
95	DEO	Diageo plc Common Stock	Consumer Non-Durables
96	SPGI	S&P Global Inc. Common Stock	Technology
97	RIO	Rio Tinto Plc Common Stock	Basic Industries
98	JBHT	J.B. Hunt Transport Services Inc. Common Stock	Transportation
99	DE	Deere & Company Common Stock	Capital Goods
100	CAT	Caterpillar Inc. Common Stock	Capital Goods

Appendix C

Chapter 5 Additional Information

C.1 The Security Indexes Used in Empirical Study

The securities No. 1 to No. 30 are the stock index on the Nasdaq market. The securities No. 31 to No. 35 are the cryptocurrency index on the cryptocurrency market.

No	Symbol	Name	Sector
1	HDB	HDFC Bank Limited Common Stock	Finance
2	BMY	Bristol-Myers Squibb Company Common Stock	Health Care
3	AMT	American Tower Corporation (REIT) Common Stock	Finance
4	SBUX	Starbucks Corporation Common Stock	
5	BLK	BlackRock Inc. Common Stock	Finance
6	SCHW	Charles Schwab Corporation (The) Common Stock	Finance
7	UNP	Union Pacific Corporation Common Stock	Transportation
8	BBL	BHP Group PlcSponsored ADR	Energy

9	AXP	American Express Company Common Stock	Finance
10	GS	Goldman Sachs Group Inc. (The) Common Stock	Finance
11	RTX	Raytheon Technologies Corporation Common Stock	Capital Goods
12	AMD	Advanced Micro Devices Inc. Common Stock	Technology
13	BA	Boeing Company (The) Common Stock	Capital Goods
14	AMAT	Applied Materials Inc. Common Stock	Capital Goods
15	AMGN	Amgen Inc. Common Stock	Health Care
16	ISRG	Intuitive Surgical Inc. Common Stock	Health Care
17	IBM	International Business Machines Corporation Common Stock	Technology
18	SNY	Sanofi ADS	Health Care
19	TGT	Target Corporation Common Stock	Consumer Services
20	TD	Toronto Dominion Bank (The) Common Stock	
21	TTE	TotalEnergies SE	Energy
22	EL	Estee Lauder Companies Inc. (The) Common Stock	Consumer Non-Durables
23	CVS	CVS Health Corporation Common Stock	Health Care
24	GE	General Electric Company Common Stock	Consumer Durables
25	DEO	Diageo plc Common Stock	Consumer Non-Durables
26	SPGI	S&P Global Inc. Common Stock	Technology
27	RIO	Rio Tinto Plc Common Stock	Basic Industries
28	JBHT	J.B. Hunt Transport Services Inc. Common Stock	Transportation
29	DE	Deere & Company Common Stock	Capital Goods
30	CAT	Caterpillar Inc. Common Stock	Capital Goods
31	BTC	Bitcoin	Cryptocurrency
32	ETH	Ethereum	Cryptocurrency
33	USDT	Tether	Cryptocurrency
34	BNB	BNB	Cryptocurrency
35	XRP	XRP	Cryptocurrency

Appendix D

MATLAB Code Implementation

This appendix demonstrates the code implementation in the MATLAB for the particular proposed models in this thesis. There are three main sections in this appendix where (i) presents the code examples implement the Griddy-Gibbs algorithm; (ii) the code specimens implement the random-walk Metropolis-Hasting algorithm in Bayesian Markov chain Monte Carlo inference and the variational Bayes with data augmentation algorithm; and (iii) the code specimens implement the whole procedure of the Inference for the Margins method (IFM). The algorithms in the IFM include the algorithms of the forecasting model and the algorithm of the machine learning optimisation such as the clonal selection algorithm and the genetic algorithm. Note that all algorithms which presented in the thesis are implemented with the parallel computing technique when possible.

D.1 The Algorithms in Modelling and Forecasting COVID-19 Stock Returns using Asymmetric GARCH-ICAPM with Mixture and Heavy-Tailed Distributions

This part demonstrates the Griddy-Gibbs algorithm of the models of EGARCH-ICAPM-Mixture and EGARCH-ICAPM-GED.

D.1.1 The Griddy-Gibbs Algorithm for EGARCH-ICAPM-Mixture Model

```
% Griddy-Gibbs Algorithm in ICAPM-EGARCH-Mixture model
% Date 9 December 2022
% If you find any bugs, feel free to email me.
% By Rewat Khanthaporn
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

% Insert initial input
% x is initial guess of model parameters, i.e., from QMLE method.
% yt is a vector of time series data.
% The model parameters specification: xil_dw,xil_up,xir_dw,xir_up,
    lambda1_dw,
```

```
% lambda1_up,lambda2_dw,lambda2_up,omega_dw,omega_up,gamma_dw,
    gamma_up,
% delta_dw,delta_up,nu_dw,nu_up.

% Prior parameter:

% n_grid is number of grid points.
% draws is number of the MCMC iteration.
% diss is number of the burn-ins.


zeta=zeros(draws,8);
zeta(1,:)= x'; % arbitrary setting/initial guess

for i=2:draws
    zeta(i,1)=fnGG_ICAPM_EGARCH_Mix_xil1(yt,zeta(i-1,:),xil_dw,xil_up
        ,n_grid,a1,b1);
    zeta(i,2)=fnGG_ICAPM_EGARCH_Mix_xir1(yt,zeta(i-1:i,:),xir_dw,
        xir_up,n_grid,a2,b2);
    zeta(i,3)=fnGG_ICAPM_EGARCH_Mix_lambda11(yt,zeta(i-1:i,:),
        lambda1_dw,lambda1_up,n_grid,a3,b3);
    zeta(i,4)=fnGG_ICAPM_EGARCH_Mix_lambda21(yt,zeta(i-1:i,:),
        lambda2_dw,lambda2_up,n_grid,a4,b4);
    zeta(i,5)=fnGG_ICAPM_EGARCH_Mix_omega1(yt,zeta(i-1:i,:),omega_dw,
        omega_up,n_grid,a5,b5);
    zeta(i,6)=fnGG_ICAPM_EGARCH_Mix_gamma1(yt,zeta(i-1:i,:),gamma_dw,
        gamma_up,n_grid,a6,b6);
    zeta(i,7)=fnGG_ICAPM_EGARCH_Mix_delta1(yt,zeta(i-1:i,:),delta_dw,
        delta_up,n_grid,a7,b7);
    zeta(i,8)=fnGG_ICAPM_EGARCH_Mix_nu1(yt,zeta(i-1:i,:),nu_dw,nu_up,
        n_grid,a8,b8);
end
zeta=zeta(draws*diss:draws,:);
```

```
function delta_next=fnGG_ICAPM_EGARCH_Mix_delta1(yt,zeta,delta_dw,
    delta_up,n_grid,a7,b7)

xil=zeta(2,1);
xir=zeta(2,2);
lambda1=zeta(2,3);
lambda2=zeta(2,4);
omega=zeta(2,5);
gam=zeta(2,6);
nu=zeta(1,8);
%
p_next=[0.8;0.1;0.1];
betal=(1-xil)*(1-(2*xil))^0.5; %  beta
betar=(1-xir)*(1-(2*xir))^0.5; %  beta
delta_grid = linspace(delta_dw+rand/100,delta_up-rand/100,n_grid)';

n=size(yt,1);
hts1= zeros(n,n_grid);
z11 = zeros(n,n_grid);
hts1(1,:)=0.01;
z11(1,:)=(yt(1,1) - lambda1 - (lambda2.*exp(hts1(1,:)))./sqrt(exp(
    hts1(1,:)));
```

```
for t=2:n
    hts1(t,:)= omega + gam.*z11(t-1,:) + delta_grid'.*abs(z11(t-1,:))
         + nu.*(hts1(t-1,:));
    z11(t,:) = (yt(t,1) - lambda1 - (lambda2.*exp(hts1(t,:)))))./...
                                       sqrt(exp(hts1(t,:)));
end

et=z11.*sqrt(exp(hts1));
lik_normal = pdf('normal',et,0,sqrt(exp(hts1)));
lik_normal=lik_normal.*p_next(1);
lik_gpdr=pdf('Generalized Pareto', et,xir,betar.*sqrt(exp(hts1)),0);
lik_gpdr=lik_gpdr.*p_next(2);
lik_gpdl=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts1)),0);
lik_gpdl=lik_gpdl.*p_next(3);

liksum1 = lik_normal + lik_gpdr + lik_gpdl;
lglik1  = log(liksum1);
prior   = log(betapdf(delta_grid,a7,b7));
lglik   = sum(lglik1)+prior';

p_delta = exp(lglik - max(lglik)); % density (unnormalized)
p_delta = p_delta/sum(p_delta); % density (normalized)
cdf_delta = cumsum(p_delta); % cdf
delta_next = delta_grid(find(rand<cdf_delta, 1));
```

```
function gamma_next=fnGG_ICAPM_EGARCH_Mix_gamma1(yt,zeta,gamma_dw,
    gamma_up,n_grid,a6,b6)

xil=zeta(2,1);
xir=zeta(2,2);
lambda1=zeta(2,3);
lambda2=zeta(2,4);
omega=zeta(2,5);
delta=zeta(1,7);
nu=zeta(1,8);
%

p_next=[0.8;0.1;0.1];
betal=(1-xil)*(1-(2*xil))^0.5; %  beta
betar=(1-xir)*(1-(2*xir))^0.5; %  beta
gamma_grid = linspace(gamma_dw+rand/100,gamma_up-rand/100,n_grid)';

n=size(yt,1);
hts1= zeros(n,n_grid);
z11 = zeros(n,n_grid);
hts1(1,:)=0.01;
z11(1,:)=(yt(1,1) - lambda1 - (lambda2.*exp(hts1(1,:)))))./sqrt(exp(
    hts1(1,:)));
for t=2:n
    hts1(t,:)= omega + gamma_grid'.*z11(t-1,:) + delta.*abs(z11(t
        -1,:)) + nu.*(hts1(t-1,:));
    z11(t,:) = (yt(t,1) - lambda1 - (lambda2.*exp(hts1(t,:)))))./...
                                       sqrt(exp(hts1(t,:)));
end

et=z11.*sqrt(exp(hts1));
```

```

lik_normal = pdf('normal',et,0,sqrt(exp(hts1)));
lik_normal=lik_normal.*p_next(1);
lik_gpdr=pdf('Generalized Pareto', et,xir,betar.*sqrt(exp(hts1)),0);
lik_gpdr=lik_gpdr.*p_next(2);
lik_gpdl=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts1)),0);
lik_gpdl=lik_gpdl.*p_next(3);

liksum1 = lik_normal + lik_gpdr + lik_gpdl;
lglik1  = log(liksum1);
prior  = log(normpdf(gamma_grid,a6,b6));
lglik  = sum(lglik1)+prior';

p_gamma = exp(lglik - max(lglik)); % density (unnormalized)
p_gamma = p_gamma/sum(p_gamma); % density (normalized)
cdf_gamma = cumsum(p_gamma); % cdf
gamma_next = gamma_grid(find(rand<cdf_gamma, 1));
```

```
function lambda1_next=fnGG_ICAPM_EGARCH_Mix_lambda11(yt,zeta,
    lambda1_dw,lambda1_up,n_grid,a3,b3)

xil=zeta(2,1);
xir=zeta(2,2);
lambda2=zeta(1,4);
omega=zeta(1,5);
gam=zeta(1,6);
delta=zeta(1,7);
nu=zeta(1,8);

p_next=[0.8;0.1;0.1];

betal=(1-xil)*(1-(2*xil))^0.5; %  beta
betar=(1-xir)*(1-(2*xir))^0.5; %  beta
lambda1_grid = linspace(lambda1_dw+rand/100,lambda1_up-rand/100,
    n_grid)';

n=size(yt,1);
hts1= zeros(n,n_grid);
z11 = zeros(n,n_grid);
hts1(1,:)=0.01;
z11(1,:)=(yt(1,1) - lambda1_grid' - (lambda2.*exp(hts1(1,:)))./sqrt(
    exp(hts1(1,:)));
for t=2:n
     hts1(t,:)=omega + gam.*z11(t-1,:) + delta.*abs(z11(t-1,:)) + nu
         .*(hts1(t-1,:));
     z11(t,:)=(yt(t,1) - lambda1_grid' - (lambda2.*exp(hts1(t,:))))
         ./...
                                    sqrt(exp(hts1(t,:)));
end

et=z11.*sqrt(exp(hts1));
lik_normal = pdf('normal',et,0,sqrt(exp(hts1)));
lik_normal=lik_normal.*p_next(1);
lik_gpdr=pdf('Generalized Pareto', et,xir,betar.*sqrt(exp(hts1)),0);
lik_gpdr=lik_gpdr.*p_next(2);
lik_gpdl=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts1)),0);
```

```
lik_gpdl=lik_gpdl.*p_next(3);

liksum1 = lik_normal + lik_gpdr + lik_gpdl;
lglik1  = log(liksum1);
prior   = log(normpdf(lambda1_grid,a3,b3));
lglik   = sum(lglik1)+prior';

p_lambda1 = exp(lglik - max(lglik)); % density (unnormalized)
p_lambda1 = p_lambda1/sum(p_lambda1); % density (normalized)
cdf_lambda1 = cumsum(p_lambda1); % cdf
lambda1_next = lambda1_grid(find(rand<cdf_lambda1, 1));
```

```
function lambda2_next=fnGG_ICAPM_EGARCH_Mix_lambda21(yt,zeta,
    lambda2_dw,lambda2_up,n_grid,a4,b4)

xil=zeta(2,1);
xir=zeta(2,2);
lambda1=zeta(2,3);
omega=zeta(1,5);
gam=zeta(1,6);
delta=zeta(1,7);
nu=zeta(1,8);
p_next=[0.8;0.1;0.1];
betal=(1-xil)*(1-(2*xil))^0.5; %  beta
betar=(1-xir)*(1-(2*xir))^0.5; %  beta
lambda2_grid = linspace(lambda2_dw+rand/100,lambda2_up-rand/100,
    n_grid)';

n=size(yt,1);
hts1= zeros(n,n_grid);
z11 = zeros(n,n_grid);
hts1(1,:)=0.01;
z11(1,:)=(yt(1,1) - lambda1 - (lambda2_grid'.*exp(hts1(1,:))))./sqrt(
    exp(hts1(1,:)));
for t=2:n
    hts1(t,:)=omega + gam.*z11(t-1,:) + delta.*abs(z11(t-1,:)) + nu
        .*(hts1(t-1,:));
    z11(t,:)=(yt(t,1) - lambda1 - (lambda2_grid'.*exp(hts1(t,:))))
        ./...
                                    sqrt(exp(hts1(t,:)));
end

et=z11.*sqrt(exp(hts1));
lik_normal = pdf('normal',et,0,sqrt(exp(hts1)));
lik_normal=lik_normal.*p_next(1);
lik_gpdr=pdf('Generalized Pareto', et,xir,betar.*sqrt(exp(hts1)),0);
lik_gpdr=lik_gpdr.*p_next(2);
lik_gpdl=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts1)),0);
lik_gpdl=lik_gpdl.*p_next(3);

liksum1 = lik_normal + lik_gpdr + lik_gpdl;
lglik1  = log(liksum1);
prior   = log(normpdf(lambda2_grid,a4,b4));
lglik   = sum(lglik1)+prior';

p_lambda2 = exp(lglik - max(lglik)); % density (unnormalized)
```

```
p_lambda2 = p_lambda2/sum(p_lambda2); % density (normalized)
cdf_lambda2 = cumsum(p_lambda2); % cdf
lambda2_next = lambda2_grid(find(rand<cdf_lambda2, 1));
```

```
function nu_next=fnGG_ICAPM_EGARCH_Mix_nu1(yt,zeta,nu_dw,nu_up,n_grid
    ,a8,b8)

xil=zeta(2,1);
xir=zeta(2,2);
lambda1=zeta(2,3);
lambda2=zeta(2,4);
omega=zeta(2,5);
gam=zeta(2,6);
delta=zeta(2,7);
p_next=[0.98;0.01;0.01];

betal=(1-xil)*(1-(2*xil))^0.5; %  beta
betar=(1-xir)*(1-(2*xir))^0.5; %  beta
nu_grid = linspace(nu_dw+rand/100,nu_up-rand/100,n_grid)';

n=size(yt,1);
hts1= zeros(n,n_grid);
z11 = zeros(n,n_grid);
hts1(1,:)=0.01;
z11(1,:)=(yt(1,1) - lambda1 - (lambda2.*exp(hts1(1,:)))./sqrt(exp(
    hts1(1,:)));
for t=2:n
    hts1(t,:)= omega + gam.*z11(t-1,:) + delta.*abs(z11(t-1,:)) +
        nu_grid'.*(hts1(t-1,:));
    z11(t,:) = (yt(t,1) - lambda1 - (lambda2.*exp(hts1(t,:)))./...
                                    sqrt(exp(hts1(t,:)));
end

et=z11.*sqrt(exp(hts1));
lik_normal = pdf('normal',et,0,sqrt(exp(hts1)));
lik_normal=lik_normal.*p_next(1);
lik_gpdr=pdf('Generalized Pareto', et,xir,betar.*sqrt(exp(hts1)),0);
lik_gpdr=lik_gpdr.*p_next(2);
lik_gpdl=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts1)),0);
lik_gpdl=lik_gpdl.*p_next(3);

liksum1 = lik_normal + lik_gpdr + lik_gpdl;
lglik1  = log(liksum1);
prior   = log(betapdf(nu_grid,a8,b8));
lglik   = sum(lglik1)+prior';

p_nu = exp(lglik - max(lglik)); % density (unnormalized)
p_nu = p_nu/sum(p_nu); % density (normalized)
cdf_nu = cumsum(p_nu); % cdf
nu_next = nu_grid(find(rand<cdf_nu, 1));
```

```
function omega_next=fnGG_ICAPM_EGARCH_Mix_omega1(yt,zeta,omega_dw,
    omega_up,n_grid,a5,b5)

xil=zeta(2,1);
```

```
xir=zeta(2,2);
lambda1=zeta(2,3);
lambda2=zeta(2,4);
gam=zeta(1,6);
delta=zeta(1,7);
nu=zeta(1,8);
%

p_next=[0.8;0.1;0.1];
betal=(1-xil)*(1-(2*xil))^0.5; %  beta
betar=(1-xir)*(1-(2*xir))^0.5; %  beta
omega_grid = linspace(omega_dw+rand/100,omega_up-rand/100,n_grid)';

n=size(yt,1);
hts1= zeros(n,n_grid);
z11 = zeros(n,n_grid);
hts1(1,:)=0.01;
z11(1,:)=(yt(1,1) - lambda1 - (lambda2.*exp(hts1(1,:)))))./sqrt(exp(
    hts1(1,:)));
for t=2:n
    hts1(t,:)=omega_grid' + gam.*z11(t-1,:) + delta.*abs(z11(t-1,:))
        + nu.*(hts1(t-1,:));
    z11(t,:)=(yt(t,1) - lambda1 - (lambda2.*exp(hts1(t,:))))./...
                                   sqrt(exp(hts1(t,:)));
end

et=z11.*sqrt(exp(hts1));
lik_normal = pdf('normal',et,0,sqrt(exp(hts1)));
lik_normal=lik_normal.*p_next(1);
lik_gpdr=pdf('Generalized Pareto', et,xir,betar.*sqrt(exp(hts1)),0);
lik_gpdr=lik_gpdr.*p_next(2);
lik_gpdl=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts1)),0);
lik_gpdl=lik_gpdl.*p_next(3);

liksum1 = lik_normal + lik_gpdr + lik_gpdl;
lglik1  = log(liksum1);
prior   = log(normpdf(omega_grid,a5,b5));
lglik   = sum(lglik1)+prior';

p_omega = exp(lglik - max(lglik)); % density (unnormalized)
p_omega = p_omega/sum(p_omega); % density (normalized)
cdf_omega = cumsum(p_omega); % cdf
omega_next = omega_grid(find(rand<cdf_omega, 1));
```

```
function xil_next=fnGG_ICAPM_EGARCH_Mix_xil1(yt,zeta,xil_dw,xil_up,
    n_grid,a1,b1)

xir=zeta(1,2);
lambda1=zeta(1,3);
lambda2=zeta(1,4);
omega=zeta(1,5);
gam=zeta(1,6);
delta=zeta(1,7);
nu=zeta(1,8);
%
p_next=[0.8;0.1;0.1];
```

```
xil_grid = linspace(xil_dw+rand/100,xil_up-rand/100,n_grid)';

betar=(1-xir)*(1-(2*xir))^0.5; %  beta
betal=(1-xil_grid).*(1-(2.*xil_grid)).^0.5; %  beta

n=size(yt,1);
hts1= zeros(n,1);
z11 = zeros(n,1);
hts1(1,1)=0.01;
z11(1,1)=(yt(1,1) - lambda1 - (lambda2.*exp(hts1(1,1)))./sqrt(exp(
    hts1(1,1)));
for t=2:n
    hts1(t,1)=omega + gam.*z11(t-1,1) + delta.*abs(z11(t-1,1)) + nu
        .*(hts1(t-1,1));
    z11(t,1)=(yt(t,1) - lambda1 - (lambda2.*exp(hts1(t,1))))./...
                                        sqrt(exp(hts1(t,1)));
end

et=z11.*sqrt(exp(hts1));
lik_normal = pdf('normal',et,0,sqrt(exp(hts1)));
lik_normal=lik_normal.*p_next(1);

a=sqrt(exp(hts1))*betal';
b=ones(1,n_grid);
c=-et*b;
lik_gpdl=zeros(n,n_grid);
parfor i=1:n_grid
    lik_gpdl(:,i)=pdf('Generalized Pareto', c(:,i),xil_grid(i),a(:,i)
        ,0);
end
lik_gpdl=lik_gpdl.*p_next(2);

lik_gpdr=pdf('Generalized Pareto',et,xir,betar.*sqrt(exp(hts1)),0);
lik_gpdr=lik_gpdr.*p_next(3);

liksum1 = lik_normal + lik_gpdr + lik_gpdl;
lglik1  = log(liksum1);
prior   = log(unifpdf(xil_grid,a1,b1));
lglik   = sum(lglik1)+prior';

p_xil = exp(lglik - max(lglik)); % density (unnormalized)
p_xil = p_xil/sum(p_xil); % density (normalized)
cdf_xil = cumsum(p_xil); % cdf
xil_next = xil_grid(find(rand<cdf_xil, 1));
```

```
function xir_next=fnGG_EGARCH_MixNrLfRtGPD_xir1(yt,zeta,xir_dw,xir_up
    ,n_grid,a2,b2)

xil=zeta(2,1);
lambda1=zeta(1,3);
lambda2=zeta(1,4);
omega=zeta(1,5);
gam=zeta(1,6);
delta=zeta(1,7);
nu=zeta(1,8);
%
```

```

p_next=[0.8;0.1;0.1];
xir_grid = linspace(xir_dw+rand/100,xir_up-rand/100,n_grid)';

betal=(1-xil)*(1-(2*xil))^0.5; %  beta
betar=(1-xir_grid).*(1-(2.*xir_grid)).^0.5; %  beta

n=size(yt,1);
hts1= zeros(n,1);
z11 = zeros(n,1);
hts1(1,1)=0.01;
z11(1,1)=(yt(1,1) - lambda1 - (lambda2.*exp(hts1(1,1)))))./sqrt(exp(
    hts1(1,1)));
for t=2:n
    hts1(t,1)=omega + gam.*z11(t-1,1) + delta.*abs(z11(t-1,1)) + nu
        .*(hts1(t-1,1));
    z11(t,1)=(yt(t,1) - lambda1 - (lambda2.*exp(hts1(t,1))))./...
                                        sqrt(exp(hts1(t,1)));
end

et=z11.*sqrt(exp(hts1));
lik_normal = pdf('normal',et,0,sqrt(exp(hts1)));
lik_normal=lik_normal.*p_next(1);
a=sqrt(exp(hts1))*betar';
b=ones(1,n_grid);
c=et*b;
lik_gpdr=zeros(n,n_grid);
parfor i=1:n_grid
    lik_gpdr(:,i)=pdf('Generalized Pareto', c(:,i),xir_grid(i),a(:,i)
        ,0);
end
lik_gpdr=lik_gpdr.*p_next(2);
lik_gpdl=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts1)),0);
lik_gpdl=lik_gpdl.*p_next(3);

liksum1 = lik_normal + lik_gpdr + lik_gpdl;
lglik1  = log(liksum1);
prior   = log(unifpdf(xir_grid,a2,b2));
lglik   = sum(lglik1)+prior';

p_xir = exp(lglik - max(lglik)); % density (unnormalized)
p_xir = p_xir/sum(p_xir); % density (normalized)
cdf_xir = cumsum(p_xir); % cdf
xir_next = xir_grid(find(rand<cdf_xir, 1));
```

D.1.2 The Griddy-Gibbs Algorithm for EGARCH-ICAPM-GED Model

```
% Griddy-Gibbs Algorithm in ICAPM-EGARCH-GED model
% Date 9 December 2022
% If you find bugs, feel free to email me.
% By Rewat Khanthaporn
```

```
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

% Insert initial input
% x is initial guess of model parameters, i.e., from QMLE method.
% yt is a vector of time series data.
% The model parameters specification: lamda2_up, omega_up,omega_dw,
% gam_up,gam_dw,delta_up,delta_dw,nu_up,beta_up,lamda1_dw,lamda1_up.
% Prior parameter: lb1para,lb2para,omega1,omega2,,gam1,gam2,delta1,delta2
% nu1,nu2,beta1,beta2,lamda1,lamda2.
% n_grid is number of grid points.
% draws is number of the MCMC iteration.
% diss is number of the burn-ins.

zeta=zeros(draws,7);
x0=x;
zeta(1,:)= x0';

for i=2:draws
    zeta(i,1)=fnGG_EGARCH_M_GED_lambda2(yt,zeta(i-1,:),lamda2_up,n_grid,lb1para,lb2para);
    zeta(i,2)=fnGG_EGARCH_M_GED_omega(yt,zeta(i-1:i,:),omega_up,omega_dw,n_grid,omega1,omega2);
    zeta(i,3)=fnGG_EGARCH_M_GED_gam(yt,zeta(i-1:i,:),gam_up,gam_dw,n_grid,gam1,gam2);
    zeta(i,4)=fnGG_EGARCH_M_GED_delta(yt,zeta(i-1:i,:),delta_up,delta_dw,n_grid,delta1,delta2);
    zeta(i,5)=fnGG_EGARCH_M_GED_nu(yt,zeta(i-1:i,:),nu_up,n_grid,nu1,nu2);
    zeta(i,6)=fnGG_EGARCH_M_GED_beta(yt,zeta(i-1:i,:),beta_up,beta_dw,n_grid,beta1,beta2);
    zeta(i,7)=fnGG_EGARCH_M_GED_lambda1(yt,zeta(i-1:i,:),lamda1_up,lamda1_dw,n_grid,lamda1,lamda2);
end
```

```
function beta_next=fnGG_EGARCH_M_GED_beta(yt,zeta,beta_up,beta_dw,n_grid,beta1,beta2)
lamda2=zeta(2,1);
omega=zeta(2,2);
gam=zeta(2,3);
delta=zeta(2,4);
nu=zeta(2,5);
lamda1=zeta(1,7);
y=yt;
T=size(y,1);

hts= zeros(T,1);
z1 = zeros(T,1);
c  = zeros(T,n_grid);
a  = zeros(n_grid,1);
hts(1,1)=0.01; z1(1,1)=0.01;
beta_grid = linspace(beta_dw+rand/100,beta_up-rand/100,n_grid)';

for t=2:T
    hts(t,1)=omega + gam*z1(t-1,1) + delta*abs(z1(t-1,1)) + nu*(hts(t-1,1));
```

```
    z1(t,1)=(y(t,1) - lamda1 - (lamda2*exp(hts(t,1))))...
             /sqrt(exp(hts(t,1)));
end

% Perform log likelihood of each grid
b = exp(hts); b=sqrt(b);
b = 2*sum( log(b(2:T,1)),1);
alpha  = sqrt(gamma(1./beta_grid)./gamma(3./beta_grid));
c1 = abs(z1)./alpha';
for i=1:n_grid
    c(:,i)=c1(:,i).^beta_grid(i);
end
c=sum(c);
a = (T-1)*log( beta_grid.*sqrt(gamma(3./beta_grid))./((gamma(1./
   beta_grid)).^(3/2)));

prior=log(gampdf(beta_grid,beta1,beta2));
lglik= a-b-c'+prior;

p_beta = exp(lglik - max(lglik)); % density (unnormalized)
p_beta = p_beta/sum(p_beta); % density (normalized)
cdf_beta = cumsum(p_beta); % cdf
beta_next = beta_grid(find(rand<cdf_beta, 1));
```

```
function delta_next=fnGG_EGARCH_M_GED_delta(yt,zeta,delta_up,delta_dw
   ,n_grid,delta1,delta2)
lamda2=zeta(2,1);
omega=zeta(2,2);
gam=zeta(2,3);
nu=zeta(1,5);
beta=zeta(1,6);
lamda1=zeta(1,7);
alpha  = sqrt(gamma(1/beta)/gamma(3/beta));

y=yt;
T=size(y,1);
hts= zeros(n_grid, T);
z1 = zeros(n_grid, T);
hts(:,1)=0.01; z1(:,1)=0.01;

delta_grid = linspace(delta_dw+rand/100,delta_up-rand/100,n_grid)';

for t=2:T
    hts(:,t)=omega + gam.*z1(:,t-1) + delta_grid.*abs(z1(:,t-1)) + nu
       .*(hts(:,t-1));
    z1(:,t)=(y(t,1) - lamda1 - (lamda2.*exp(hts(:,t))))...
              ./sqrt(exp(hts(:,t)));
end

% Perform log likelihood of each grid
b = exp(hts); b=sqrt(b); b = 2*sum( log(b(:,2:T)),2);
c = abs(z1)./alpha;
c = sum(c(:,2:T).^beta,2);
a = (T-1)*log( beta*sqrt(gamma(3/beta))/((gamma(1/beta))^(3/2)));
prior=log(gampdf(delta_grid,delta1,delta2));
lglik= a-b-c+prior;
```

```
p_delta = exp(lglik - max(lglik)); % density (unnormalized)
p_delta = p_delta/sum(p_delta); % density (normalized)
cdf_delta = cumsum(p_delta); % cdf
delta_next = delta_grid(find(rand<cdf_delta, 1));
```

```
function gam_next=fnGG_EGARCH_M_GED_gam(yt,zeta,gam_up,gam_dw,n_grid,
    gam1,gam2)
lamda2=zeta(2,1);
omega=zeta(2,2);
delta=zeta(1,4);
nu=zeta(1,5);
beta=zeta(1,6);
lamda1=zeta(1,7);
alpha  = sqrt(gamma(1/beta)/gamma(3/beta));
y=yt;
T=size(y,1);

hts= zeros(n_grid, T);
z1 = zeros(n_grid, T);
hts(:,1)=0.01; z1(:,1)=0.01;
gam_grid = linspace(gam_dw+rand/100,gam_up-rand/100,n_grid)';

for t=2:T
    hts(:,t)=omega + gam_grid.*z1(:,t-1) + delta.*abs(z1(:,t-1)) + nu
        .*(hts(:,t-1));
    z1(:,t)=(y(t,1) - lamda1 - (lamda2.*exp(hts(:,t))))...
              ./sqrt(exp(hts(:,t)));
end

% Perform log likelihood of each grid
b = exp(hts); b=sqrt(b); b = 2*sum( log(b(:,2:T)),2);
c = abs(z1)./alpha;
c = sum(c(:,2:T).^beta,2);
a = (T-1)*log( beta*sqrt(gamma(3/beta))/((gamma(1/beta))^(3/2)));
prior=log(normpdf(gam_grid,gam1,gam2));
lglik= a-b-c+prior;

p_gam = exp(lglik - max(lglik)); % density (unnormalized)
p_gam = p_gam/sum(p_gam); % density (normalized)
cdf_gam = cumsum(p_gam); % cdf
gam_next = gam_grid(find(rand<cdf_gam, 1));
```

```
function lamda1_next=fnGG_EGARCH_M_GED_lambda1(yt,zeta,lambda1_up,
    lambda1_dw,n_grid,lamda1,lamda2)

lambda2=zeta(2,1);
omega=zeta(2,2);
gam=zeta(2,3);
delta=zeta(2,4);
nu=zeta(2,5);
beta=zeta(2,6);
alpha  = sqrt(gamma(1/beta)/gamma(3/beta));

y=yt;
```

```
T=size(y,1);

hts= zeros(n_grid, T);
z1 = zeros(n_grid, T);
hts(:,1)=0.01; z1(:,1)=0.01;
lambda1_grid = linspace(lambda1_dw+rand/100,lambda1_up-rand/100,
    n_grid)';

for t=2:T
    hts(:,t)=omega + gam.*z1(:,t-1) + delta.*abs(z1(:,t-1)) + nu.*(
        hts(:,t-1));
    z1(:,t)=(y(t,1) - lambda1_grid - (lambda2.*exp(hts(:,t))))...
                ./sqrt(exp(hts(:,t)));
end

% Perform log likelihood of each grid
b = exp(hts); b=sqrt(b); b = 2*sum( log(b(:,2:T)),2);
c = abs(z1)./alpha;
c = sum(c(:,2:T).^beta,2);
a = (T-1)*log( beta*sqrt(gamma(3/beta))/((gamma(1/beta))^(3/2)));
prior=log(betapdf(lambda1_grid,lamda1,lamda2));
lglik= a-b-c+prior;

p_lamda1 = exp(lglik - max(lglik)); % density (unnormalized)
p_lamda1 = p_lamda1/sum(p_lamda1); % density (normalized)
cdf_lamda1 = cumsum(p_lamda1); % cdf
lamda1_next = lambda1_grid(find(rand<cdf_lamda1, 1));
```

```
function lamda2_next=fnGG_EGARCH_M_GED_lambda2(yt,zeta,lamda2_up,
    n_grid,lb1para,lb2para)
omega=zeta(1,2);
gam=zeta(1,3);
delta=zeta(1,4);
nu=zeta(1,5);
beta=zeta(1,6);
lamda1=zeta(1,7);
alpha   = sqrt(gamma(1/beta)/gamma(3/beta));

T=size(yt,1);
hts= zeros(n_grid, T);
z1 = zeros(n_grid, T);
hts(:,1)=0.01; z1(:,1)=0.01;

lamda2_grid = linspace(0+rand/100,lamda2_up-rand/100,n_grid)';

for t=2:T
    hts(:,t)=omega + gam.*z1(:,t-1) + delta.*abs(z1(:,t-1)) + nu.*(
        hts(:,t-1));
    z1(:,t)=(yt(t,1) - lamda1 - (lamda2_grid.*exp(hts(:,t))))...
                ./sqrt(exp(hts(:,t)));
end

% Perform log likelihood of each grid
b = exp(hts); b=sqrt(b); b = 2*sum( log(b(:,2:T)),2);
c = abs(z1)./alpha;
c = sum(c(:,2:T).^beta,2);
```

```
a = (T-1)*log( beta*sqrt(gamma(3/beta))/((gamma(1/beta))^(3/2)));
prior=log(gampdf(lamda2_grid,lb1para,lb2para));
lglik= a-b-c+prior;

p_lamda2 = exp(lglik - max(lglik)); % unnormalized density
p_lamda2 = p_lamda2/sum(p_lamda2); % normalized density
cdf_lamda2 = cumsum(p_lamda2); % cdf
lamda2_next = lamda2_grid(find(rand<cdf_lamda2, 1));
```

```
function nu_next=fnGG_EGARCH_M_GED_nu(yt,zeta,nu_up,n_grid,nu1,nu2)

lamda2=zeta(2,1);
omega=zeta(2,2);
gam=zeta(2,3);
delta=zeta(2,4);
beta=zeta(1,6);
lamda1=zeta(1,7);
alpha   = sqrt(gamma(1/beta)/gamma(3/beta));
y=yt;
T=size(y,1);
hts= zeros(n_grid, T);
z1 = zeros(n_grid, T);
hts(:,1)=0.01; z1(:,1)=0.01;
nu_grid = linspace(0.5+rand/100,nu_up-rand/100,n_grid)';

for t=2:T
    hts(:,t)=omega + gam.*z1(:,t-1) + delta.*abs(z1(:,t-1)) + nu_grid
        .*(hts(:,t-1));
    z1(:,t)=(y(t,1) - lamda1 - (lamda2.*exp(hts(:,t))))...
              ./sqrt(exp(hts(:,t)));
end

% Perform log likelihood of each grid
b = exp(hts); b=sqrt(b); b = 2*sum( log(b(:,2:T)),2);
c = abs(z1)./alpha;
c = sum(c(:,2:T).^beta,2);
a = (T-1)*log( beta*sqrt(gamma(3/beta))/((gamma(1/beta))^(3/2)));
prior=log(gampdf(nu_grid,nu1,nu2));
lglik= a-b-c+prior;

p_nu = exp(lglik - max(lglik)); % density (unnormalized)
p_nu = p_nu/sum(p_nu); % density (normalized)
cdf_nu = cumsum(p_nu); % cdf
nu_next = nu_grid(find(rand<cdf_nu, 1));
```

```
function omega_next=fnGG_EGARCH_M_GED_omega(yt,zeta,omega_up,omega_dw
    ,n_grid,omega1,omega2)

lamda2=zeta(2,1);
gam=zeta(1,3);
delta=zeta(1,4);
nu=zeta(1,5);
beta=zeta(1,6);
lamda1=zeta(1,7);
alpha   = sqrt(gamma(1/beta)/gamma(3/beta));
```

```

y=yt;
T=size(y,1);
hts= zeros(n_grid, T);
z1 = zeros(n_grid, T);
hts(:,1)=0.01; z1(:,1)=0.01;
omega_grid = linspace(omega_dw+rand/100,omega_up-rand/100,n_grid)';

for t=2:T
    hts(:,t)=omega_grid + gam.*z1(:,t-1) + delta.*abs(z1(:,t-1)) + nu
        .*(hts(:,t-1));
    z1(:,t)=(y(t,1) - lamda1 - (lamda2.*exp(hts(:,t))))...
              ./sqrt(exp(hts(:,t)));
end

% Perform log likelihood of each grid
b = exp(hts); b=sqrt(b); b = 2*sum( log(b(:,2:T)),2);
c = abs(z1)./alpha;
c = sum(c(:,2:T).^beta,2);
a = (T-1)*log( beta*sqrt(gamma(3/beta))/((gamma(1/beta))^(3/2)));
prior=log(normpdf(omega_grid,omega1,omega2));
lglik= a-b-c+prior;

p_omega = exp(lglik - max(lglik)); % density (unnormalized)
p_omega = p_omega/sum(p_omega); % density (normalized)
cdf_omega = cumsum(p_omega); % cdf
omega_next = omega_grid(find(rand<cdf_omega, 1));
```

D.2 The Algorithms in Variational Bayes with Latent Variables and Data Augmentation in High N-dimensional Mixed Regular Vine Dependence Structure and Mixture Innovation Model in Financial Time Series

D.2.1 The Random-walk Metropolis-Hasting Algorithm of the R-vine-ICAPM-EGARCH-Mixture model (The IFM Method)

The IFM Step 1

```
% rw-MH Algorithm in R-vine-ICAPM-EGARCH-Mixture model
% Date 9 December 2022
% If you find any bugs, feel free to email me.
% By Rewat Khanthaporn
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

clear;

% Identification of input
% ret_demean: time series data
% c: any constant number
% sigma: data standard deviation
```

```

% x_margin: initial gauss, i.e. from QMLE method.

n=size(ret_demean,2);

parfor i=1:n  % parallel computing
    rt_run=ret_demean(3:ret_demean(2,i)+2,i)';
    thetdraw_all=x_margin(i,:);
    blc=8;
    sz=blc;
    lp_blc=size(thetdraw_all,2)/blc;
    store_theta_MH_EGARCH =zeros(1,size(thetdraw_all,2));
    C=1;
    %number of burning MH replications
    nburn=3000;
    %number of included MH replications
    nrep=20000;
    x0_MH=zeros(nrep-nburn,size(x_margin(i,:),2));
    x0_all_bk=x_margin(i,:);
    nswitch = 0;
    % input: ua from above
    thetdraw_all_pre=x_margin(i,:);
    thetdraw_all_next=thetdraw_all_pre;
    ctt=0;ct_store=0;
    lp=nrep+nburn;
    for irep=1:lp
        theta_chnged=0;
        for i_blc=1:lp_blc
            lb=1+(i_blc-1)*blc;
            ub=lb+blc-1;
            sig_blc=sigma(lb:ub,lb:ub);
            thetdraw=thetdraw_all_pre(lb:ub);
            num_para=size(thetdraw,2);
            nswitch = 0; ct=0;
            thmean=zeros(1,sz);
            th2mo=zeros(sz,sz);
            sig_blc=C.*sig_blc;
            llike1=fn_MH_Blc4_EGARCH_Loglik(rt_run,thetdraw_all_pre);
            thetcan = thetdraw + norm_rnd(sig_blc)';
            llike2=fn_MH_ICAPM_EGARCH_Mixture(rt_run,thetcan);
            %log of acceptance probability
            laccprob = llike2 - llike1;

            % now accept candidate with appropriate probability
            if laccprob > log(rand) && ~isinf(laccprob) && ~isnan(
                laccprob)
                thetdraw=thetcan;
                theta_chnged=1;
                thetdraw_all_next(lb:ub)=thetdraw;
                if theta_chnged==1 && i_blc==lp_blc
                   ctt=ctt+1;
                   if irep > nburn
                       ct_store=ct_store+1;
                       store_theta_MH_EGARCH(ct_store,:) =
                           thetdraw_all_next;
                   end
```

```
                    thetdraw_all_pre=thetdraw_all_next;
                elseif sum(thetdraw_all_next ~= thetdraw_all_pre) > 0
                     && i_blc~=lp_blc
                    thetdraw_all_pre=thetdraw_all_next;
                end
            end
            series_store_theta_MH_EGARCH{i}=store_theta_MH_EGARCH
                (1:5:end,:);
            series_acpted_rate(i)=ct_store/(irep-nburn);
        end
    end
end
```

The IFM Step 2

```
% rw-MH Algorithm in R-vine-ICAPM-EGARCH-Mixture model
% Date 9 December 2022
% If you find any bugs, feel free to email me.
% By Rewat Khanthaporn
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

% Copula function list
% Matlab toolbox:
%        - 'Clayton'       = 1    used
%        - 'Frank'         = 2    used
%        - 'Gumbel'(EV)    = 3    used
% My coding:  1-parameter Archimedean copula
%        - 'Joe'/'B5'      = 4    used
%        - 'Galambos'(EV)  = 5    used
% 2-parameters copula
%        - 'BB1'           = 6    used
%        - 'BB2'           = 7    not used
%        - 'BB3'           = 8    used
%        - 'BB4'           = 9    used
%        - 'BB5'           = 10   not used
%        - 'BB6'           = 11   used
%        - 'BB7'           = 12   used
%        - 'BB8'           = 13   not used
%        - 'BB9','Crowder'  = 14  not used
%        - 'BB10'          = 15   used
% Elliptical copula
%        - 'Gaussian'      =16    used
%        - 't'             =17    used
clear; clc;

% input area

% ret: a metrix of returns (T by n); T=time, n= number of asset.
% series_theta_MH_EGARCH_Mean: out of step 1 to be input here.

global family_list
para_num = [ 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2];
family_list= ["Clayton","Frank","Gumbel","B5","Galambos" ...
             ,"BB1","BB2","BB3","BB4","BB5","BB6","BB7","BB8"...
             ,"BB9","BB10","Gaussian","t"];
% below is full list of copula fam for empirical data
```

```
global fix_fun
fix_fun=[1:5,16,17,6,8,9,11,12,15];
num_cop_fam=length(fix_fun);

%
global zt
global i
global k
global R_vine_family
global u
global v

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BEGIN: convert ret to unif(0,1)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
parfor lp=1:n
    x_margin(lp,:)=series_theta_MH_EGARCH_Mean{1,lp};
end
zt=fn_Empirical_conversion_EGARCH111_rt_2_zt_100Dims(ret',x_margin);

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BEGIN: Find 1st tree structure using Kendal's Tau
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
Empir_R_vine_m_MH=zeros(n,n);
Empir_R_vine_m_MH=fn_Empirical_R_vine_m_from_Tau(zt,Empir_R_vine_m_MH
   );

vdirect=cell(num_cop_fam,n,n);
vindirect=cell(num_cop_fam,n,n);
z2=cell(num_cop_fam,n,n);
z1=cell(num_cop_fam,n,n);

Empir_vdirect=cell(n,n);
Empir_vindirect=cell(n,n);
Empir_z2=cell(n,n);
Empir_z1=cell(n,n);
%
R_vine_family=cell(n,n);

for j=1:num_cop_fam
    ct=n;
    for i=1:n
        vdirect{j,n,i}=zt{i,1}';
        ct=ct-1;
    end
end

for i=1:n
    Empir_vdirect{n,i}=zt{i,1}';
end
clear M;
for k=n:-1:1
    for i=n:-1:n
         M(i,k) = max(Empir_R_vine_m_MH(i:n,k));
```

```
    end
end

clear Cmpre_BIC_qmle Cmpre_AIC_qmle Cmpre_loglik_neg_qmle
%
ct_est=1;
for j=1:num_cop_fam
    i=n;
    for k=n-1:-1:1
        R_vine_family{i,k}=family_list{fix_fun(j)};
        % est each cop para of the fam
        z1(j,i,k)=vdirect(j,i,k);
        if M(i,k)==Empir_R_vine_m_MH(i,k)
            z2(j,i,k)=vdirect(j,i,n-M(i,k)+1);
        else
            z2(j,i,k)=vindirect(j,i,n-M(i,k)+1);
        end
        A = []; b = [];
        Aeq = [];  beq = [];
        R_vine_type=fix_fun(j);
        [lb, ub] = fn_R_vine_PCC_bound_para_empir(R_vine_type,1,1);
        lb = lb(2:end);  ub = ub(2:end);
        v=z1{j,i,k};
        u=z2{j,i,k};
        [u,v]=fn_Empirical_PCC_ck_size(u,v);
        if -isinf(lb)
            x0=1;
        else
            x0=lb+0.1; % get initial guess
        end
        x0=[x0 fix_fun(j)];
        [R_vine_x_qmle{j,i,k},loglik_neg_qmle,exitflag,output,lambda,grad,hessian]=...
                         fmincon(@fn_Empirical_QMLE_RVineCopula2,x0,A,b,Aeq,beq,lb,ub);
        sz_para=size(lb,2);
        R_vine_x_qmle{j,i,k}=R_vine_x_qmle{j,i,k}(1:sz_para);
        fprintf('Seq QMLE Est column %i at cop fam %i \n',k,j);
        ct_est=ct_est+1;
        Covar_QMLE{j,i,k}=inv(hessian);
        % cal AIC, BIC: compare cop fam, lower is better fit
        ua=[u;v];
        Cmpre_loglik_neg_qmle(j,i,k) = loglik_neg_qmle;
        Cmpre_AIC_qmle(j,i,k) = (2*Cmpre_loglik_neg_qmle(j,i,k)) + (2*sz_para);
        Cmpre_BIC_qmle(j,i,k) = (2*Cmpre_loglik_neg_qmle(j,i,k)) + (sz_para*log(sz_para));
    end
end
fprintf('Done... \n')
% 1st tree
% select the best candiate: min BIC
%
clear Empir_R_vine_family_mh Empir_R_vine_x_qmle Empir_R_vine_type_MH
clear Empir_qmle_AIC Empir_qmle_BIC Empir_qmle_lglik
for k=n-1:-1:1
```

```
    tp=Cmpre_BIC_qmle(:,n,k);
    pos=find(tp==min(tp));
    if family_list{fix_fun(pos)} == 't'
       if R_vine_x_qmle{pos,n,k}(2) > 30
            Empir_R_vine_family_mh{n,k}=family_list{fix_fun(6)};
            Empir_R_vine_type_MH(n,k)=16;
            Empir_R_vine_x_qmle{n,k}=R_vine_x_qmle{6,n,k}(1:2);
       else
            Empir_R_vine_family_mh{n,k}=family_list{fix_fun(pos)};
            Empir_R_vine_type_MH(n,k)=fix_fun(pos);
            Empir_R_vine_x_qmle{n,k}=R_vine_x_qmle{pos,n,k}(1:2);
       end
    else
        Empir_R_vine_family_mh{n,k}=family_list{fix_fun(pos)};
        Empir_R_vine_type_MH(n,k)=fix_fun(pos);
        Empir_R_vine_x_qmle{n,k}=R_vine_x_qmle{pos,n,k};
    end
    Empir_qmle_AIC(n,k)=Cmpre_AIC_qmle(pos,n,k);
    Empir_qmle_BIC(n,k)=Cmpre_BIC_qmle(pos,n,k);
    Empir_qmle_lglik(n,k)=-Cmpre_loglik_neg_qmle(pos,n,k);
end

% fit data to copula for 1st tree
vdirect=cell(n,n);
vindirect=cell(n,n);
z2=cell(n,n);
z1=cell(n,n);
Empir_vdirect=cell(n,n);
Empir_vindirect=cell(n,n);
Empir_z2=cell(n,n);
Empir_z1=cell(n,n);
Num_fun=n*(n-1)/2;

for i=1:n
    vdirect{n,i}=zt{i,1}';
end

% 1st tree: cal each dependance pair for all candidate cop list
% and pick cop fam best fit to the data
ct_est=1;
acted_rate=0;
series_x0_MH_all=cell(1,1);
store_theta_MH_R_vine=cell(1,1);
R_vine_x_MH_all=cell(1,1);
R_vine_x_MH_acpted=cell(1,1);
R_vine_x_MH_acpted_std=cell(1,1);

% -------------------------------------------------------------------------------
% performs tree n, parallelly
% -------------------------------------------------------------------------------

i=n;
```

```
R_vine_family=Empir_R_vine_family_mh;
%number of burning MH replications
nburn=3000; % 3000
%number of included MH replications
nrep=20000; % 20000
tic;
for k=1:n-1
    z1(i,k)=vdirect(i,k);
    if M(i,k)==Empir_R_vine_m_MH(i,k)
        z2(i,k)=vdirect(i,n-M(i,k)+1);
    else
        z2(i,k)=vindirect(i,n-M(i,k)+1);
    end
end
parfor k=1:n-1
    A = []; b = [];
    Aeq = [];  beq = [];
    R_vine_type=Empir_R_vine_type_MH(i,k);
    [lb, ub] = fn_R_vine_PCC_bound_para_empir(R_vine_type,1,1);
    lb = lb(2:end);  ub = ub(2:end);
    v=z1{i,k};
    u=z2{i,k};
    ua=[u;v];
    thetdraw=Empir_R_vine_x_qmle{i,k}; % get initial guess
    %   PERFORM RW-MH ESTIMATION METHOD
    % ************************************************
    % STEP: Main Seq-MH to get para posterior distn
    %         using covar above
    % ************************************************
    store_theta_MH_R_vine_int =zeros(1,sz);
    C=1;
    ctt=0;
    x0_MH=ones(nrep-nburn,size(thetdraw,2));
    x0_all_bk=Empir_R_vine_x_qmle{i,k}; % back up R_vine_x
    nswitch = 0;
    lg=size(thetdraw,2);
    v=zeros(1,lg)+0.01;
    sigma=diag(v);
    % --------------------------------------------------------------
    % performs tree n: rw-MH
    %
       -----------------------------------------------------------------

    for irep=1:nrep+nburn

        brk_condition=0; % low acceptance rate break condition
        accept=0;
        num_para=size(thetdraw,2);
        kk = size(thetdraw,2);
        ck_bound=kk*2;
        ct=0;
        thmean=zeros(1,kk);
        th2mo=zeros(kk,kk);
        sigma=C.*sigma;

```

```
        llike1=-fn_RVine_PCC_Loglik(thetdraw,ua,i,k,R_vine_family);
        thetcan = thetdraw + norm_rnd(sigma)';
        ck_in_bound=sum(thetcan > lb) + sum(thetcan<ub);

        % log likelihood of copula fam: a candidate draw
        llike2=-fn_RVine_PCC_Loglik(thetcan,ua,i,k,R_vine_family);
        %log of acceptance probability
        laccprob = llike2 - llike1;

        %now accept candidate with appropriate probability
        if laccprob > log(rand) && ~isinf(laccprob)
            thetdraw=thetcan;
            accept = 1;
            nswitch = nswitch+1;
            if irep>nburn
                ctt=ctt+1;
                store_theta_MH_R_vine_int = [
                    store_theta_MH_R_vine_int; thetcan];
            end
        end
        Acpt_rate_new=nswitch/irep;
        if irep>nburn
            x0_MH(irep-nburn,:)=thetdraw;
        end

    acted_rate=[acted_rate ctt/nrep];
    series_x0_MH_all=[series_x0_MH_all {x0_MH}];
    store_theta_MH_R_vine=[store_theta_MH_R_vine {
        store_theta_MH_R_vine_int(2:5:end,:)}];
    R_vine_x_MH_acpted=[R_vine_x_MH_acpted {mean(
        store_theta_MH_R_vine_int(2:5:end,:))}];
end

R_vine_x_MH_acpted_matrix(n,:)=R_vine_x_MH_acpted(2:end);
acted_rate_mat(n,:)=acted_rate(2:end);

%
   --------------------------------------------------------------------------
% perform
% gen 1st tree empirical of
% z1 z2 vdirect invdirect
%
   --------------------------------------------------------------------------

i=n;
for k=n-1:-1:1
    v=z1{i,k};
    u=z2{i,k};
    fam=Empir_R_vine_family_mh{i,k};
    para=R_vine_x_MH_acpted_matrix{i,k};
    vdirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);

    u=z1{i,k};
    v=z2{i,k};
```

```
    vindirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);
end

%
  --------------------------------------------------------------------------
% performs tree n-1 until last tree, tree 2
%
  --------------------------------------------------------------------------

global zt
global i
global k
global R_vine_family
global u
global v
for k=n-2:-1:1 % colum operation
    for i=n-1:-1:k+1 % row operation
        tic;
        % update Empir_R-vine-m and M tree (maximum spanning tree)
        % perform qmle: find the best cop fam by min BIC.
        [Empir_R_vine_m_MH,Empir_R_vine_x_qmle,Empir_R_vine_family_mh
            ,Empir_R_vine_type_MH,M]=...
            fn_find_r_vine_structure_by_QMLE(Empir_R_vine_type_MH,...
              Empir_R_vine_m_MH,z1,z2,M,vdirect,vindirect,n,
                 Empir_R_vine_family_mh,family_list,
                 Empir_R_vine_x_qmle);
        % find all candidate from empir cop fam
        z1(i,k)=vdirect(i,k);
        if M(i,k)==Empir_R_vine_m_MH(i,k)
            z2(i,k)=vdirect(i,n-M(i,k)+1);
        else
            z2(i,k)=vindirect(i,n-M(i,k)+1);
        end
        v=z1{i,k};
        u=z2{i,k};
        ua=[u;v];


        A = []; b = [];
        Aeq = [];  beq = [];
        R_vine_type=Empir_R_vine_type_MH(i,k);
        [lb, ub] = fn_R_vine_PCC_bound_para_empir(R_vine_type,1,1);
        lb = lb(2:end);  ub = ub(2:end);
        thetdraw=Empir_R_vine_x_qmle{i,k}; % get initial guess
    %   PERFORM RW-MH ESTIMATION METHOD
        sz=size(thetdraw,2);
        % ****************************************
        % STEP   : Main Seq-MH to get para posterior distn
        %          using covar above
        % ****************************************
        store_theta_MH_R_vine_int =zeros(1,sz);
        C=1;
        ctt=0;
        x0_MH=ones(nrep-nburn,size(thetdraw,2));
```

```
        x0_all_bk=Empir_R_vine_x_qmle{i,k}; % back up R_vine_x from seq qmle
        nswitch = 0;
        lg=size(thetdraw,2);
        v=zeros(1,lg)+0.01;
        sigma=diag(v);

        % ----------------------------------------------------
        % performs rw-MH tree n-1 until last tree, tree 2
        % --------------------------------------------------

        for irep=1:nrep+nburn

            brk_condition=0; % low acceptance rate break condition
            accept=0;
            num_para=size(thetdraw,2);
            kk = size(thetdraw,2);
            ck_bound=kk*2;
            ct=0;
            thmean=zeros(1,kk);
            th2mo=zeros(kk,kk);
            sigma=C.*sigma;
            llike1=-fn_RVine_PCC_Loglik(thetdraw,ua,i,k,Empir_R_vine_family_mh);
            thetcan = thetdraw + norm_rnd(sigma)';
            ck_in_bound=sum(thetcan > lb) + sum(thetcan<ub);
            % log likelihood of copula fam: a candidate draw
            llike2=-fn_RVine_PCC_Loglik(thetcan,ua,i,k,Empir_R_vine_family_mh
            %log of acceptance probability
            laccprob = llike2 - llike1;
            %now accept candidate with appropriate probability
            if laccprob > log(rand) && ~isinf(laccprob)
                thetdraw=thetcan;
                accept = 1;
                nswitch = nswitch+1;
                if irep>nburn
                    ctt=ctt+1;
                    store_theta_MH_R_vine_int = [store_theta_MH_R_vine_int; thetcan];
                end
            end
            Acpt_rate_new=nswitch/irep;
            if irep>nburn
                x0_MH(irep-nburn,:)=thetdraw;
            end

        end

% -------------------------------------------------------------------------
% end of rw-MH of tree n-1 until last tree, tree 2
% -------------------------------------------------------------------------
```

```

    acted_rate=[acted_rate ctt/nrep];
    acted_rate_mat(i,k)=ctt/nrep;
    series_x0_MH_all=[series_x0_MH_all {x0_MH}];
    store_theta_MH_R_vine=[store_theta_MH_R_vine {
       store_theta_MH_R_vine_int(2:5:end,:)}];
    R_vine_x_MH_all=[R_vine_x_MH_all {mean(x0_MH)}];
    R_vine_x_MH_acpted=[R_vine_x_MH_acpted {mean(
       store_theta_MH_R_vine_int(2:5:end,:))}];
    R_vine_x_MH_acpted_matrix{i,k}=mean(store_theta_MH_R_vine_int
       (2:5:end,:),1);

    v=z1{i,k};
    u=z2{i,k};
    fam=Empir_R_vine_family_mh{i,k};
    para=R_vine_x_MH_acpted_matrix{i,k};
    vdirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);

    u=z1{i,k};
    v=z2{i,k};
    vindirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);

    para_out=R_vine_x_MH_acpted_matrix{i,k};
    end
end
```

D.2.2 The Variational Bayes with Data Augmentation Type 1 Algorithm of the R-vine-ICAPM-EGARCH-Mixture Model (The IFM Method)

The IFM Step 1

```
%-------------  Using VBDA on multivariate time series
   ------------------%
% data augmentation
%
% Developer: Rewat Khanthaporn
% Date 12 Sep 2021
% Email: raysuge044@hotmail.com
%
% This mod is based on 2 main paper as follows
% 2012, Smith, Estimation of Copula Models With Discrete Margins via
   Bayesian Data Augmentation
% 2019, Maya and Smith, Variational Bayes Estimation of Discrete-
   Margined Copula Models With Application to Time Series
%
% there are three type modify VB according to the 2 vars setting up
% VA variable is data augmentation method to the VB
%   VA = DA = 1 and va_0 = 0
%       mean no data augmentation type ==> q_u evaluated
%   VA = DA = 1 and va_0 = 1
%       mean no data augmentation type ==> not q_u evaluated
%   VA = DA = 2 and va_0 = 0
%       mean data augmentation type ==> q_u evaluated
```

```

clear
clc
k = 1;                    %Number of factors for covariance matrix
S = 300;                   %Number of evaluations to compute
    gradient
nVB = 20000;                %Number of VB steps

% input area
% ret: a metrix of returns (T by n); T=time, n= number of asset.
% initial gauss, i.e., from QMLE method
    % x_margin
    % covar
nn=size(ret,2);
%
va_0=0;        % set 1=yes incase of qu=0 mean no data augmentation,
    see in fun: FFVB_Est_all_EGARCH_gradient_computeM.m
VA = 1;        % Variational Bayes approach for estimation
zt1=0;
parfor i=1:n
        y=ret(:,i);
        x_pre=x_margin(i,:);
        r = size(y,2);
        VBDAobjA(i) = Empirical_FFVB_Est_all_EGARCH_VBDAfit_multi...
                             (y,p,k,S,nVB,VA,x_pre,va_0,i);%Using VBDA
end

clear zt;
parfor lp=1:n
    x_margin(lp,:)=VBDAobjA(lp).gamma_mean;
    z=VBDAobjA(lp).zt_mean;
    x1=x_margin(lp,:);
    xil_1=x1(1);
    xir_1=x1(2);
    betar_1=(1-xir_1)*(1-(2*xir_1))^0.5; %  beta
    betal_1=(1-xil_1)*(1-(2*xil_1))^0.5; %  beta
    zt{lp,1}=fn_CDF_Conversion_mix_Nor_GPD(z,betar_1,xir_1,betal_1,
        xil_1);
end
```

The IFM Step 2

```
% VBDA1 Algorithm in R-vine-ICAPM-EGARCH-Mixture model
% Date 9 December 2022
% If you find any bugs, feel free to email me.
% By Rewat Khanthaporn
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

% copula function
% Matlab toolbox:
%       - 'Clayton'      = 1    used
%       - 'Frank'        = 2    used
%       - 'Gumbel'(EV)   = 3    used
% My coding:  1-parameter Archimedean copula
%       - 'Joe'/'B5'     = 4    used
%       - 'Galambos'(EV) = 5    used
```

```
% 2-parameters copula
%        - 'BB1'           = 6    used
%        - 'BB2'           = 7    not used
%        - 'BB3'           = 8    used
%        - 'BB4'           = 9    used
%        - 'BB5'           = 10   not used
%        - 'BB6'           = 11   used
%        - 'BB7'           = 12   used
%        - 'BB8'           = 13   not used
%        - 'BB9','Crowder' = 14   not used
%        - 'BB10'          = 15   used
% Elliptical copula
%        - 'Gaussian'      =16    used
%        - 't'             =17    used
clear; clc;

% input area

% ret: a metrix of returns (T by n); T=time, n= number of asset.
% series_theta_MH_EGARCH_Mean: out of step 1 to be input here.

global family_list
para_num = [ 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2];
family_list= ["Clayton","Frank","Gumbel","B5","Galambos" ...
             ,"BB1","BB2","BB3","BB4","BB5","BB6","BB7","BB8"...
             ,"BB9","BB10","Gaussian","t"];
global fix_fun
fix_fun=[1:5,16,17,6,8,9,11,12,15];
num_cop_fam=length(fix_fun);

global zt
global i
global k
global u
global v

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BEGIN: convert ret to unif(0,1)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
parfor lp=1:n
    x_margin(lp,:)=series_theta_MH_EGARCH_Mean{1,lp};
end
zt=fn_Empirical_conversion_EGARCH111_rt_2_zt_100Dims(ret',x_margin);

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BEGIN: Find 1st tree structure using Kendal's Tau
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Empir_R_vine_m_vbda1=zeros(n,n);
Empir_R_vine_m_vbda1=fn_Empirical_R_vine_m_from_Tau(zt,
   Empir_R_vine_m_vbda1);

% fit data to copula for 1st tree
% load('Empir_R_vine_m')
vdirect=cell(num_cop_fam,n,n);
vindirect=cell(num_cop_fam,n,n);
```

```
z2=cell(num_cop_fam,n,n);
z1=cell(num_cop_fam,n,n);

Empir_vdirect=cell(n,n);
Empir_vindirect=cell(n,n);
Empir_z2=cell(n,n);
Empir_z1=cell(n,n);
%
R_vine_family=cell(n,n);
for j=1:num_cop_fam
    ct=n;
    for i=1:n
        vdirect{j,n,i}=zt{i,1}';
        ct=ct-1;
    end
end

for i=1:n
    Empir_vdirect{n,i}=zt{i,1}';
end
clear M;
for k=n:-1:1
    for i=n:-1:n
         M(i,k) = max(Empir_R_vine_m_vbda1(i:n,k));
    end
end

%%
clear Cmpre_BIC_qmle Cmpre_AIC_qmle Cmpre_loglik_neg_qmle
%
ct_est=1;
for j=1:num_cop_fam
    i=n;
    for k=n-1:-1:1
        z1(j,i,k)=vdirect(j,i,k);
        if M(i,k)==Empir_R_vine_m_vbda1(i,k)
            z2(j,i,k)=vdirect(j,i,n-M(i,k)+1);
        else
            z2(j,i,k)=vindirect(j,i,n-M(i,k)+1);
        end
        A = []; b = [];
        Aeq = [];  beq = [];
        R_vine_type=fix_fun(j);
        [lb, ub] = fn_R_vine_PCC_bound_para_empir(R_vine_type,1,1);
        lb = lb(2:end);  ub = ub(2:end);
        v=z1{j,i,k};
        u=z2{j,i,k};
        [u,v]=fn_Empirical_PCC_ck_size(u,v);
        if -isinf(lb)
            x0=1;
        else
            x0=lb+0.1; % get initial guess
        end
        x0=[x0 fix_fun(j)];
        [R_vine_x_qmle{j,i,k},loglik_neg_qmle,exitflag,output,lambda,
           grad,hessian]=...
```

```
                        fmincon(@fn_Empirical_QMLE_RVineCopula2,x0,A
                            ,b,Aeq,beq,lb,ub);
        sz_para=size(lb,2);
        R_vine_x_qmle{j,i,k}=R_vine_x_qmle{j,i,k}(1:sz_para);
        ct_est=ct_est+1;
        Covar_QMLE{j,i,k}=inv(hessian);
        ua=[u;v];
        Cmpre_loglik_neg_qmle(j,i,k) = loglik_neg_qmle;
        Cmpre_AIC_qmle(j,i,k) = (2*Cmpre_loglik_neg_qmle(j,i,k)) +
            (2*sz_para);
        Cmpre_BIC_qmle(j,i,k) = (2*Cmpre_loglik_neg_qmle(j,i,k)) + (
            sz_para*log(sz_para));
    end
end
%
% 1st tree
% select the best candiate: min BIC
clear Empir_R_vine_family_vbda1 Empir_R_vine_x_qmle
    Empir_R_vine_type_vbda1
clear Empir_qmle_AIC Empir_qmle_BIC Empir_qmle_lglik
for k=n-1:-1:1
    tp=Cmpre_BIC_qmle(:,n,k);
    pos=find(tp==min(tp));
    if family_list{fix_fun(pos)} == 't'
       if R_vine_x_qmle{pos,n,k}(2) > 30
            Empir_R_vine_family_vbda1{n,k}=family_list{fix_fun(6)};
            Empir_R_vine_type_vbda1(n,k)=16;
            Empir_R_vine_x_qmle{n,k}=R_vine_x_qmle{6,n,k}(1:2);
       else
            Empir_R_vine_family_vbda1{n,k}=family_list{fix_fun(pos)};
            Empir_R_vine_type_vbda1(n,k)=fix_fun(pos);
            Empir_R_vine_x_qmle{n,k}=R_vine_x_qmle{pos,n,k}(1:2);
       end
    else
        Empir_R_vine_family_vbda1{n,k}=family_list{fix_fun(pos)};
        Empir_R_vine_type_vbda1(n,k)=fix_fun(pos);
        Empir_R_vine_x_qmle{n,k}=R_vine_x_qmle{pos,n,k};
    end
    Empir_qmle_AIC(n,k)=Cmpre_AIC_qmle(pos,n,k);
    Empir_qmle_BIC(n,k)=Cmpre_BIC_qmle(pos,n,k);
    Empir_qmle_lglik(n,k)=-Cmpre_loglik_neg_qmle(pos,n,k);
end

% fit data to copula for 1st tree
vdirect=cell(n,n);
vindirect=cell(n,n);
z2=cell(n,n);
z1=cell(n,n);

Empir_vdirect=cell(n,n);
Empir_vindirect=cell(n,n);
Empir_z2=cell(n,n);
Empir_z1=cell(n,n);
Num_fun=n*(n-1)/2;

for i=1:n
```

```
    vdirect{n,i}=zt{i,1}';
end

ct_est=1;
series_x0_vbda1_all=cell(1,1);
store_theta_vbda1_R_vine=cell(1,1);
R_vine_x_vbda1_all=cell(1,1);
R_vine_x_vbda1_acpted=cell(1,1);
R_vine_x_vbda1_acpted_std=cell(1,1);

%
    ------------------------------------------------------------------------
% performs tree n, parallelly
%
    ------------------------------------------------------------------------

i=n;
for k=1:n-1
    z1(i,k)=vdirect(i,k);
    if M(i,k)==Empir_R_vine_m_vbda1(i,k)
        z2(i,k)=vdirect(i,n-M(i,k)+1);
    else
        z2(i,k)=vindirect(i,n-M(i,k)+1);
    end
end
clear VBDAobjA;
va_0=0;       % set 1=yes incase of qu=0 mean no data augmentation,
    see in fun: FFVB_Est_all_EGARCH_gradient_computeM.m
VA = 1;       % Variational Bayes approach for estimation
kk = 1;       %Number of factors for covariance matrix
S = 100;      %Number of evaluations to compute gradient
nVB = 10000; %Number of VB steps
tw=25;
clear R_vine_x_vbda1_ested_para;
parfor k=1:n-1

    A = []; b = [];
    Aeq = [];  beq = [];
    R_vine_typ=Empir_R_vine_type_vbda1(i,k);
    [lb, ub] = fn_R_vine_PCC_bound_para_empir(R_vine_typ,1,1);
    lb = lb(2:end);  ub = ub(2:end);
    v=z1{i,k};
    u=z2{i,k};
    num_para=size(lb,2);
    sigma=eye(num_para)*0.1;
    family = Empir_R_vine_family_vbda1{i,k};       %Copula family for
        pair-copula components
    x_pre=Empir_R_vine_x_qmle{i,k};
    y=[u; v]';
    y = gpuArray(y);
    VBDAobjA(n,k) = Empirical_FFVB_New_VBDAfit_multi(y,family,kk,S,
       nVB,VA,va_0,x_pre,sigma);%Using VBDA
end
```

```
%
    ----------------------------------------------------------------------------

% performs
% gen 1st tree empirical of
% z1 z2 vdirect invdirect
%
    ----------------------------------------------------------------------------


i=n;
for k=n-1:-1:1
    v=z1{i,k};
    u=z2{i,k};
    fam=Empir_R_vine_family_vbda1{i,k};
    para=R_vine_x_vbda1_ested_para{i,k};
    vdirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);

    u=z1{i,k};
    v=z2{i,k};
    vindirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);
end

%
    ----------------------------------------------------------------------------

% performs tree n-1 until last tree, tree 2
%
    ----------------------------------------------------------------------------

%
global zt
global i
global k
global u
global v
for k=n-2:-1:1 % colum operation
    for i=n-1:-1:k+1 % row operation
        % update Empir_R-vine-m and M tree (maximum spanning tree)
        % perform qmle: find the best cop fam by min BIC.
        [Empir_R_vine_m_vbda1,Empir_R_vine_x_qmle,...
         Empir_R_vine_family_vbda1,Empir_R_vine_type_vbda1,M]=...
                fn_find_r_vine_structure_by_QMLE(...
                      Empir_R_vine_type_vbda1,...
                      Empir_R_vine_m_vbda1,z1,z2,...
                      M,vdirect,vindirect,n,...
                      Empir_R_vine_family_vbda1,...
                      family_list,Empir_R_vine_x_qmle);

        z1(i,k)=vdirect(i,k);
        if M(i,k)==Empir_R_vine_m_vbda1(i,k)
            z2(i,k)=vdirect(i,n-M(i,k)+1);
        else
            z2(i,k)=vindirect(i,n-M(i,k)+1);
        end

```

```
        A = []; b = [];
        Aeq = [];  beq = [];
        R_vine_typ=Empir_R_vine_type_vbda1(i,k);
        [lb, ub] = fn_R_vine_PCC_bound_para_empir(R_vine_typ,1,1);
        lb = lb(2:end);  ub = ub(2:end);
        v=z1{i,k};
        u=z2{i,k};
        num_para=size(lb,2);
        sigma=eye(num_para)*0.1;

        family = Empir_R_vine_family_vbda1{i,k};  %Copula family for
            pair-copula components
        x_pre=Empir_R_vine_x_qmle{i,k};
        y=[u; v]';
        VBDAobjA(i,k) = Empirical_FFVB_New_VBDAfit_multi(y,family,p,
            kk,S,nVB,VA,va_0,x_pre,sigma);%Using VBDA
        R_vine_x_vbda1_ested_para{i,k}=VBDAobjA(i,k).gamma_mean;
        R_vine_x_vbda1_last_nVB(i,k)=VBDAobjA(i,k).t;
        R_vine_x_vbda1_time(i,k)=VBDAobjA(i,k).time;

        v=z1{i,k};
        u=z2{i,k};
        fam=Empir_R_vine_family_vbda1{i,k};
        para=R_vine_x_vbda1_ested_para{i,k};
        vdirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);

        u=z1{i,k};
        v=z2{i,k};
        vindirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);
    end
end
```

D.2.3 The Miscellaneous Functions

```
function z=fn_CDF_Conversion_mix_Nor_GPD(x,betar,xir,betal,xil)

sz=size(x,1);
q_R=norminv(0.9);   % upper quantial at 0.9, this can be changed
q_L=norminv(0.1);   % lower quantial at 0.1, this can be changed
for i=1:sz
    if x(i) >= q_R % right tail
        z(i,1) = -(x(i)*xir/betar + 1 )^(-1/xir) + 1 ;
    elseif x(i) <= q_L % left tail
        z(i,1) = (-(xil/betal*x(i)) +1)^(-1/xil);
    else
        z(i,1)=normcdf(x(i));
    end
end

z=U_transf(z);
```

```
function [R_vine_x_qmle_can,loglik_neg_qmle,sz_para]=...
                            fn_Empir_fmincon_r_vine(aaa)
```

```
A = []; b = [];
Aeq = [];  beq = [];
global fix_fun
global zt
global i
global k
global R_vine_family
global u
global v
R_vine_typ=fix_fun(aaa);
[lb, ub] = fn_R_vine_PCC_bound_para_empir(R_vine_typ,1,1);
lb = lb(2:end);  ub = ub(2:end);
sz_para=size(lb,2);
if -isinf(lb)
    x0=1;
else
    x0=lb+0.1; % get initial guess
end
x0=[x0 fix_fun(aaa)];
options = optimoptions('fmincon','Display','off');
[R_vine_x_qmle_can,loglik_neg_qmle,~,~,~,~,~]=...
  fmincon(@fn_Empirical_QMLE_RVineCopula2,x0,A,b,Aeq,beq,lb,ub,[],options);
```

```
function zt=fn_Empirical_conversion_EGARCH111_rt_2_zt_100Dims(rt,x_margin1)

n=size(rt,1);
zt=cell(n,1);
for i=1:n
    x1=x_margin1(i,:);
    rt_run=rt(i,1:end)';
    T=size(rt_run,1);
    xil_1=x1(1);
    xir_1=x1(2);
    lambda1_1=x1(3);
    lambda2_1=x1(4);
    omega_1=x1(5);
    gam_1=x1(6);
    delta_1=x1(7);
    nu_1=x1(8);
    betar_1=(1-xir_1)*(1-(2*xir_1))^0.5; %  beta
    betal_1=(1-xil_1)*(1-(2*xil_1))^0.5; %  beta
    z1_1=zeros(T,1); hts_1=zeros(T,1);
    hts_1(1,1)=omega_1;
    z1_1(1,1)=(rt_run(1,1) - lambda1_1 - (lambda2_1*exp(hts_1(1,1))))/sqrt(exp(hts_1(1,1)));
    for t=2:T
        hts_1(t,1)=omega_1 + gam_1*z1_1(t-1,1) + delta_1*abs(z1_1(t-1,1)) + nu_1*(hts_1(t-1,1));
        z1_1(t,1)=(rt_run(t,1) - lambda1_1 - (lambda2_1*exp(hts_1(t,1))))/sqrt(exp(hts_1(t,1)));
    end

    % fun to convert z1 above to unif
```

```
    zt{i,1}=fn_CDF_Conversion_mix_Nor_GPD(z1_1,betar_1,xir_1,betal_1,
        xil_1);
end
```

```
function [Empir_R_vine_m,M]=fn_Empirical_m_finding(i,k,Empir_R_vine_m
    ,M,Empir_vdirect,Empir_vindirect)

global u
global v
n=size(Empir_R_vine_m,1);
set1=[Empir_R_vine_m(i+1:n,k)];
for a=k+1:i
    set2(a,:)=[Empir_R_vine_m(a,a) Empir_R_vine_m(i+1:n,a)'];
    set_tp=set2(a,:);
    m_can=setdiff(set_tp,set1);
    if size(m_can,2) == 1 % meet promixity con
        can_list(a)=1;
    else
        can_list(a)=0;
    end
end

if sum(can_list) == 1 % no choice
    pos=find(can_list==1);
    set_tp=set2(pos(1),:);
    Empir_R_vine_m(i,k)=setdiff(set_tp,set1);
else % select candidate as the highest empirical tau
    for a=k+1:i % loop for all candiate
        if can_list(a) == 1
            set_tp=set2(a,:);
            m_can=setdiff(set_tp,set1);
            Empir_R_vine_m(i,k)=m_can;
            v=Empir_vdirect{i,k};
            M(i,k)=max([m_can,Empir_R_vine_m(i+1:n,k)']);
            if M(i,k)==Empir_R_vine_m(i,k)
                u=Empir_vdirect{i,n-M(i,k)+1};
            else
                u=Empir_vindirect{i,n-M(i,k)+1};
            end
            [u,v]=fn_Empirical_PCC_ck_size(u,v);
            r = corr([u' v'],'type','Kendall');
            empir_r_t(a)=fn_cal_cor_tcop;
            empir_tau(a)=r(2,1);
        else
            empir_tau(a)=0;
        end
    end
    empir_r_t_abs=abs(empir_r_t);
    pos1=find(empir_r_t_abs==max(abs(empir_r_t_abs)));

    empir_tau_abs=abs(empir_tau);
    pos2=find(empir_tau_abs==max(abs(empir_tau_abs)));

    pos=pos2(1);
    if pos1(1) ~= pos2(1)
        pos=pos1(1);
```

```
    end

    set_tp=set2(pos,:);
    Empir_R_vine_m(i,k)=setdiff(set_tp,set1);
end
M(i,k)=max(Empir_R_vine_m(i:n,k));
```

```
function [u,v]=fn_Empirical_PCC_ck_size(u,v)
if size(u,2) < size(v,2)
    l1=size(u,2);
    l2=size(v,2);
    v=v(1,l2-l1+1:l2);
elseif size(u,2) > size(v,2)
    l2=size(u,2);
    l1=size(v,2);
    u=u(1,l2-l1+1:l2);
end
```

```
function Empir_R_vine_m=fn_Empirical_R_vine_m_from_Tau(zt,
   Empir_R_vine_m)
clear r;
n=size(zt,1);
for i=1:n
    fprintf('Cal empirical Tau at %i out of %i\n',i,n)
    parfor j=i:n
        if i==j
            r(i,j)=1;
        else
            if size(zt{i,1},1) == size(zt{j,1},1)
                z=[zt{i,1} zt{j,1}]; % colum vector
            elseif size(zt{i,1},1) < size(zt{j,1},1)
                l1=size(zt{i,1},1);
                l2=size(zt{j,1},1);
                z_cut=zt{j,1}(l2-l1+1:l2,1);
                z=[zt{i,1} z_cut];
            elseif size(zt{i,1},1) > size(zt{j,1},1)
                l2=size(zt{i,1},1);
                l1=size(zt{j,1},1);
                z_cut=zt{i,1}(l2-l1+1:l2,1);
                z=[zt{j,1} z_cut];
            end
            r_temp=corr(z,'type','Kendall');
            r(i,j)=r_temp(2,1);
        end
    end
    for j=i:n
        r(j,i)=r(i,j);
    end
end

ct_time=1;
pair_pos_1=0;
pair_pos_2=0;
for i=n:-1:2
    pos_max=2;
```

```
    Empir_R_vine_m(ct_time,ct_time)=i;
    pair_pos_1=[pair_pos_1 i];
    r_ab=abs(r(i,:));
    f=sort(r_ab,'descend');
    f_2nd_max=f(pos_max);
    pos=find(r_ab==f_2nd_max);
    pair_pos_2=[pair_pos_2 pos];
    ok=fn_dup_pair(pair_pos_1,pair_pos_2);
    Empir_R_vine_m(n,ct_time)=pos;

    while ok==0
        pos_max=pos_max+1;
        f_next_max=f(pos_max);
        pos=find(r_ab==f_next_max);
        pair_pos_2(end)=pos(1);
        ok=fn_dup_pair(pair_pos_1,pair_pos_2);
        Empir_R_vine_m(n,ct_time)=pos(1);
    end

    ct_time=ct_time+1;
    if i==2
        Empir_R_vine_m(n,n)=pos;
    end
end
```

```
function [R_vine_m,Empir_R_vine_x_qmle,Empir_R_vine_family,
    Empir_R_vine_type,M]=...
            fn_find_r_vine_structure_by_QMLE(Empir_R_vine_type,...
              R_vine_m,z1,z2,M,vdirect,vindirect,n,
                  Empir_R_vine_family,family_list,Empir_R_vine_x_qmle)

global fix_fun
global zt
global i
global k
global u
global v
aa=1:13;
% update Empir_R-vine-m and M tree (maximum spanning tree)
% from highest empirical tau
[R_vine_m,M]=fn_Empirical_m_finding(i,k,R_vine_m,M,...
                                    vdirect,vindirect);
% find all candidate from empir cop fam
z1(i,k)=vdirect(i,k);
if M(i,k)==R_vine_m(i,k)
    z2(i,k)=vdirect(i,n-M(i,k)+1);
else
    z2(i,k)=vindirect(i,n-M(i,k)+1);
end
clear Cmpre_BIC_qmle1 R_vine_x_qmle_can loglik_neg_qmle1
    Cmpre_AIC_qmle1
v=z1{i,k};
u=z2{i,k};
[u,v]=fn_Empirical_PCC_ck_size(u,v);
% est all can cop fam and BIC
```

```
lg=length(fix_fun);
for j=1:lg
    [R_vine_x_qmle_can{j},loglik_neg_qmle1(j),sz_para]=...
                                fn_Empir_fmincon_r_vine(aa(1,j));
    R_vine_x_qmle_can{j}=R_vine_x_qmle_can{j}(1:end-1);
    Cmpre_AIC_qmle1(j) = (2*loglik_neg_qmle1(j)) + (2*sz_para);
    Cmpre_BIC_qmle1(j) = (2*loglik_neg_qmle1(j)) + (sz_para*log(
        sz_para));
end
pos=find(Cmpre_BIC_qmle1==min(Cmpre_BIC_qmle1));

if family_list{fix_fun(pos)} == 't'
   if R_vine_x_qmle_can{pos}(2) > 30
        Empir_R_vine_family{i,k}=family_list{fix_fun(6)};
        Empir_R_vine_type(i,k)=16;
        Empir_R_vine_x_qmle{i,k}=R_vine_x_qmle_can{6};
   else
        Empir_R_vine_family{i,k}=family_list{fix_fun(pos)};
        Empir_R_vine_type(i,k)=fix_fun(pos);
        Empir_R_vine_x_qmle{i,k}=R_vine_x_qmle_can{pos};
   end
else
    Empir_R_vine_family{i,k}=family_list{fix_fun(pos)};
    Empir_R_vine_type(i,k)=fix_fun(pos);
    Empir_R_vine_x_qmle{i,k}=R_vine_x_qmle_can{pos};
end
Empir_qmle_AIC(i,k)=Cmpre_AIC_qmle1(pos);
Empir_qmle_BIC(i,k)=Cmpre_BIC_qmle1(pos);
Empir_qmle_lglik(i,k)=-loglik_neg_qmle1(pos);
```

```
function f=fn_MH_Blc4_EGARCH_Loglik(yt,zeta_all)

xil=zeta_all(1,1);
xir=zeta_all(1,2);
lambda1=zeta_all(1,3);
lambda2=zeta_all(1,4);
omega=zeta_all(1,5);
gam=zeta_all(1,6);
delta=zeta_all(1,7);
nu=zeta_all(1,8);
%
p_next=[0.8;0.1;0.1];

betar=(1-xir)*(1-(2*xir))^0.5; %  beta
betal=(1-xil).*(1-(2.*xil)).^0.5; %  beta
yt=yt';
n=size(yt,1);
hts1= zeros(n,1);
z11 = zeros(n,1);
hts1(1,1)=0.01;
z11(1,1)=(yt(1,1) - lambda1 - (lambda2.*exp(hts1(1,1)))))./sqrt(exp(
    hts1(1,1)));
for t=2:n
    hts1(t,1)=omega + gam.*z11(t-1,1) + delta.*abs(z11(t-1,1)) + nu
        .*(hts1(t-1,1));
    z11(t,1)=(yt(t,1) - lambda1 - (lambda2.*exp(hts1(t,1))))./...
```

```
                                            sqrt(exp(hts1(t,1)));
end

et=z11.*sqrt(exp(hts1));
lik(:,1)= pdf('normal',et,0,sqrt(exp(hts1)));
lik(:,2)=pdf('Generalized Pareto',et,xir,betar.*sqrt(exp(hts1)),0);
lik(:,3)=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts1)),0);
lik_w=lik.*p_next';
f=sum(lik_w,2);
f= sum(log(f));
```

```
function [lb, ub] = fn_R_vine_PCC_bound_para_empir(fam_rnd,i,j)
lb=999; ub=999;
switch fam_rnd(i,j)
    case 16 %{'Gaussian'}
        lb = [lb -0.99]; % rho; linear correlation
        ub = [ub 0.99];
    case 17 %{'t'}
        lb = [lb -0.99 3]; % rho nu
        ub = [ub  0.99 60];
    case 1 % 'Clayton' %
        lb = [lb 0]; % delta zeta
        ub = [ub inf ];
    case 2 % 'Frank' %
        lb = [lb -inf ]; % delta zeta
        ub = [ub inf ];
    case 3 % 'gumbel' %
        lb = [lb 1 ]; % delta zeta
        ub = [ub inf ];
    case 4 % 'Joe'/'B5'
        lb = [lb 1 ]; % delta zeta
        ub = [ub inf ];
    case 5 % 'Galambos' % [0,inf)
        lb = [lb 0 ]; % delta zeta
        ub = [ub inf];
    case 6 % 'BB1' % (Joe and Hu, 1996)
        lb = [lb 1 0]; % delta zeta
        ub = [ub inf inf];
    case 7 %'BB2' % (Joe and Hu, 1996) (Joe book page 194)
        lb = [lb 0 0]; % delta zeta
        ub = [ub inf inf];
    case 8 %'BB3' % (Joe and Hu, 1996) (Joe book page 195)
        lb = [lb 0 1]; % delta zeta
        ub = [ub inf inf];
    case 9 %'BB4' % (Joe and Hu, 1996) (Joe book page 197)
        lb = [lb 0 0]; % delta zeta
        ub = [ub inf inf];
    case 10 %'BB5' % (Joe and Hu, 1996) (Joe book page 199)
        lb = [lb 0 1]; % delta zeta
        ub = [ub inf inf];
    case 11 %'BB6' % (Joe and Hu, 1996) (Joe book page 200)
        lb = [lb 1 1]; % delta zeta
        ub = [ub inf inf];
    case 12 %'BB7' % (Joe and Hu, 1996) (Joe book page 202)
        lb = [lb 0 1]; % delta zeta
        ub = [ub inf inf];
```

```
    case 13 %'BB8' % (Joe 1993) (Joe book page 204)
        lb = [lb 0 1]; % delta zeta
        ub = [ub 1 inf];
    case 14 %{'BB9'} % or 'Crowder' (Joe book page 205)
        lb = [lb 0 1]; % delta zeta
        ub = [ub inf inf];
    case 15 %{'BB10'} % (Joe book page 206)
        lb = [lb 0 0]; % delta zeta
        ub = [ub 1 inf];
    otherwise
     fprintf(' Abending: fncopularnd_h_fun, Input family is not in \n
        ');
end
```

```
function f=fn_RVine_PCC_Loglik(x1,ua,i,j,R_vine_family)
delta=x1(1);
if size(x1,2) > 1
    zeta=x1(2);
end
fam=R_vine_family{i,j};
ua(1,:)=U_transf(ua(1,:));
ua(2,:)=U_transf(ua(2,:));
m=ua(1,:);
n=ua(2,:);
u=ua(1,:);
v=ua(2,:);   % v|u
    switch fam % pcc log likelihood (uses pdf function)
        case 'Gaussian'
            ua=ua';
            f = -sum(log(copulapdf('Gaussian',ua,delta)));
        case 't'
            ua=ua';
            f = -sum(log(copulapdf('t',ua,delta,zeta)));
        case 'Clayton'
             f = -sum(log(copulapdf('Clayton',ua',delta)));
        case 'Frank'
            ua=ua';
            f = -sum(log(copulapdf('Frank',ua,delta)));
        case 'Gumbel'
            ua=ua';
            f = -sum(log(copulapdf('Gumbel',ua,delta)));
        case {'Joe','B5'}
            m=1-m;
            n=1-n;
            z=m.^delta + n.^delta - (m.^delta).*(n.^delta);
            a=(-2+(1/delta))*sum(log(z));
            b=(delta-1)*sum(log(m));
            c=(delta-1)*sum(log(n));
            d=sum(log(delta - 1 + z));
            f = -(a+b+c+d);
        case {'Galambos'}
            x= -log(u); y= -log(v);
            c1=x.^(-delta) + y.^(-delta);
            c3=x.^(-delta-1) + y.^(-delta-1);
            cdf1= sum(c1.^(-1/delta));
```

```
            pdf = sum(log(1 - ((c1).^(-1-(1/delta))).*(c3) + ((c1).^(-2-(1/delta))).* ...
                      ((x.*y).^(-delta-1)).*(1 + delta + (c1.^(-1/delta))) ));
            f=-(cdf1+pdf);
        case {'JM'} % Joe and Ma,2000
            T= size(v,1);
            uinv=gaminv(1-u,delta,1);
            vinv=gaminv(1-v,delta,1);
            s= (uinv.^delta) + (vinv.^delta);
            a=T*log(delta^(-1));
            b=log(s.^((1/delta)-1));
            c=-s.^(1/delta);
            d=-T*log(gamma(1+delta));
            phi2diff=a+sum(b)+sum(c)+d;
            gam=2*T*log((gamma(1+delta)));
            denor=-sum(uinv+vinv);
            f= -(gam + phi2diff - denor);
        case {'BB1'} % (Joe and Hu, 1996)
            x= ((u.^(-zeta))-1 ).^delta;
            y= ((v.^(-zeta))-1 ).^delta;
            a=sum(((-1/zeta)-2).*log(1+(x+y).^(1/delta)));
            b=sum(((1/delta)-2).*log(x+y));
            c=sum(log(zeta*(delta-1) + (zeta*delta +1).*(x+y).^(1/delta) ));
            d=sum((1-(1/delta)).*log(x.*y));
            e=sum((-zeta-1).*log(u.*v));
            f= -(a+b+c+d+e);
        case {'BB2'} % (Joe and Hu, 1996)
            x= exp(delta.*((u.^(-zeta))-1))-1;
            y= exp(delta.*((v.^(-zeta))-1))-1;
            a= x+y+1;
            h= sum( ((-1/zeta)-2).*log( 1 + (delta.^(-1)).*log(a)) );
            i= sum( -2.*log(a) );
            j= sum( log(1+zeta+ (zeta*delta) + zeta.*log(a) ) );
            k= sum( log(x+1) );
            l= sum( log(y+1) );
            m= sum( (-zeta-1).*log(u.*v) );
            f= -(h+i+j+k+l+m);

        case {'BB3'} % (Joe and Hu, 1996)
             T= size(v,2);
             x= exp(delta.*(-log(u)).^zeta)-1;
             y= exp(delta.*(-log(v)).^zeta)-1;
             a=log(x+y+1);
             lgG_bar= -((1/delta).*a).^(1/zeta);

             h= sum(lgG_bar);
             i= sum( ((1/zeta)-2).*log(a) );
             j= sum( -2.*a );
             k1=(delta^(1/zeta))*(zeta-1);
             k2=zeta.*(delta.^(1/zeta)).*a;
             k3= a.^(1/zeta);
             k= sum( log( k1 + k2 + k3 ));
             l= T*sum(log( delta^(2-(2/zeta)) ));
             m= sum( log(x+1) );
```

```
            n= sum( log(y+1) );
            o= sum( (zeta-1).*log(-log(u)) );
            p= sum( (zeta-1).*log(-log(v)) );
            q= sum( log(u.*v));
            f= -(h+i+j+k+l+m+n+o+p-q);
        case {'BB4'} % (Joe and Hu, 1996)
            x=((u.^(-zeta))-1).^(-delta);
            y=((v.^(-zeta))-1).^(-delta);
            a= 1+ (x.^(-1/delta)) + (y.^(-1/delta)) - ((x+y).^(-1/delta));

            h= sum( ((-1/zeta)-2).*log( a ) );
            i= sum( (1+(1/delta)).*log(x.*y) );
            j= sum( (-zeta-1).*log( u.*v ) );
            k1= zeta+1;
            k2= (x.^((-1/delta)-1)) - ((x+y).^((-1/delta)-1) );
            k3= (y.^((-1/delta)-1)) - ((x+y).^((-1/delta)-1) );
            k4= zeta.*(1+delta);
            k5= a.*((x+y).^((-1/delta)-2));
            k= sum( log( (k1.*k2.*k3) + (k4.*k5)) );

            f= -(h+i+j+k);
        case {'BB5'} % (Joe and Hu, 1996)
            x= -log(u);
            y= -log(v);
            a1= (x.^(-zeta*delta)) + (y.^(-zeta*delta));
            t= (x.^zeta)+ (y.^zeta) - (a1.^(-1/delta));
            lgG_bar= -t.^(1./zeta);
            a=x; b=y;
            z1 = (a.^(zeta-1)) - ((a.^(-zeta*delta)) + b.^(-zeta*delta)).^((-1/delta)-1).*...
              (a.^(-(zeta*delta)-1));

            h= sum( -log( u.*v ) );
            i= sum( lgG_bar );
            j= sum( ((1/zeta)-2).*log(t) );

            k1= (t.^(1/zeta)) + zeta-1;
            a=y; b=x;
            z2 = (a.^(zeta-1)) - ((a.^(-zeta*delta)) + b.^(-zeta*delta)).^((-1/delta)-1).*...
              (a.^(-(zeta*delta)-1));
            k2= zeta.*(1+delta).*t;
            k3= (a1.^((-1/delta)-2)).*(x.^(-(zeta*delta)-1)).*(y.^(-(zeta*delta)-1));
            k= sum( log( (k1.*z1.*z2) + (k2.*k3) ) );

            f= -(h+i+j+k);
        case {'BB6'} % (Joe and Hu, 1996)
            x= -log(1-(1-u).^zeta);
            y= -log(1-(1-v).^zeta);
            w= exp(-(x.^delta + y.^delta).^(1/delta));
            lgw= -(x.^delta + y.^delta).^(1/delta);
            a=(x.^delta + y.^delta);
            u_bar= 1-u; v_bar=1-v;

```

```
            h= sum( ((1/zeta)-2).*log( (1-w) ) );
            i= sum( lgw );
            j= sum( ((1/delta)-2).*log(a) );
            k1= (zeta-w).*(a.^(1/delta));
            k2= (zeta.*(delta-1).*(1-w));
            k= sum( log( k1+k2 ) );
            l= sum( (delta-1).*log( x.*y ) );
            m= sum( -log( 1-u_bar.^zeta ) );
            n= sum( -log( 1-v_bar.^zeta ) );
            p= sum( (zeta-1).*log( u_bar.*v_bar ) );
            f= -(h+i+j+k+l+m+n+p);
        case {'BB7'} % (Joe and Hu, 1996)
            x= ((1-(1-u).^zeta).^(-delta))-1;
            y= ((1-(1-v).^zeta).^(-delta))-1;

            h= sum( ((1/zeta)-2).*log( (1-(x+y+1).^(-1/delta) ) ));
            i= sum( ((-1/delta)-2).*log(x+y+1) );
            j= sum( (1+(1/delta)).*log((x+1).*(y+1)) );
            k1= zeta.*(delta+1);
            k2= ((zeta*delta)+1).*(x+y+1).^(-1/delta);
            k= sum( log(k1-k2) );
            l= sum( (zeta-1).*log((1-u).*(1-v)) );
            f= -(h+i+j+k+l);
        case {'BB8'} % (Joe 1996)
             T=size(v,2);
             eta=1-(1-delta).^zeta;
             x=1-(1-(delta.*u)).^zeta;
             y=1-(1-(delta.*v)).^zeta;
             a1=1-((1./eta).*x.*y);

             h= -T.*log(eta);
             i= T.*log(delta);
             j= sum( ((1/zeta)-2).*log(a1) );
             k= sum( log(zeta-(1/eta).*x.*y ) );
             l= sum( (zeta-1).*log(1-(delta.*u)) );
             m= sum( (zeta-1).*log(1-(delta.*v)) );

             f= -(h+i+j+k+l+m);
        case {'BB9','Crowder'}
             x=(delta.^(-1))-log(u);
             y=(delta.^(-1))-log(v);
             a1= x.^zeta + y.^zeta - delta.^(-zeta);
             lgG_bar= -a1.^(1/zeta) + (delta.^(-1)) ;

             h= sum(lgG_bar) + sum( ((1/zeta)-2).*log(a1) );
             i= sum( log( (a1.^(1/zeta))+zeta-1 ) );
             j= sum( (zeta-1).*log(x.*y) );
             k= sum(-log(u.*v));
             f=-(h+i+j+k);

        case {'BB10'}

             h=sum( ((-1/zeta)-2).*log( 1-(delta.*(1-u.^zeta).*(1-v
                 .^zeta)) ) );
             i=sum( log( 1-delta+...
                                (delta.*(1+zeta).*((u.*v).^(zeta)))-...
```

```
                                   (delta.*(1-delta).*(1-u.^zeta).*(1-v.^
                                       zeta)) ));
                  f=-(h+i);
          otherwise
                fprintf(' Abending: Input family is not in \n');

     end
end
```

```
function p=fncopularnd_h_fun(family,dd,u,v)
% this is conditional cdf copula function
% it is also called h function
% it comes from partial diff of cdf fun by u ; given cdf is C(u,v;
    delta)
delta=dd(1);
if size(dd,2) > 1
     zeta=dd(2);
end

u=U_transf(u);
v=U_transf(v);

switch family

     % conditional cdf v|u function = h fun
      case 'Gaussian' % joe book page 163
          Rho=dd(1);
          if dd(1) < -1 || dd(1) > 1
                dd(1);
          end
          p=normcdf((norminv(v) - Rho.*norminv(u))./(sqrt(1-Rho^2)));
     case 't' % joe book page 163 equa 4.33
          Rho=dd(1); nu=dd(2);
          p=tcdf((tinv(v,nu) - Rho.*tinv(u,nu))./...
               (sqrt( (1-Rho^2).*...
                       (nu+(tinv(u,nu)).^2) ./ ...
                       (nu+1) ...
                     )...
                ), nu+1);
     case 'Clayton' % a.k.a. Cook-Johnson
            p = (1+ (u.^delta.*((v.^(-delta))-1) )).^(-1-(1/delta));
     case 'Frank'
            p = exp(-delta.*u).*( (1-exp(-delta)).*(1-exp(-delta.*v))
                .^(-1) -...
                                  (1-exp(-delta.*u))).^(-1);
     case 'Gumbel' % a.k.a. Gumbel-Hougaard
          x=-log(u); y=-log(v);
          p = (u.^(-1)).*exp( -(x.^(delta) + y.^(delta)).^(1/delta) ).*
               ...
                (1 + (y./x).^delta).^((1/delta)-1);
     case {'Joe','B5'}
          p = ((1 + ((1-v).^delta).*((1-u).^(-delta)) - ...
                ((1-v).^delta)).^(-1+(1./delta))).*(1-(1-v).^delta);
     case 'Galambos'
          x=-log(u); y=-log(v);
          p = v.*exp((x.^(-delta) + y.^(-delta)).^(-1/delta)).*...
```

```
                (1 - ( 1 + (x./y).^delta).^(-1-(1/delta))) ;
    case 'BB1' % (Joe and Hu, 1996)
        x = ((u.^(-zeta))-1 ).^delta;
        y = ((v.^(-zeta))-1 ).^delta;
        p = ((1 + (x+y).^(1/delta)).^((-1/zeta)-1)).*  ...
                    (( x+y).^((1/delta)-1)).* ...
                    ( x.^(1-(1/delta))).* ...
                    ( u.^(-zeta-1));
     case 'BB2' % (Joe and Hu, 1996) (Joe book page 194)
        x= exp(delta.*((u.^(-zeta))-1))-1;
        y= exp(delta.*((v.^(-zeta))-1))-1;
        a=x+y+1;
        p = ((1 + (delta.^(-1)).*log(a)).^((-1/zeta)-1)).* ...
                    ( a.^(-1) ).* ...
                    ( (x+1).*(u.^(-zeta-1)) );
      case 'BB3' % (Joe and Hu, 1996) (Joe book page 195)
        x= exp(delta.*(-log(u)).^zeta)-1;
        y= exp(delta.*(-log(v)).^zeta)-1;
        G_bar= exp( -((1/delta).*log(x+y+1)).^(1/zeta) );
        p = G_bar.*(( log(x+y+1) ).^( (1/zeta) -1)).*...
                    (1./(x+y+1)).* ...
                    delta.^(1-(1/zeta)).* ...
                    (x+1).*( (-log(u)).^(zeta-1))./u;
      case 'BB4' % (Joe and Hu, 1996) (Joe book page 197)
        x=((u.^(-zeta))-1).^(-delta);
        y=((v.^(-zeta))-1).^(-delta);
        a= 1+ (x.^(-1/delta)) + (y.^(-1/delta)) - ((x+y).^(-1/delta))
            ;
        p = (a.^((-1/zeta)-1)).* ...
                    ( (x.^((-1/delta)-1)) - (x+y).^((-1/delta)-1) )
                        .*...
                    (x.^(1+(1/delta)))./(u.^(1+zeta));
      case 'BB5' % (Joe and Hu, 1996) (Joe book page 199)
          x= -log(u);
          y= -log(v);
          a= (x.^(-zeta*delta)) + (y.^(-zeta*delta));
          t= (x.^zeta)+ (y.^zeta) - (a.^(-1/delta));
          G_bar= exp(-t.^(1./zeta) );

          b1= x.^(zeta-1);
          b2= (x.^(-zeta*delta) + y.^(-zeta*delta)).^((-1/delta)-1) ;
          b3= x.^((-zeta*delta)-1);
          p= (1./u).*G_bar.*(t.^((1/zeta)-1)).* ...
                      ( b1 - (b2.*b3));
    case 'BB6' % (Joe and Hu, 1996) (Joe book page 200)
          x= -log(1-(1-u).^zeta);
          y= -log(1-(1-v).^zeta);
          w= exp(-(x.^delta + y.^delta).^(1/delta));
          p= ((1-w).^((1/zeta)-1)).*w.* ...
                      ( (x.^delta+y.^delta).^((1/delta)-1) ).*...
                      (x.^(delta-1)).*exp(x).* ...
                      ( (1-exp(-x)).^(1-(1/zeta)) );
   case 'BB7' % (Joe and Hu, 1996) (Joe book page 202)
          x= ((1-(1-u).^zeta).^(-delta))-1;
          y= ((1-(1-v).^zeta).^(-delta))-1;
          p= ((1-(x+y+1).^(-1/delta)).^((1/zeta)-1)).* ...
```

```
                       ((x+y+1).^((-1/delta)-1)).* ...
                       ( (x+1).^(1+(1/delta)) ).* ...
                       ( (1-u).^(zeta-1));
   case 'BB8' % (Joe 1993) (Joe book page 204)
          eta=1-(1-delta)^zeta;
          x=1-(1-(delta.*u)).^zeta;
          y=1-(1-(delta.*v)).^zeta;
          a1=1-((1./eta).*x.*y);
          p= ((1./eta).*y.*(a1.^((1/zeta)-1)))./ ...
                       ((1-x).^((1/zeta)-1));
   case {'BB9'} % or 'Crowder' (Joe book page 205)
          x=(delta.^(-1))-log(u);
          y=(delta.^(-1))-log(v);
          a1= x.^zeta + y.^zeta - (delta.^(-zeta));
          G_bar= exp( -(a1.^(1/zeta)) + (delta.^(-1)) );
          p = G_bar.*(a1.^((1/zeta)-1)).*(x.^(zeta-1)).*(1./u);
    case {'BB10'} % (Joe book page 206)
        p= ((1-(delta.*(1-u.^zeta).*(1-v.^zeta))).^((-1/zeta)-1)).* ...
                     (v.*(1-(delta.*(1-(v.^zeta)))) );
    otherwise
        fprintf(' Abending: fncopularnd_h_fun, Input family is not in \n');
end

end
```

```
function U = U_transf(U)
    llim=1.d-4;ulim=1.d0-llim;
    U(U<llim) =  llim;
    U(U>ulim) =  ulim;
```

The following miscellaneous functions are used in the variational Bayes algorithm.

```
function b = B2vechB(B,K)
ind = zeros(K,K)+tril(ones(K,K));
ind = ind(:);
nozero = B(1:K,1:K);
nozero = nozero(:);
b1 = nozero(ind~=0);
b2 = B(K+1:end,1:K);
b2 = b2(:);
b = [b1;b2];
end
```

```
function [g_lambda_L,loghhat,log_theta,log_q_lambda,g_lambda_logq,er_z,...
          er_llh,abnor_endjob,zt_mean,ht_mean] = ...
                Empirical_FFVB_Est_all_EGARCH_gradient_computeM...
                (theta,gamma,u,z,ai,bi,lambda,k,p,VA,ct,x_pre,va_0,Ub,Lb)
n = size(theta,2);     % fix 25                                  %Number of copula parameters
[~,r] = size(ai);
S = size(theta,1);
```

```
numelait = size(u,1); % lenght of time series
if lambda(3) < -1e-8
    lambda;
end
[mu,Sigma,B,D,muz,logsigmaz,C,er_dm2] = FFVB_EGARCH_lambda2musigma...
                                  (lambda,n,numelait,k,VA);
er_llh=0;
abnor_endjob=0;
for i_lgh=1:S
    x_pre=gamma(i_lgh,:);
    [loghhat(i_lgh),zts(:,i_lgh),hts(:,i_lgh)]= FFVB_EGARCH_loghhat(u
        ,x_pre);
end
zt_mean=mean(zts,2);
ht_mean=mean(hts,2);
log_theta = sum(log(0.995)-theta-2*log(exp(-theta)+1),2)';
                        %Implied by uniform on gamma
log_q_lambda =  log_mvnpdf(theta',repmat(mu,1,S),Sigma)';
er_z=0;
if VA == 1
    if va_0==1
        log_q_u=0;
    else
        u1 = unifrnd(0,1,S,numelait);
        log_q_u=sum(log(u1'));
    end
elseif VA == 2
    z1pdf = normpdf(z,muz,exp(2.*logsigmaz)); % exp(2*logsigmaz)
    z2pdf = normpdf(z,0,1);
    u1=log(z1pdf./z2pdf);
    log_q_u=sum(u1);
    er_z=0;
end
log_q_lambda = log_q_lambda + log_q_u;
g_lambda_logq = FFVB_EGARCH_grad_lambda_logq(theta,z,lambda,k,
   numelait,VA);
nLambda = size(g_lambda_logq,1);
g_lambda_L = [];
if ~isempty(ct) && abnor_endjob==0
    g_lambda_L = zeros(nLambda,1);
    parfor i = 1:nLambda % cal 2.b
        g_lambda_L(i) = mean((loghhat+log_theta-log_q_lambda-ct(i)).*
            g_lambda_logq(i,:));
    end
end
```

```
function VBDAobj = Empirical_FFVB_Est_all_EGARCH_VBDAfit_multi...
                   (y,p,k,S,nVB,VA,x_pre,va_0,i)
disp('----> VB estimation underway')
% x_pre
Lb = [ 0.00,0.00, x_pre(3)-0.1, x_pre(4)-0.1,x_pre(5)-0.3,x_pre(6)
   -0.3,   0.00,0.4];
Ub = [ 0.50,0.50,x_pre(3)+0.1,x_pre(4)+0.1,x_pre(5)+0.3,x_pre(6)+0.3,
    0.50,   1];

[ai,bi,n,LB,ct,lambda,t,mu,Sigma,r,muz,logsigmaz,C,T,B,D] =...
```

```
      Empirical_FFVB_Est_all_EGARCH_VBinivalM...
             (y,p,k,S,nVB,VA,x_pre,va_0,Ub,Lb);
ADA = [];
ck=100; tw=20; patience_max=20;
brkCondition=0;
loghhat=0;
while(t<nVB)
    t = t+1;
    % 3.2.2 and 3.2.1 respectively
    [theta,gamma] = FFVB_EGARCH_q_lambda(mu,Sigma,S,B,D,Ub,Lb);
    z = FFVB_EGARCH_q_u(muz,logsigmaz,C,S,VA);
    u = FFVB_EGARCH_uDaibi2u(bi,S);
    loghhat_pre=loghhat;
    [g_lambda_L,loghhat,log_theta,log_q_lambda,grad_lambda,er_z,
       er_llh,...
     abnor_endjob,zt_mean,ht_mean] = ...
         Empirical_FFVB_Est_all_EGARCH_gradient_computeM...
         (theta,gamma,u,z,ai,bi,lambda,k,p,VA,ct,x_pre,va_0,Ub,Lb);
     lambda_pre=lambda;
    if abnor_endjob==0 % zero = normal. one = abnormal.
    ct = FFVB_EGARCH_ct_constant(loghhat,log_theta,log_q_lambda,
       grad_lambda);
    [Change_delta,ADA] = FFVB_New_ADADELTA(ADA,g_lambda_L);
    lambda = lambda + Change_delta;
    [mu,Sigma,B,D,muz,logsigmaz,C,er_dm2] =
       FFVB_EGARCH_lambda2musigma(lambda,n,T,k,VA);
    LB(t) = mean(loghhat+log_theta-log_q_lambda);
    end
    if t > ck || abnor_endjob==1
        lb=0;
        parfor k=1:tw
            a=LB(t-k+1);
            lb=lb+a;
        end
        LB_bar(t-tw) = lb/tw;
        if LB_bar(t-tw) >= max(LB_bar(ck-tw+1:end))
            best_mu=mu;
            best_Sigma=Sigma;
            best_B=B;
            best_D=D;
            best_zt=zt_mean;
            best_ht=ht_mean;
            patience = 0;
            loglk=mean(loghhat);
            [~,gamma] = FFVB_EGARCH_q_lambda(best_mu,best_Sigma
                ,10000,best_B,best_D,Ub,Lb);
        else
            if mod(patience,20)==0
                patience;
            end
            patience = patience + 1;
        end
        if patience >= patience_max
            brkCondition=1;
            break;
        end
```

```
    end
end
[~,gamma] = FFVB_EGARCH_q_lambda(best_mu,best_Sigma,10000,best_B,
    best_D,Ub,Lb);
VBDAobj.gamma_mean = mean(gamma);
VBDAobj.gamma_sd = std(gamma);
VBDAobj.mu = best_mu;
VBDAobj.B = best_B;
VBDAobj.D = best_D;
VBDAobj.muz = muz;
VBDAobj.logsigmaz = logsigmaz;
VBDAobj.C = C;
VBDAobj.lambda = lambda;
VBDAobj.loglik = loglk;
VBDAobj.LB = LB;
VBDAobj.y = y;
VBDAobj.p = p;
VBDAobj.k = k;
VBDAobj.S = S;
VBDAobj.nVB = nVB;
VBDAobj.ck = ck;
VBDAobj.t = t;
VBDAobj.time = time;
VBDAobj.tw = tw;
VBDAobj.brkCondition=brkCondition;
VBDAobj.patience_max = patience_max;
VBDAobj.patience = patience;
VBDAobj.VA = VA;
VBDAobj.va_0=va_0;
VBDAobj.AIC=0;
VBDAobj.BIC=0;
VBDAobj.zt_mean=best_zt;
VBDAobj.ht_mean=best_ht;
```

```
function [ai,bi,n,LB,ct,lambda,t,mu,Sigma,r,muz,...
          logsigmaz,C,T,B,D,Ub,Lb] =
              Empirical_FFVB_Est_all_EGARCH_VBinivalM(y,p,k,S,nVB,VA,
              x_pre,va_0,Ub,Lb)

[T,r] = size(y);
ai = zeros(size(y));
bi=y;
numelait = length(ai);
if VA == 1
    muz = [];
    logsigmaz = [];
    C = [];
elseif VA == 2
    muz = zeros(numelait,1);
    logsigmaz = zeros(numelait,1);
    C = [];
end
Inipref =x_pre; %x_pre,x_covar
n_par=size(Inipref,2);
n = (r*(r-1)*0.5+r*r*p)*n_par;                % Number of copula
    parameters
```

```
% Initial value of mean variational parameters
mu = repmat(Inipref(:),(r*(r-1)*0.5+r*r*p),1);
Sigma = eye(n)*0.1;
LB = zeros(nVB,1); % lower bound
[lambda,B,D]= FFVB_EGARCH_musigma2lambda(mu,Sigma,muz,logsigmaz,C,k);
% 3.2.2 and 3.2.1 respectively
[theta,gamma] = FFVB_EGARCH_q_lambda(mu,Sigma,S,B,D,Ub,Lb);
z = FFVB_EGARCH_q_u(muz,logsigmaz,C,S,VA);
u = FFVB_EGARCH_uDaibi2u(bi,S);

[~,loghhat,log_theta,log_q_lambda,grad_lambda,er_z,er_llh,
    abnor_endjob] = ...
         Empirical_FFVB_Est_all_EGARCH_gradient_computeM...
         (theta,gamma,u,z,ai,bi,lambda,k,p,VA,[],x_pre,va_0,Ub,Lb);
ct = FFVB_EGARCH_ct_constant(loghhat,log_theta,log_q_lambda,
    grad_lambda);
t = 0;
```

```
function [UB,LB]=Empirical_FFVB_New_PCC_cop_boundary(family,S,npar)

UB = ones(S,npar);
LB = ones(S,npar);

switch family
    case 'Gaussian'
        UB =  0.99.*UB;
        LB = -0.99.*LB;
    case 't'
        UB = [0.99 30].*UB;
        LB = [-0.99 3].*LB;
    case  'Gumbel'
        UB = 5.*UB;
        LB = 1.*LB;
    case  'Clayton' %
        UB = 2.*UB;
        LB = 0.*LB;
    case  'Frank'
        UB =  5.*UB;
        LB = -2.*LB;
    case  {'Joe','B5' }
        UB = 5.*UB;
        LB = 1.*LB;
    case  'Galambos'
        UB = 2.*UB;
        LB = 0.*LB;
    case 'BB1'
        UB = [5 3].*UB;
        LB = [1 0].*LB;
    case 'BB6'
        UB = [1000 1000].*UB;
        LB = [1 1].*LB;
    case 'BB3'
        UB = [2.5 2.5].*UB;
        LB = [0 1].*LB;
    case {'BB2','BB4'} % (Joe and Hu, 1996) (Joe book page 197)
        UB = [5 5].*UB;
```

```
        LB = [0 0].*LB;
    case {'BB5','BB7','BB9'}  % or 'Crowder' (Joe book page 205)
        UB = [3 4].*UB;
        LB = [0 1].*LB;
    case 'BB8'
        UB = [1 1000].*UB;
        LB = [0 1].*LB;
    case 'BB10'
        UB = [1 60].*UB;
        LB = [0 0].*LB;
    otherwise
     fprintf(' Abending: FFVB_New_PCC_cop_boundary, No family
        boundary identification in \n');
end
```

```
function [theta,T_theta] = Empirical_FFVB_New_q_lambda(mu,Sigma,S,
    family,B,D)
npar = length(mu);
% para of copula boundary
[UB,LB]=Empirical_FFVB_New_PCC_cop_boundary(family,S,npar);
if nargin == 5
    target.mu = mu';
    target.chol = chol(Sigma);
    target.M = S;
    target.Dims   = length(mu);
    theta = q_mvnrnd(target);
elseif nargin == 6
    k = size(B,2);
    z = randn(k,S);
    eps = randn(npar,S);
    theta = repmat(mu,1,S)+B*z+D*eps;
    theta = theta';
end
T_theta = (UB-LB)./(exp(-theta)+1)+LB;
```

```
function VBDAobj = Empirical_FFVB_New_VBDAfit_multi(y,family,k,S,nVB,
    VA,va_0,x_pre,sigma)

disp('----> VB estimation underway')
[n,LB,ct,lambda,t,mu,Sigma,r,muz,logsigmaz,C,B,D] =...
      Empirical_FFVB_New_VBinivalM_fix_low_lglik(y,family,k,S,nVB,VA,
         va_0,x_pre,sigma);
ADA = [];
T=size(y,1);
ck=250; tw=20; patience_max=30;
%
brkCondition=0;
u=y';
while(t<nVB)
    t = t+1;
    % 3.2.2 and 3.2.1 respectively
    [theta,gamma] = Empirical_FFVB_New_q_lambda(mu,Sigma,S,family,B,D
       );
    z = FFVB_New_q_u(muz,logsigmaz,C,S,VA);
    [g_lambda_L,loghhat,log_theta,log_q_lambda,grad_lambda,er_z] =
       ...
```

```
        FFVB_New_gradient_computeM_fix_low_lglik...
          (theta,gamma,u,z,family,lambda,k,VA,va_0,ct);
  if er_z < 1000
    ct = FFVB_New_ct_constant(loghhat,log_theta,log_q_lambda,
       grad_lambda);
    [Change_delta,ADA] = FFVB_New_ADADELTA(ADA,g_lambda_L);
    lambda = lambda + Change_delta;
    [mu,Sigma,B,D,muz,logsigmaz,C,er_dm2] = FFVB_New_lambda2musigma
       ...
                                  (lambda,n,T,k,VA);
    LB(t) = mean(loghhat+log_theta-log_q_lambda);

    if t > ck
        lb=0;
        parfor k=1:tw
            a=LB(t-k+1);
            lb=lb+a;
        end
        LB_bar(t-tw) = lb/tw;
        if LB_bar(t-tw) >= max(LB_bar(ck-tw+1:end))
            best_mu=mu;
            best_Sigma=Sigma;
            best_B=B;
            best_D=D;
            patience = 0;
            [~,gamma] = Empirical_FFVB_New_q_lambda(mu,Sigma,10000,
               family,B,D);
            patience = patience + 1;
        end
        if patience >= patience_max
            brkCondition=1;
            break;
        end
    end
  else
      break;
  end
end
[~,gamma] = Empirical_FFVB_New_q_lambda(best_mu,best_Sigma,10000,
   family,best_B,best_D);
VBDAobj.gamma_mean = mean(gamma);
VBDAobj.gamma_sd = std(gamma);
VBDAobj.family = family;
VBDAobj.mu = best_mu;
VBDAobj.B = best_B;
VBDAobj.D = best_D;
VBDAobj.muz = muz;
VBDAobj.logsigmaz = logsigmaz;
VBDAobj.C = C;
VBDAobj.lambda = lambda;
VBDAobj.LB = LB;
VBDAobj.loghhat = mean(loghhat);
VBDAobj.y = y;
VBDAobj.p = p;
VBDAobj.k = k;
VBDAobj.S = S;
```

```
VBDAobj.nVB = nVB;
VBDAobj.ck = ck;
VBDAobj.t = t;
VBDAobj.time = time;
VBDAobj.tw = tw;
VBDAobj.brkCondition=brkCondition;
VBDAobj.patience_max = patience_max;
VBDAobj.patience = patience;
VBDAobj.VA = VA;
VBDAobj.va0 = va_0;
```

```
function [n,LB,ct,lambda,t,mu,Sigma,r,muz,logsigmaz,C,B,D] =...
                Empirical_FFVB_New_VBinivalM_fix_low_lglik(y,family,k
                    ,S,nVB,VA,va_0,x_pre,sigma)
[numelait,r] = size(y);

if VA == 1
    muz = [];
    logsigmaz = [];
    C = [];
elseif VA == 2
    muz = zeros(numelait,1);
    logsigmaz = zeros(numelait,1);
    C = [];
elseif VA == 3
    muz = zeros(numelait,1);
    logsigmaz = [];
    C = sparse(1:numelait,1:numelait,1,numelait,numelait);
end

Inipref = x_pre;                                                    %
    Initial value of mean variational parameters
n_par=size(Inipref,2);
n = n_par;

mu = repmat(Inipref(:),1);
Sigma = eye(n)*0.1;
LB = zeros(nVB,1); % lower bound
[lambda,B,D]= FFVB_New_musigma2lambda(mu,Sigma,muz,logsigmaz,C,k);
[theta,gamma] = Empirical_FFVB_New_q_lambda(mu,Sigma,S,family,B,D);
z = FFVB_New_q_u(muz,logsigmaz,C,S,VA);
u=y';
[~,loghhat,log_theta,log_q_lambda,grad_lambda,~] =
    FFVB_New_gradient_computeM_fix_low_lglik...
            (theta,gamma,u,z,family,lambda,k,VA,va_0,[]);
ct = FFVB_New_ct_constant(loghhat,log_theta,log_q_lambda,grad_lambda)
    ;
t = 0;
```

```
function ct = FFVB_EGARCH_ct_constant(LogLhat,log_theta,log_q_lambda,
    grad_lambda)
nLambda = size(grad_lambda,1);
ct = zeros(nLambda,1);
parfor i = 1:nLambda
    v = cov( (LogLhat+log_theta-(log_q_lambda)) .*grad_lambda(i,:),
        grad_lambda(i,:));
```

```
    ct(i) = v(1,2)/v(2,2);
end
```

```
function grad_lambda = FFVB_EGARCH_grad_lambda_logq...
                                (theta,z,lambda,k,T,VA)
S = size(theta,1);
d = size(theta,2);
[mu,Sigma,B,D,muz,logsigmaz,C] = FFVB_EGARCH_lambda2musigma...
                                        (lambda,d,T,k,VA);
Dm2 = sparse(1:d,1:d,1./(diag(D).^2),d,d);
Sigmainv = Dm2-Dm2*B/(sparse(1:k,1:k,1,k,k)+B'*Dm2*B)*B'*Dm2;
SigmainvB = Sigmainv*B;
SigmainvD = Sigmainv*D;
Term1 = (theta-repmat(mu',S,1))';
dmu = Sigmainv*Term1;
dB = zeros(size(B2vechB(B,k),1),S);
dD = zeros(size(Sigma,1),S);
parfor s = 1:S
    % Appendix C: cal gradient b
    dBtemp = -SigmainvB+Sigmainv*(theta(s,:)'-mu)*(theta(s,:)'-mu)'*
       SigmainvB;
    dB(:,s) = B2vechB(dBtemp,k);
    % Appendix C: cal gradient d
    dD(:,s) = diag(-SigmainvD+Sigmainv*(theta(s,:)'-mu)*(theta(s,:)'-
       mu)'*SigmainvD);
end
if VA == 1
    dmuz = [];
    dlogsigmaz = [];
    dC = [];
elseif VA == 2
    sigma2 = exp(2*logsigmaz);
    sigma2rep = repmat(sigma2,1,S);
    muZ = repmat(muz,1,S);
    dlogsigmaz = -1+(1./sigma2rep).*(z-muZ).^2;
    dmuz = (1./sigma2rep).*(z-muZ);
    dC = [];
end
grad_lambda = [dmu; dB; dD; dmuz; dlogsigmaz;dC];
```

```
function [mu,Sigma,B,D,muz,logsigmaz,C,er_dm2] =
   FFVB_EGARCH_lambda2musigma(lambda,d,T,k,VA)
er_dm2=0;
if VA == 1
    mu = lambda(1:d);
    B =  vechB2B(lambda((d+1):(end-d)),d,k);
    D =  sparse(1:d,1:d,lambda((end-d+1):end),d,d);
    Dm2 = sparse(1:d,1:d,1./(diag(D).^2),d,d);
    Sigma = B*B'+D.^2;
    muz = [];
    logsigmaz = [];
    C = [];
elseif VA == 2
    mu = lambda(1:d);
    B =  vechB2B(lambda((d+1):(end-d-2*T)),d,k);
```

```
    D =  sparse(1:d,1:d,lambda((end-d+1-2*T):(end-2*T)),d,d);
    Sigma = B*B'+D.^2;
    Dm2 = sparse(1:d,1:d,1./(diag(D).^2),d,d);
    muz = lambda((end-2*T+1):(end-T));
    logsigmaz = lambda((end-T+1):(end));
    C = [];
end
```

```
function [loghhat,z11,hts1]=FFVB_EGARCH_loghhat(yt,para_temp)

xil=para_temp(1,1);
xir=para_temp(1,2);
lambda1=para_temp(1,3);
lambda2=para_temp(1,4);
omega=para_temp(1,5);
gam=para_temp(1,6);
delta=para_temp(1,7);
nu=para_temp(1,8);
p_next=[0.8;0.1;0.1];
betar=(1-xir)*(1-(2*xir))^0.5; %  beta
betal=(1-xil).*(1-(2.*xil)).^0.5; %  beta
n=size(yt,1);
hts1= zeros(n,1);
z11 = zeros(n,1);
hts1(1,1)=0.01;
z11(1,1)=(yt(1,1) - lambda1 - (lambda2.*exp(hts1(1,1)))))./sqrt(exp(
    hts1(1,1)));
for t=2:n
    hts1(t,1)=omega + gam.*z11(t-1,1) + delta.*abs(z11(t-1,1)) + nu
        .*(hts1(t-1,1));
    z11(t,1)=(yt(t,1) - lambda1 - (lambda2.*exp(hts1(t,1))))./...
                                    sqrt(exp(hts1(t,1)));
end

et=z11.*sqrt(exp(hts1));

lik(:,1)= pdf('normal',et,0,sqrt(exp(hts1)));
lik(:,2)=pdf('Generalized Pareto',et,xir,betar.*sqrt(exp(hts1)),0);
lik(:,3)=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts1)),0);
lik_w=lik.*p_next';
f=sum(lik_w,2);
loghhat= sum(log(f));
```

```
function [lambda,B,D] = FFVB_EGARCH_musigma2lambda(mu,Sigma,muz,
    logsigmaz,C,k)
d = size(Sigma,1);
B = zeros(d,k)+0.001;
D = diag(sqrt(diag(Sigma)));
b = B2vechB(B,k);
T = size(muz,1);
indPositiveC = [];
if ~isempty(C)
PositiveC = diag(ones(T,1))+diag(ones(T-1,1),-1);
indPositiveC = (PositiveC(:)==1);
end
```

```
lambda = [mu;b;sqrt(diag(Sigma));muz;logsigmaz;C(indPositiveC)];
```

```
function [theta,T_theta] = FFVB_EGARCH_q_lambda(mu,Sigma,S,B,D,Ub,Lb)
npar = length(mu);
    UB =Ub.*ones(S,npar);
    LB =Lb.*ones(S,npar);
if nargin == 5
    target.mu = mu';
    target.chol = chol(Sigma);
    target.M = S;
    target.Dims   = length(mu);
    theta = q_mvnrnd(target);
elseif nargin == 7
    k = size(B,2);
    z = randn(k,S);
    eps = randn(npar,S);
    theta = repmat(mu,1,S)+B*z+D*eps;
    theta = theta';
end
T_theta = (UB-LB)./(exp(-theta)+1)+LB;
```

```
function z = FFVB_EGARCH_q_u(muz,logsigmaz,C,S,VA)
T = size(muz,1);
if VA == 1
    z = [];
elseif VA == 2
    z = normrnd(0,1,T,S).*repmat(exp(logsigmaz),1,S)+repmat(muz,1,S);
end
```

```
function u = FFVB_EGARCH_uDaibi2u(bi,S)
bitc = bi';
bitc = bitc(:);
u= repmat(bitc,1,S);
```

```
function [Change_delta,ADA] = FFVB_New_ADADELTA(ADA,L)
if isempty(ADA)
    % initial value, follow appendix A
    Edelta2 = zeros(length(L),1);
    Eg2 = zeros(length(L),1);
    ADA.rho = 0.95;
    ADA.eps_step = 10^-6;
    ADA.Edelta2 = Edelta2;
    ADA.Eg2 = Eg2;
end
rho = ADA.rho;
eps_step = ADA.eps_step;
oldEdelta2 = ADA.Edelta2;
oldEg2 = ADA.Eg2;
%% mu update
ADA.Eg2 = rho*oldEg2 + (1-rho)*L.^2; % cal E_g^2 at k in appendix A;
    so, rho is actually zeta!!!
deltaK=sqrt(oldEdelta2 + eps_step)./sqrt(ADA.Eg2 + eps_step); % is
    rho in appendix A
lambdaK=L; % L is H and H is g in step 2.b
Change_delta = deltaK.*lambdaK;
```

```
ADA.Edelta2 = rho*oldEdelta2 + (1- rho)*Change_delta.^2;
```

```
function ct = FFVB_New_ct_constant(LogLhat,log_theta,log_q_lambda,
    grad_lambda)
% cal 2.c step and 1.b is a formula
nLambda = size(grad_lambda,1);
ct = zeros(nLambda,1);
parfor i = 1:nLambda
    v = cov( (LogLhat+log_theta-(log_q_lambda)) .*grad_lambda(i,:),
        grad_lambda(i,:));
    ct(i) = v(1,2)/v(2,2);
end
```

```
function grad_lambda = FFVB_New_grad_lambda_logq(theta,z,lambda,k,T,
    VA)

S = size(theta,1);
d = size(theta,2);
[mu,Sigma,B,D,muz,logsigmaz,C] = FFVB_New_lambda2musigma(lambda,d,T,k
    ,VA);

Dm2 = sparse(1:d,1:d,1./(diag(D).^2),d,d);
Sigmainv = Dm2-Dm2*B/(sparse(1:k,1:k,1,k,k)+B'*Dm2*B)*B'*Dm2;
SigmainvB = Sigmainv*B;
SigmainvD = Sigmainv*D;
Term1 = (theta-repmat(mu',S,1))';
dmu = Sigmainv*Term1;
dB = zeros(size(B2vechB(B,k),1),S);
dD = zeros(size(Sigma,1),S);
parfor s = 1:S
    % Appendix C: cal gradient b
    dBtemp = -SigmainvB+Sigmainv*(theta(s,:)'-mu)*(theta(s,:)'-mu)'*
        SigmainvB;
    dB(:,s) = B2vechB(dBtemp,k);
    % Appendix C: cal gradient d
    dD(:,s) = diag(-SigmainvD+Sigmainv*(theta(s,:)'-mu)*(theta(s,:)'-
        mu)'*SigmainvD);
end
if VA == 1
    dmuz = [];
    dlogsigmaz = [];
    dC = [];
elseif VA == 2
    sigma2 = exp(2*logsigmaz);
    sigma2rep = repmat(sigma2,1,S);
    muZ = repmat(muz,1,S);
    dlogsigmaz = -1+(1./sigma2rep).*(z-muZ).^2;
    dmuz = (1./sigma2rep).*(z-muZ);
    dC = [];
end
grad_lambda = [dmu; dB; dD; dmuz; dlogsigmaz;dC];
```

```
function [g_lambda_L,loghhat,log_theta,log_q_lambda,g_lambda_logq,
    er_z] = ...
    FFVB_New_gradient_computeM_fix_low_lglik(theta,gamma,u,z,family,
        lambda,k,VA,va_0,ct)
```

```
n = size(theta,2);
S = size(theta,1);
numelait = size(u,2); % leght of time series
[mu,Sigma,B,D,muz,logsigmaz,C] = FFVB_New_lambda2musigma(lambda,n,
    numelait,k,VA);
u(1,:) = U_transf(u(1,:));
u(2,:) = U_transf(u(2,:));
er_z=0;
parfor i=1:S
    pcc=fn_copula_pdf(gamma(i,:),u',1,1,family);
    loghhat(i) = sum(log(pcc));
end
er_z=sum(er_z);
log_theta = sum(log(0.995)-theta-2*log(exp(-theta)+1),2)';
                        %Implied by uniform on gamma
log_q_lambda =  log_mvnpdf(theta',repmat(mu,1,S),Sigma)';
if VA == 1
    if va_0==1
        log_q_u=0;
    else
        u1 = unifrnd(0,1,S,numelait);
        log_q_u=sum(log(u1'));
    end
elseif VA == 2
    z1pdf = normpdf(z,muz,exp(2.*logsigmaz));
    z2pdf = normpdf(z,0,1);
    u1=log(z1pdf./z2pdf);
    log_q_u=sum(u1);
    er_z=0;
    while( sum(isinf(log_q_u)) > 0 ...
          || sum(-isinf(log_q_u)) > 0 ...
          || sum(isnan(log_q_u)) > 0)
        zt=normrnd(0,1,numelait,S);
        z1pdf = normpdf(zt,muz,exp(2.*logsigmaz));
        z2pdf = normpdf(zt,0,1);
        u1=log(z1pdf./z2pdf);
        log_q_u=sum(u1);
    end
end
log_q_lambda = log_q_lambda + log_q_u;
g_lambda_logq = FFVB_New_grad_lambda_logq(theta,z,lambda,k,numelait,
    VA);
nLambda = size(g_lambda_logq,1);
g_lambda_L = [];
if ~isempty(ct)
    g_lambda_L = zeros(nLambda,1);
    parfor i = 1:nLambda % cal 2.b
        g_lambda_L(i) = mean((loghhat+log_theta-log_q_lambda-ct(i)).*
            g_lambda_logq(i,:));
    end
end
```

```
function [mu,Sigma,B,D,muz,logsigmaz,C,er_dm2] =
    FFVB_New_lambda2musigma...
                                (lambda,d,T,k,VA)
er_dm2=0;
```

```
if VA == 1
    mu = lambda(1:d);
    B =  vechB2B(lambda((d+1):(end-d)),d,k);
    D =  sparse(1:d,1:d,lambda((end-d+1):end),d,d);
    Dm2 = sparse(1:d,1:d,1./(diag(D).^2),d,d);
    Sigma = B*B'+D.^2;
    muz = [];
    logsigmaz = [];
    C = [];
elseif VA == 2
    mu = lambda(1:d);
    B =  vechB2B(lambda((d+1):(end-d-2*T)),d,k);
    D =  sparse(1:d,1:d,lambda((end-d+1-2*T):(end-2*T)),d,d);
    Sigma = B*B'+D.^2;
    Dm2 = sparse(1:d,1:d,1./(diag(D).^2),d,d);
    muz = lambda((end-2*T+1):(end-T));
    logsigmaz = lambda((end-T+1):(end));
    C = [];
end
```

```
function [lambda,B,D] = FFVB_New_musigma2lambda(mu,Sigma,muz,
    logsigmaz,C,k)
d = size(Sigma,1);
B = zeros(d,k)+0.001;
D = diag(sqrt(diag(Sigma)));
b = B2vechB(B,k);
T = size(muz,1);
indPositiveC = [];
if ~isempty(C)
PositiveC = diag(ones(T,1))+diag(ones(T-1,1),-1);
indPositiveC = (PositiveC(:)==1);
end
lambda = [mu;b;sqrt(diag(Sigma));muz;logsigmaz;C(indPositiveC)];
```

```
function z = FFVB_New_q_u(muz,logsigmaz,C,S,VA)
T = size(muz,1);
if VA == 1
    z = [];
elseif VA == 2
    z = normrnd(0,1,T,S).*repmat(exp(logsigmaz),1,S)+repmat(muz,1,S);
end
```

```
function f=fn_copula_pdf(x1,ua,i,j,R_vine_family)
delta=x1(1);
if size(x1,2) > 1
    zeta=x1(2);
end
fam=R_vine_family;
m=ua(:,1); n=ua(:,2);
u=ua(:,1); v=ua(:,2);    % v|u
v = U_transf(v); u = U_transf(u);
m = U_transf(m); n = U_transf(n);
ua=[u'; v'];
    switch fam % pcc likelihood (uses pdf function)
        case 'Gaussian'
```

```
            if delta >= 0.99 || delta <= -0.99
                delta
            end
            ua=ua';
            f = copulapdf('Gaussian',ua,delta);
        case 't'
            ua=ua';
            f = copulapdf('t',ua,delta,zeta);
        case 'Clayton'
            f = copulapdf('Clayton',ua',delta);
        case 'Frank'
            ua=ua';
            f = copulapdf('Frank',ua,delta);
        case 'Gumbel'
            ua=ua';
            f = copulapdf('Gumbel',ua,delta);
        case {'Joe','B5'}
            m=1-m;
            n=1-n;
            z=m.^delta + n.^delta - (m.^delta).*(n.^delta);
            a=z.^(-2+(1/delta));
            b=m.^(delta-1);
            c=n.^(delta-1);
            d=delta - 1 + z;
            f = a.*b.*c.*d;
        case {'Galambos'}
            x= -log(u); y= -log(v);
            c1=x.^(-delta) + y.^(-delta);
            c3=x.^(-delta-1) + y.^(-delta-1);
            cdf1= exp(c1.^(-1/delta));
            pdf = 1 - ((c1).^(-1-(1/delta))).*(c3) + ((c1).^(-2-(1/delta))).* ...
                       ((x.*y).^(-delta-1)).*(1 + delta + (c1.^(-1/delta)));
            f=cdf1.*pdf;

         case {'BB1'} % (Joe and Hu, 1996)
            x= ((u.^(-zeta))-1 ).^delta;
            y= ((v.^(-zeta))-1 ).^delta;
            a=(1+(x+y).^(1/delta)).^((-1/zeta)-2);
            b=(x+y).^((1/delta)-2);
            c=zeta*(delta-1) + (zeta*delta +1).*(x+y).^(1/delta) ;
            d=(x.*y).^(1-(1/delta));
            e=(u.*v).^(-zeta-1);
            f= a.*b.*c.*d.*e;
         case {'BB2'} % (Joe and Hu, 1996)
            x= exp(delta.*((u.^(-zeta))-1))-1;
            y= exp(delta.*((v.^(-zeta))-1))-1;
            a= x+y+1;
            h= ( 1 + (delta.^(-1)).*log(a)).^((-1/zeta)-2) ;
            i= a.^(-2);
            j= 1+zeta+ (zeta*delta) + zeta.*log(a);
            k= x+1;
            l= y+1;
            m= (u.*v).^(-zeta-1);
            f= h.*i.*j.*k.*l.*m;
```

```
        case {'BB3'} % (Joe and Hu, 1996)
            x= exp(delta.*(-log(u)).^zeta)-1;
            y= exp(delta.*(-log(v)).^zeta)-1;
            a=log(x+y+1);
            G_bar= exp(-((1/delta).*a).^(1/zeta));

            i=a.^((1/zeta)-2);
            j= (x+y+1).^(-2);
            k1=(zeta-1).*delta.^(1/zeta);
            k2=zeta.*(delta.^(1/zeta)).*a;
            k3= a.^(1/zeta);
            k= k1 + k2 + k3 ;
            l= delta^(2-(2/zeta));
            m= x+1;
            n= y+1;
            o= (-log(u)).^(zeta-1);
            p= (-log(v)).^(zeta-1);
            q= 1./(u.*v);
            f= G_bar.*i.*j.*k.*l.*m.*n.*o.*p.*q;
        case {'BB4'} % (Joe and Hu, 1996)
            x=((u.^(-zeta))-1).^(-delta);
            y=((v.^(-zeta))-1).^(-delta);
            a= 1+ (x.^(-1/delta)) + (y.^(-1/delta)) - ((x+y).^(-1/
                delta));

            h= a.^((-1/zeta)-2);
            i= (x.*y).^(1+(1/delta));
            j= ( u.*v ).^(-zeta-1);
            k1= zeta+1;
            k2= (x.^((-1/delta)-1)) - ((x+y).^((-1/delta)-1) );
            k3= (y.^((-1/delta)-1)) - ((x+y).^((-1/delta)-1) );
            k4= zeta.*(1+delta);
            k5= a.*((x+y).^((-1/delta)-2));
            k= (k1.*k2.*k3) + (k4.*k5);

            f= h.*i.*j.*k;
        case {'BB5'} % (Joe and Hu, 1996)
            x= -log(u);
            y= -log(v);
            a1= (x.^(-zeta*delta)) + (y.^(-zeta*delta));
            t= (x.^zeta)+ (y.^zeta) - (a1.^(-1/delta));
            G_bar= exp(-t.^(1./zeta));
            a=x; b=y;
            z1 = (a.^(zeta-1)) - ((a.^(-zeta*delta)) + b.^(-zeta*
                delta)).^((-1/delta)-1).*...
                 (a.^(-(zeta*delta)-1));

            h= 1./(u.*v);
            i= G_bar;
            j= t.^((1/zeta)-2);

            k1= (t.^(1/zeta)) + zeta-1;
            a=y; b=x;
            z2 = (a.^(zeta-1)) - ((a.^(-zeta*delta)) + b.^(-zeta*
                delta)).^((-1/delta)-1).*...
                 (a.^(-(zeta*delta)-1));
```

```
            k2= zeta.*(1+delta).*t;
            k3= (a1.^((-1/delta)-2)).*(x.^(-(zeta*delta)-1)).*(y
                .^(-(zeta*delta)-1));
            k= (k1.*z1.*z2) + (k2.*k3);

            f= h.*i.*j.*k;
        case {'BB6'} % (Joe and Hu, 1996)
            x= -log(1-(1-u).^zeta);
            y= -log(1-(1-v).^zeta);
            w= exp(-(x.^delta + y.^delta).^(1/delta));
            a=(x.^delta + y.^delta);
            u_bar= 1-u; v_bar=1-v;

            h= (1-w).^((1/zeta)-2);
            i= w;
            j= a.^((1/delta)-2);
            k1= (zeta-w).*(a.^(1/delta));
            k2= (zeta.*(delta-1).*(1-w));
            k= k1+k2;
            l= ( x.*y ).^(delta-1);
            m= 1./(1-u_bar.^zeta );
            n= 1./(1-v_bar.^zeta );
            p= ( u_bar.*v_bar ).^(zeta-1);
            f= h.*i.*j.*k.*l.*m.*n.*p;
        case {'BB7'} % (Joe and Hu, 1996)
            x= ((1-(1-u).^zeta).^(-delta))-1;
            y= ((1-(1-v).^zeta).^(-delta))-1;

            h= (1-(x+y+1).^(-1/delta) ).^((1/zeta)-2);
            i= (x+y+1).^((-1/delta)-2);
            j= ((x+1).*(y+1)).^(1+(1/delta));
            k1= zeta.*(delta+1);
            k2= ((zeta*delta)+1).*(x+y+1).^(-1/delta);
            k= k1-k2;
            l= ((1-u).*(1-v)).^(zeta-1);
            f= h.*i.*j.*k.*l;
        case {'BB8'} % (Joe 1996)
             eta=1-(1-delta).^zeta;
             x=1-(1-(delta.*u)).^zeta;
             y=1-(1-(delta.*v)).^zeta;
             a1=1-((1./eta).*x.*y);

             h= delta/eta;
             j= a1.^((1/zeta)-2);
             k= zeta-(1/eta).*x.*y ;
             l= (1-(delta.*u)).^ (zeta-1);
             m= (1-(delta.*v)).^ (zeta-1);

             f= h.*j.*k.*l.*m;
        case {'BB9','Crowder'}
             x=(delta.^(-1))-log(u);
             y=(delta.^(-1))-log(v);
             a1= x.^zeta + y.^zeta - delta.^(-zeta);
             G_bar= exp(-a1.^(1/zeta) + (delta.^(-1))) ;

             h= G_bar.*( (a1).^((1/zeta)-2) );
```

```
                i= (a1.^(1/zeta))+zeta-1 ;
                j= (x.*y).^(zeta-1);
                k= 1./(u.*v);
                f=h.*i.*j.*k;

          case {'BB10'}

                h=( 1-(delta.*(1-u.^zeta).*(1-v.^zeta)) ).^((-1/zeta)
                    -2);
                i=1-delta+...
                     (delta.*(1+zeta).*((u.*v).^(zeta)))-...
                     (delta.*(1-delta).*(1-u.^zeta).*(1-v.^zeta));
                f=h.*i;
          otherwise
               fprintf(' Abending: Input family is not in \n');

     end
end
```

```
function log_pdf = log_mvnpdf(mu,mu_star,Sigma)
m = size(mu,1);
log_pdf = -0.5*m*log(2*pi)-0.5*log(det(Sigma))-0.5*sum(((mu - mu_star
    )'/Sigma).*(mu - mu_star)',2);
```

```
function B = vechB2B(b,n,K)
B = zeros(n,K);
Dd = DuplicationM(K);
B(1:K,1:K) = tril(reshape(Dd*b(1:(0.5*K*(K+1))),K,K));
B(K+1:end,:) = reshape(b((0.5*K*(K+1))+1:end),n-K,K);
%B = vech(B)
end
```

D.3 The Algorithms in Computational Intelligence in Dynamic Portfolio Strategy with Regular Vine Based Model among the New Normal of Financial Market

This section presents the MATLAB code of the following: algorithm for the forecasting R-vine-ICAPM-EGARCH-Mixture model, the clonal selection algorithm and the genetic algorithm for the optimisation problem 5.3.3. Note that for some miscellaneous function can be called from the previous section.

D.3.1 The Algorithm for the Forecasting R-vine-ICAPM-EGARCH-Mixture Model

There are three steps for the forecasting returns.

Step 1

```
% QMLE Algorithm in R-vine-ICAPM-EGARCH-Mixture model
```

```
% Date 9 December 2022
% If you find any bugs, feel free to email me.
% By Rewat Khanthaporn
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

% EGARCH Mean Mixture
clear;

% input area

% an asset return (T by n) consists of variables as follow:
% ret_crypto_train and ret_stocks_train, also the date of the two
    assets
% date_crypto_train
% date_stocks_train
[sz_crypto,n_crypto]=size(ret_crypto_train);
[sz_stock,n_stock]=size(ret_stocks_train);
global rto;
n=n_crypto+n_stock;
p=8;
clear AIC1 BIC1;
for i=1:n
        if i <= n_stock
            rto=ret_stocks_train(:,i);
        else
            rto=ret_crypto_train(:,i-n_stock);
        end
        run Empirical_QMLE_EGARCH_M_MixNrLfRtGPD.m


end
```

Step 2

```
% QMLE Algorithm in R-vine-ICAPM-EGARCH-Mixture model
% Date 9 December 2022
% If you find any bugs, feel free to email me.
% By Rewat Khanthaporn
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

% copula family
% Matlab toolbox:
%        - 'Clayton'       = 1    used
%        - 'Frank'         = 2    used
%        - 'Gumbel'(EV)    = 3    used
% My coding:  1-parameter Archimedean copula
%        - 'Joe'/'B5'      = 4    used
%        - 'Galambos'(EV)  = 5    used
% 2-parameters copula
%        - 'BB1'           = 6    used
%        - 'BB2'           = 7    not used
%        - 'BB3'           = 8    used
%        - 'BB4'           = 9    used
%        - 'BB5'           = 10   not used
%        - 'BB6'           = 11   used
%        - 'BB7'           = 12   used
```

```
%         - 'BB8'           = 13   not used
%         - 'BB9','Crowder' = 14   not used
%         - 'BB10'          = 15   used
% Elliptical copula
%         - 'Gaussian'      =16    used
%         - 't'             =17    used
clear
global family_list
para_num = [ 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2];
family_list= ["Clayton","Frank","Gumbel","B5","Galambos" ...
             ,"BB1","BB2","BB3","BB4","BB5","BB6","BB7","BB8"...
             ,"BB9","BB10","Gaussian","t"];
global fix_fun
fix_fun=[1:5,16,17,6,8,9,11,12,15];
num_cop_fam=length(fix_fun);

%
global zt
global i
global k
global R_vine_family
global u
global v
clear zt
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BEGIN: convert ret to unif(0,1)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% x_margin1=x_margin(1,1:end);
% rt=ret(:,1:end);
zt=fn_Empirical_conversion_EGARCH111_rt_2_zt_wt_crypto(
    ret_crypto_train',ret_stocks_train',x_margin);

T = sz_crypto;
% get zt above only weekday
ii=1;
for i=1:T
    pos=find(date_stocks_train==date_crypto_train(i));
    if ~isempty(pos)
        for j=1:n_stock
            z=zt{j,1};
            z_temp(ii,j)=z(pos,1);
        end
        for j=n_stock+1:n_stock+n_crypto
            z=zt{j,1};
            z_temp(ii,j)=z(i,1);
        end
        date_test(ii,1)=date_crypto_train(i);
        ii=ii+1;
    end
end

for j=1:n_stock+n_crypto
    zt{j,1}=z_temp(:,j);
end
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% BEGIN: Find 1st tree structure using Kendal's Tau
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic;
Empir_R_vine_m_qmle=zeros(n,n);
Empir_R_vine_m_qmle=fn_Empirical_R_vine_m_from_Tau(zt,
    Empir_R_vine_m_qmle);

% fit data to copula for 1st tree
vdirect=cell(num_cop_fam,n,n);
vindirect=cell(num_cop_fam,n,n);
z2=cell(num_cop_fam,n,n);
z1=cell(num_cop_fam,n,n);

Empir_vdirect=cell(n,n);
Empir_vindirect=cell(n,n);
Empir_z2=cell(n,n);
Empir_z1=cell(n,n);
R_vine_family=cell(n,n);
for j=1:num_cop_fam
    ct=n;
    for i=1:n
        vdirect{j,n,i}=zt{i,1}';
        ct=ct-1;
    end
end
for i=1:n
    Empir_vdirect{n,i}=zt{i,1}';
end
clear M;
    for k=n:-1:1
        for i=n:-1:n
             M(i,k) = max(Empir_R_vine_m_qmle(i:n,k));
        end
    end
%
% 1st tree: cal each dependance pair for all candidate cop list
% and pick cop fam best fit to the data
clear Cmpre_BIC_qmle Cmpre_AIC_qmle Cmpre_loglik_neg_qmle
ct_est=1;
for j=1:num_cop_fam
    i=n;
    for k=n-1:-1:1
        R_vine_family{i,k}=family_list{fix_fun(j)};
        z1(j,i,k)=vdirect(j,i,k);
        if M(i,k)==Empir_R_vine_m_qmle(i,k)
            z2(j,i,k)=vdirect(j,i,n-M(i,k)+1);
        else
            z2(j,i,k)=vindirect(j,i,n-M(i,k)+1);
        end
        A = []; b = [];
        Aeq = [];  beq = [];
        R_vine_typ=fix_fun(j);
        [lb, ub] = fn_R_vine_PCC_bound_para_empir(R_vine_typ,1,1);
        lb = lb(2:end);  ub = ub(2:end);
        v=z1{j,i,k};
        u=z2{j,i,k};
```

```
        [u,v]=fn_Empirical_PCC_ck_size(u,v);
        if -isinf(lb)
            x0=1;
        else
            x0=lb+0.1; % get initial guess
        end
        x0=[x0 fix_fun(j)];
        [R_vine_x_qmle{j,i,k},loglik_neg_qmle,exitflag,output,lambda,
           grad,hessian]=...
                          fmincon(@fn_Empirical_QMLE_RVineCopula2,x0,A
                              ,b,Aeq,beq,lb,ub);

        fprintf('Seq QMLE Est column %i at cop fam %i \n',k,j);
        ct_est=ct_est+1;
        Covar_QMLE{j,i,k}=inv(hessian);
        % cal AIC, BIC: compare cop fam, lower is better fit
        ua=[u;v];
        sz_para=size(lb,2);
        Cmpre_loglik_neg_qmle(j,i,k) = loglik_neg_qmle;
        Cmpre_AIC_qmle(j,i,k) = (2*Cmpre_loglik_neg_qmle(j,i,k)) +
            (2*sz_para);
        Cmpre_BIC_qmle(j,i,k) = (2*Cmpre_loglik_neg_qmle(j,i,k)) + (
           sz_para*log(sz_para));
    end
end
%
clear Empir_R_vine_family_qmle Empir_R_vine_x_qmle
    Empir_R_vine_type_qmle
clear Empir_qmle_AIC Empir_qmle_BIC Empir_qmle_lglik

for k=n-1:-1:1
    tp=Cmpre_BIC_qmle(:,n,k);
    pos=find(tp==min(tp));
    if family_list{fix_fun(pos)} == 't'
       if R_vine_x_qmle{pos,n,k}(2) > 30
            Empir_R_vine_family_qmle{n,k}=family_list{fix_fun(6)};
            Empir_R_vine_type_qmle(n,k)=16;
            Empir_R_vine_x_qmle{n,k}=R_vine_x_qmle{6,n,k};
       else
            Empir_R_vine_family_qmle{n,k}=family_list{fix_fun(pos)};
            Empir_R_vine_type_qmle(n,k)=fix_fun(pos);
            Empir_R_vine_x_qmle{n,k}=R_vine_x_qmle{pos,n,k};
       end
    else
        Empir_R_vine_family_qmle{n,k}=family_list{fix_fun(pos)};
        Empir_R_vine_type_qmle(n,k)=fix_fun(pos);
        Empir_R_vine_x_qmle{n,k}=R_vine_x_qmle{pos,n,k};
    end
    Empir_qmle_AIC(n,k)=Cmpre_AIC_qmle(pos,n,k);
    Empir_qmle_BIC(n,k)=Cmpre_BIC_qmle(pos,n,k);
    Empir_qmle_lglik(n,k)=-Cmpre_loglik_neg_qmle(pos,n,k);
end

i=n;
for k=n-1:-1:1
    Empir_z1(i,k)=Empir_vdirect(i,k);
```

```
    if M(i,k)==Empir_R_vine_m_qmle(i,k)
        Empir_z2(i,k)=Empir_vdirect(i,n-M(i,k)+1);
    else
        Empir_z2(i,k)=Empir_vindirect(i,n-M(i,k)+1);
    end
    v=Empir_z1{i,k};
    u=Empir_z2{i,k};
    [u,v]=fn_Empirical_PCC_ck_size(u,v);
    fam=Empir_R_vine_family_qmle{i,k};
    para=Empir_R_vine_x_qmle{i,k};
    Empir_vdirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);

    u=Empir_z1{i,k};
    v=Empir_z2{i,k};
    [u,v]=fn_Empirical_PCC_ck_size(u,v);
    Empir_vindirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);

end

%
aa=1:13;
global zt
global i
global k
global R_vine_family
global u
global v
tic;
for k=n-2:-1:1 % colum operation
    for i=n-1:-1:k+1 % row operation
        % update Empir_R-vine-m and M tree (maximum spanning tree)
        % from highest empirical tau
        [Empir_R_vine_m_qmle,M]=fn_Empirical_m_finding(i,k,
            Empir_R_vine_m_qmle,M,...
                                                Empir_vdirect,
                                                    Empir_vindirect);
        % find all candidate from empir cop fam
        Empir_z1(i,k)=Empir_vdirect(i,k);
        if M(i,k)==Empir_R_vine_m_qmle(i,k)
            Empir_z2(i,k)=Empir_vdirect(i,n-M(i,k)+1);
        else
            Empir_z2(i,k)=Empir_vindirect(i,n-M(i,k)+1);
        end
        clear Cmpre_BIC_qmle1 R_vine_x_qmle_can loglik_neg_qmle1
            Cmpre_AIC_qmle1
        v=Empir_z1{i,k};
        u=Empir_z2{i,k};
        [u,v]=fn_Empirical_PCC_ck_size(u,v);
        % est all can cop fam and BIC
        lg=length(fix_fun);
        for j=1:lg
            [R_vine_x_qmle_can{j},loglik_neg_qmle1(j),sz_para]=...
                                         fn_Empir_fmincon_r_vine(aa(1,j
                                             ));
            R_vine_x_qmle_can{j}=R_vine_x_qmle_can{j}(1:end-1);
```

```
            Cmpre_AIC_qmle1(j) = (2*loglik_neg_qmle1(j)) + (2*sz_para);
            Cmpre_BIC_qmle1(j) = (2*loglik_neg_qmle1(j)) + (sz_para*log(sz_para));
        end
        % choose min BIC
        pos=find(Cmpre_BIC_qmle1==min(Cmpre_BIC_qmle1));

        if family_list{fix_fun(pos)} == 't'
           if R_vine_x_qmle_can{pos}(2) > 30
                Empir_R_vine_family_qmle{i,k}=family_list{fix_fun(6)};
                Empir_R_vine_type_qmle(i,k)=16;
                Empir_R_vine_x_qmle{i,k}=R_vine_x_qmle_can{6};
           else
                Empir_R_vine_family_qmle{i,k}=family_list{fix_fun(pos)};
                Empir_R_vine_type_qmle(i,k)=fix_fun(pos);
                Empir_R_vine_x_qmle{i,k}=R_vine_x_qmle_can{pos};
           end
        else
            Empir_R_vine_family_qmle{i,k}=family_list{fix_fun(pos)};
            Empir_R_vine_type_qmle(i,k)=fix_fun(pos);
            Empir_R_vine_x_qmle{i,k}=R_vine_x_qmle_can{pos};
        end
        Empir_qmle_AIC(i,k)=Cmpre_AIC_qmle1(pos);
        Empir_qmle_BIC(i,k)=Cmpre_BIC_qmle1(pos);
        Empir_qmle_lglik(i,k)=-loglik_neg_qmle1(pos);

        v=Empir_z1{i,k};
        u=Empir_z2{i,k};
        [u,v]=fn_Empirical_PCC_ck_size(u,v);
        fam=Empir_R_vine_family_qmle{i,k};
        para=Empir_R_vine_x_qmle{i,k};
        Empir_vdirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);

        u=Empir_z1{i,k};
        v=Empir_z2{i,k};
        [u,v]=fn_Empirical_PCC_ck_size(u,v);
        Empir_vindirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);
    end
end

%
for k=n-1:-1:1
    for i=n:-1:k+1
        Empir_R_vine_x_qmle0=Empir_R_vine_x_qmle{i,k};
        if length(Empir_R_vine_x_qmle0) == 3
            Empir_R_vine_x_qmle{i,k}=Empir_R_vine_x_qmle0(1:2);
        end
    end
end
```

Step 3

```
% QMLE Algorithm in R-vine-ICAPM-EGARCH-Mixture model
```

```
% Date 9 December 2022
% If you find any bugs, feel free to email me.
% By Rewat Khanthaporn
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

% ***************************************
% Description
% ***************************************
% Output of 1 is
% --------------
% x is zt: 100 sims and each sim (zt) means that row = assets, column =
% Time (up to 3 months).

vol_t_stock=std(ret_train_stock');
vol_t_crypto=std(ret_train_crypto');

T=size(wkdy_ret_stocks_test,1);
sim=2000;
n=n_crypto+n_stock;

burn=100;
rng default
unif0=unifrnd(0,1,sim,n,burn+T); %sim by n by T (30 assets by 3 months)

x_margin_qmle_egarch=x_margin;

m=Empir_R_vine_m_qmle;
R_vine_family=Empir_R_vine_family_qmle;
R_vine_P=Empir_R_vine_x_qmle;
%
% *********************
% 1 gen zt using dependenc r vine copula paras
% *********************
for sm =1:sim
    z=squeeze(unif0(sm,:,burn+1:end)); % n by T = 30 assets by 3 months
    for j=1:n
        z1{n,j}=z(m(j,j),:);
        z2{n,j}=z(m(n,j),:);
        vdirect{n,j}=z(j,:);
    end
    x(sm,1,:)=vdirect{n,n};
    for k=n-1:-1:1
        for i=k+1:n
            if M(i,k)==m(i,k)
                z2(i,k)=vdirect(i,n-M(i,k)+1);
            else
                z2(i,k)=vindirect(i,n-M(i,k)+1);
            end
            v=vdirect{n,k};
            u=z2{i,k};
            rd= fncopularnd_inv_h_fun_1(R_vine_family{i,k},R_vine_P{i,k},T,v,u);
            vdirect{n,k}=rd(2,:);
```

```
        end
        x(sm,n-k+1,:)=vdirect{n,k};
        for i=n:-1:k+1
            z1{i,k}=vdirect{i,k};
            v=z1{i,k};
            u=z2{i,k};
            para=R_vine_P{i,k};
            fam=R_vine_family{i,k};
            vdirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);
            u=z1{i,k};
            v=z2{i,k};
            vindirect{i-1,k}= fncopularnd_h_fun(fam,para,u,v);
        end
    end

end

for sm=1:sim
    zt=squeeze(x(sm,:,:)); % n by T
    for j_lp=1:n
            para=x_margin_qmle_egarch(j_lp,:);
            z=zt(j_lp,:)';
            mu_t=[0    ; 0 ; 0];
            xir_t=para(2);
            betar=(1-xir_t)*(1-(2*xir_t))^0.5; %  beta
            xil_t=para(1);
            betal=(1-xil_t)*(1-(2*xil_t))^0.5; %  beta
            lamda1 = para(3);
            lamda2 = para(4);
            omega_t  = para(5);
            gam_t    = para(6);
            delta_t  = para(7);
            nu_t     = para(8);
            true1_EGARCH_Mixnrgpd = [ xil_t xir_t lamda1 lamda2
                omega_t gam_t delta_t nu_t ];
            sd_t=[1.0    ; betar ; betal];
            p_t =[0.8  ; 0.1 ; 0.1];
            nn=1; nl=1;
            z_1=zeros(T+1,1);
            ht=zeros(T+1,1);
            rto=zeros(T+1,1);
            if j_lp <= n_stock
                rto(1,1)= ret_train_stock(j_lp,end);
                ht(1,1)=vol_t_stock(1,j_lp);
            else
                rto(1,1)= ret_train_crypto(j_lp-n_stock,end);
                ht(1,1)=vol_t_crypto(1,j_lp-n_stock);
            end
            z_1(1,1)=(rto(1,1)-lamda1-(lamda2*ht(1,1)))/sqrt(ht(1,1))
                ;

            for i=1:T
                if z(i) >= 0.9 % right tail
                    tmp1=z(i);
                    z_1(i+1,1)=(sd_t(2)/xir_t).*(((-tmp1+1).^-xir_t)
                        -1) + mu_t(2);
```

```
                elseif z(i) <= 0.1 % left tail
                    tmp1=z(i);
                    z_1(i+1,1) = mu_t(3) + (sd_t(3)/xil_t)*(-tmp1^-xil_t+1);
                else
                    tmp1=z(i);
                    z_1(i+1,1)=norminv(tmp1);
                end
                ht(i+1,1) = omega_t + gam_t*z_1(i,1) + delta_t*abs(z_1(i,1)) + nu_t*(ht(i,1));
                rto(i+1,1) = lamda1 + lamda2*exp(ht(i+1,1)) + sqrt(exp(ht(i+1,1)))*z_1(i+1,1);
            end
            rt(sm,j_lp,:)=rto(2:T+1,1);
    end
end

for a=1:n
    rt_reformat(a,:,:)=rt(:,a,:);
end
rt=rt_reformat;
wo=0;

run Selected_assets_N_get_wkday_data_fundamental.m
```

D.3.2 The Clonal Selection Algorithm for the Optimisation Problem 5.3.3

```
% Date 9 December 2022
% If you find any bugs, feel free to email me.
% By Rewat Khanthaporn
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

% **************************************************************************
% Forecasting: R-vine-ICAPM-EGARCH-Mixture model
% Machine learning:          CSA
% Asset:                     30 stock + 5 crypto
% **************************************************************************

port_benchmark=Port_net_ret_wo_crypto;

wo=0; % flag to include crypto assets
alpha =1;
rt_out=wkdy_exp_ret;
asset_index=wkdy_asset_index;
master_ret_test= [wkdy_ret_stocks_test wkdy_ret_crypto_test]';
n_crypto=5;
n=35;
```

```
n_stock=30;
[~,~,Tt]=size(rt_out);
if wo==1

    clear wkdy_asset_index_wo_crypto;
    for i=1:Tt
        pos=find(wkdy_asset_index(i,:)==0);
        if isempty(pos)
            wkdy_asset_index_wo_crypto(i,:)=wkdy_asset_index(i,1:n-n_crypto);
        else
            wkdy_asset_index_wo_crypto(i,1:pos-n_crypto-1)=wkdy_asset_index(i,1:pos-n_crypto-1);
        end
    end

    master_ret_test= wkdy_ret_stocks_test';
    rt_out=wkdy_exp_ret(:,1:n_stock,:);
    asset_index=wkdy_asset_index_wo_crypto;
end
clear vfx_out vmfit_out;

mx=size(asset_index,2);

loop=1;
tol=1e-03; break_even=3;
for lp=1:loop
    tic;
    for d=1:Tt

        nAssets=size(find(asset_index(d,:) > 0),2);
        R=squeeze(rt_out(:,1:nAssets,d));

        var_assets = -prctile(R, alpha); % = 99% confidence level
        for i=1:nAssets
            Ri=R(:,i);
            cvar_assets(i,1) = -mean(Ri(Ri < -var_assets(i)));
        end
        lb=zeros(nAssets,1);up=zeros(nAssets,1)+1;        % x bound
        rg=up-lb;
        % gen random 0 1
        Ll=66; crm=100;
        [x1,x2] = meshgrid(0:0.05:1,0:0.05:1); vxp = x1; vyp = x2;
        vzp = eval(f_plot);

        run CSA_AIS_min_CVaR_subfun.m
        x = valx(1,:);
        itrp(lp,d)=it;
        vfx_out(d,:)=vfx;
        vmfit_out(d,:)=vmfit;
        if ~isempty(find(vfx==0,1))
            vfx=vfx(1:find(vfx==0,1)-1);
            vmfit=vmfit(1:find(vmfit==0,1)-1);
        end

        fit_ret=R*x';
```

```
        var_t = -prctile(fit_ret, alpha);
        cvar_t = -mean(fit_ret(fit_ret < -var_t)); % CVaR 99%
        sd=std(fit_ret);

        w=x;
        if nAssets < mx
            s=size(w,2);
            x=[w zeros(1,mx-s)];
        end

        out_CSA_min_CVaR(lp,d,:)=[mean(fit_ret) cvar_t var_t sd x];

    CSA_cvar_Port_net_ret(lp,1)=10000;
    for i=1:d
        assets=size(find(asset_index(i,:) > 0),2);
        port_ret_cvar_csa=squeeze(out_CSA_min_CVaR(lp,i,5:assets+4))
           '*master_ret_test(asset_index(i,1:assets),i);
        CSA_cvar_Port_net_ret(lp,i+1)=CSA_cvar_Port_net_ret(lp,i)
           *(1+(port_ret_cvar_csa/100));
    end
    end
end

csa_port_cvar=squeeze(out_CSA_min_CVaR(1,:,:)); % lp,day,:
for i=1:loop
    for d=1:Tt
        can=squeeze(out_CSA_min_CVaR(i,d,:))';
        if can(1,1)/can(1,2) > csa_port_cvar(d,1)/csa_port_cvar(d,2)
            csa_port_cvar(d,:)=can;
            lst(d)=i;
        end
    end
end

csa_cvar_Port_net(1,1)=10000;
for i=1:Tt
    assets=size(find(asset_index(i,:) > 0),2);
    port_ret_cvar_csa(i,1)=csa_port_cvar(i,5:assets+4)*
       master_ret_test(asset_index(i,1:assets),i);
    csa_cvar_Port_net(1,i+1)=csa_cvar_Port_net(1,i)*(1+(
       port_ret_cvar_csa(i,1)/100));
end
```

D.3.3 The Genetic Selection Algorithm for the Optimisation Problem 5.3.3

```
% Date 9 December 2022
% If you find any bugs, feel free to email me.
% By Rewat Khanthaporn
% Email: rkhantha@aut.ac.nz or raysuge044@hotmail.com

```

```
% ************************************************************************
% Forecasting: R-vine-ICAPM-EGARCH-Mixture model
% Machine learning:          GA
% Asset:                     30 stock + 5 crypto
% ************************************************************************
port_benchmark=Port_net_ret_wo_crypto;

alpha =1; % = alpha =0.99

rt_out=wkdy_exp_ret;
asset_index=wkdy_asset_index;
master_ret_test= [wkdy_ret_stocks_test wkdy_ret_crypto_test]';

[~,~,T]=size(rt_out);
mx=size(asset_index,2);
clear vx_out vmfit_out;
loop=1;
for lp=1:loop
for d=1:T
    nAssets=size(find(asset_index(d,:) > 0),2);
    R=squeeze(rt_out(:,1:nAssets,d));
    expected_rt=mean(R);
    covar=cov(R);
    var_assets = -prctile(R, alpha); % = 99% confidence level
    for i=1:nAssets
        Ri=R(:,i);
        cvar_assets(i,1) = -mean(Ri(Ri < -var_assets(i)));
    end
    Ll=66;% 66;
    cross_pos=round(Ll*nAssets*0.45);    % cross over 45%
    crm=1000;
    rk=1;
    x=[];
    f = '(-1).*(x*cvar_assets)';
    riskport=0;
    run ga_Ray_min_CVaR_N_dims_subfun.m
    itrp(lp,d)=itr;
    xo = x(indb,:);
    vx_out(lp,:)=vx;
    vmfit_out(lp,:)=vmfit;
    portReturns = R * xo';
    sd=std(portReturns);
    var = -prctile(portReturns, alpha);
    cvar = -mean(portReturns(portReturns < -var));
    w=xo;
    if nAssets < mx
        s=size(w,2);
        xo=[w zeros(1,mx-s)];
    end
    out_ga_cvar(lp,d,:)=[mean(portReturns) cvar var sd xo];
    figure(10)
    subplot(2,1,1);
```

```
    plot(vx); title('Min. fun')
    subplot(2,1,2)
    plot(vmfit); title('Mean fun')

    GA_cvar_Port_net_ret(lp,1)=10000;
    for i=1:d
        assets=size(find(asset_index(i,:) > 0),2);
        port_ret_cvar=squeeze(out_ga_cvar(lp,i,5:assets+4))'*
            master_ret_test(asset_index(i,1:assets),i);
        GA_cvar_Port_net_ret(lp,i+1)=GA_cvar_Port_net_ret(lp,i)*(1+(
            port_ret_cvar/100));
    end

end

end
GA_port_cvar=squeeze(out_ga_cvar(1,:,:)); % lp,day,:
for i=1:loop
    for d=1:T
        can=squeeze(out_ga_cvar(i,d,:))';
        if can(1,1)/can(1,2) > GA_port_cvar(d,1)/GA_port_cvar(d,2)
            GA_port_cvar(d,:)=can;
            lst(d)=i;
        end
    end
end

GA_cvar_Port_net(1,1)=10000;
for i=1:T
    assets=size(find(asset_index(i,:) > 0),2);
    port_ret_cvar(i,1)=GA_port_cvar(i,5:assets+4)*master_ret_test(
        asset_index(i,1:assets),i);
    GA_cvar_Port_net(1,i+1)=GA_cvar_Port_net(1,i)*(1+(port_ret_cvar(i
        ,1)/100));
end
```

D.3.4 The Miscellaneous Functions

```
P = cadeia(crm,Ll*nAssets,0,0,0);

% ger = 2; n = size(P,1); pm = 0.01; per = 0.0; fat = .10;
ger = 300; n = size(P,1); pm = 0.01; per = 0.0; fat = .10;
% Hypermutation controlling parameters
pma = pm; itpm = ger; pmr = 0.8;

x=[];
for i=1:nAssets
    bgn=1+(i-1)*Ll;
    ed=bgn + Ll - 1;
    x(:,i) = decode(P(:,bgn:ed),lb(i),rg(i),Ll);
end
x=x./sum(x,2);
x1=x(:,1); x2=x(:,2);
```

```
fit = eval(f_plot);

% General defintions
vpm = []; vfx = []; vmfit = []; valfx = 1;
[N,L] = size(P); it = 0; PRINT = 1;
dif=1;ct_break=0;

% Generations
x=[]; % cal weight, 1st random one.
parfor i=1:nAssets
    bgn=1+(i-1)*Ll;
    ed=bgn + Ll - 1;
    x(:,i) = decode(P(:,bgn:ed),lb(i),rg(i),Ll);
end
x=x./sum(x,2);
cs = [];T = [];
port_exp_ret=mean(R*x');
fit_ret=R*x';
var_fit = -prctile(fit_ret, alpha);
fit=[];
parfor abc=1:n
    ft=fit_ret(:,abc);
    fit(abc,1) = mean(ft(ft < -var_fit(abc))); % CVaR 99%
end
gap=[fit -port_exp_ret'./abs(fit) port_exp_ret' var_fit']; % ret per CVaR
[a,ind] = sort(gap(:,2)); % sort by the highest ret per cvar
valx = x(ind(end-n+1:end),:);
fx = gap(ind(end-n+1:end),1);

while it <= ger
    x1=x(:,1); x2=x(:,2);
   % Reproduction
    [T,pcs] = reprod(n,fat,N,ind,P,T);
   % Hypermutation
    M = rand(size(T,1),L) <= pm;
    T = T - 2 .* (T.*M) + M;
    T(pcs,:) = P(ind(end-n+1:end),:);
   % New Re-Selection (Multi-peak solution)
    x=[];
    parfor i=1:nAssets
        bgn=1+(i-1)*Ll;
        ed=bgn + Ll - 1;
        x(:,i) = decode(T(:,bgn:ed),lb(i),rg(i),Ll);
    end
    x=x./sum(x,2);
    port_exp_ret=mean(R*x');
    fit_ret=R*x';
    var_fit = -prctile(fit_ret, alpha);
    parfor abc=1:size(var_fit,2)
        ft=fit_ret(:,abc);
        fit(abc,1) = mean(ft(ft < -var_fit(abc))); % CVaR 99%
    end
    pcs = [0 pcs];
    out_AIS_minCVaR1=[]; bcs =[];
    gap0=[];
```

```
    gap0=[fit -port_exp_ret'./abs(fit) port_exp_ret' var_fit']; %

    for i=1:n
      pos=find(gap0(pcs(i)+1:pcs(i+1),2) == min(gap0(pcs(i)+1:pcs(i+1),2)));
      bcs(i)=pos(1);
      bcs(i) = bcs(i) + pcs(i);
      out_AIS_minCVaR1(i) = gap0(bcs(i),1);              % Maximizationn problem
    end
    P(ind(end-n+1:end),:) = T(bcs,:);
   % Editing (Repertoire shift)
    nedit = round(per*N); it = it + 1;
    P(ind(1:nedit),:) = cadeia(nedit,L,0,0,0);
    pm = pmcont(pm,pma,pmr,it,itpm);

    % cal weight after done all processes.
    x=[];
    parfor i=1:nAssets
        bgn=1+(i-1)*Ll;
        ed=bgn + Ll - 1;
        x(:,i) = decode(P(:,bgn:ed),lb(i),rg(i),Ll);
    end
    x=x./sum(x,2);
    cs = [];T = [];
    port_exp_ret=mean(R*x');
    fit_ret=R*x';
    var_fit = -prctile(fit_ret, alpha);
    fit=[];
    parfor abc=1:n
        ft=fit_ret(:,abc);
        fit(abc,1) = mean(ft(ft < -var_fit(abc))); % CVaR 99%
    end
    gap=[fit -port_exp_ret'./abs(fit) port_exp_ret' var_fit'];
    [a,ind] = sort(gap(:,2)); % sort by the closest VCaR target

    valx = x(ind(end-n+1:end),:);  % the best weight
    fx   = gap(ind(end-n+1:end),1);% the best CVaR at 1st
    vpm = [vpm pm]; vfx = [vfx fx(1)]; vmfit = [vmfit mean(fit)];
    if it > 1
       dif=abs(vfx(it)-vfx(it-1));
    else
       dif=1;
    end
   if dif < tol
        ct_break=ct_break+1;
        if ct_break > break_even
            brk_code(lp)=1;
            break;
        end
   else
        ct_break=0;
   end
end
vfx=[vfx zeros(1,ger-size(vfx,2)+1)];
vmfit=[vmfit zeros(1,ger-size(vmfit,2)+1)];
```

```
% MLE for the model of EGARCH-in-Mean with Mix of Normal, GPD to the
    left and right tails

% input area
% x0 is initial guass (1 by 8)

fun=@fnMLE_EGARCH_MixNrLfRtGPD_empiri;

A = [];
b = [];
Aeq = [];
beq = [];

lb = [ 0.00,0.00,-0.1,-0.00,-0.5,-1.00,-1.00,0.4];
ub = [ 0.50,0.50, 0.3,0.3 , 0.5, 1.00,1.00,  1];

[x_margin(i,:),fval(i),exitflag,output,lambda,grad,hessian]=fmincon(
    fun,x0,A,b,Aeq,beq,lb,ub);

covar{i}=inv(hessian);
```

```
function z=fn_CDF_Conversion_mix_Nor_GPD(x,betar,xir,betal,xil)

sz=size(x,1);
q_R=norminv(0.9);    % upper quantial at 0.9, this can be changed
q_L=norminv(0.1);    % lower quantial at 0.1, this can be changed
for i=1:sz
    if x(i) >= q_R % right tail
        z(i,1) = -(x(i)*xir/betar + 1 )^(-1/xir) + 1 ;
    elseif x(i) <= q_L % left tail
        z(i,1) = (-(xil/betal*x(i)) +1)^(-1/xil);
    else
        z(i,1)=normcdf(x(i));
    end
end

z=U_transf(z);
```

```
function zt=fn_Empirical_conversion_EGARCH111_rt_2_zt_wt_crypto(
    ret_crypto_train,ret_stocks_train,x_margin1)

n=size(ret_crypto_train,1)+size(ret_stocks_train,1);
zt=cell(n,1);
for i=1:n
    x1=x_margin1(i,:);
    if i <= size(ret_stocks_train,1)
        rt_run=ret_stocks_train(i,1:end)';
    else
        rt_run=ret_crypto_train(i-size(ret_stocks_train,1),1:end)';
    end

    T=size(rt_run,1);
    xil_1=x1(1);
```

```
    xir_1=x1(2);
    lambda1_1=x1(3);
    lambda2_1=x1(4);
    omega_1=x1(5);
    gam_1=x1(6);
    delta_1=x1(7);
    nu_1=x1(8);
    betar_1=(1-xir_1)*(1-(2*xir_1))^0.5; %  beta
    betal_1=(1-xil_1)*(1-(2*xil_1))^0.5; %  beta
    z1_1=zeros(T,1); hts_1=zeros(T,1);
    hts_1(1,1)=omega_1;
    z1_1(1,1)=(rt_run(1,1) - lambda1_1 - (lambda2_1*exp(hts_1(1,1))))
       /sqrt(exp(hts_1(1,1)));
    for t=2:T
        hts_1(t,1)=omega_1 + gam_1*z1_1(t-1,1) + delta_1*abs(z1_1(t
           -1,1)) + nu_1*(hts_1(t-1,1));
        z1_1(t,1)=(rt_run(t,1) - lambda1_1 - (lambda2_1*exp(hts_1(t
           ,1))))/sqrt(exp(hts_1(t,1)));
    end

    % fun to convert z1 above to unif
    zt{i,1}=fn_CDF_Conversion_mix_Nor_GPD(z1_1,betar_1,xir_1,betal_1,
       xil_1);
end
```

```
function [rt_out, asset_index]=fn_select_asset_fundamental(rt,n_stock
   ,n_crypto,wo,fun_select)

if fun_select==1
    selected_list=[1,3,4,6:9,11:13,15:16,18:23,25:30];
else
    selected_list=1:35;
end
[asset,sm,T]=size(rt);
rt_out=zeros(sm,asset,T);
asset_index=zeros(T,asset);
for t=1:T
    close all;
    R=squeeze(rt(:,:,t))';
    expected_rt=mean(R); % forecaste
    covar=cov(R);   % 1 day forecaste by student't copula
    plot(diag(covar),expected_rt,'*'); hold on;

    rskRet=expected_rt./diag(covar)';
    plot(diag(covar),rskRet,'o'); hold on;
    rsk_all=diag(covar)';
    p=1;
    clear pick ;
    for i=1:n_stock
        rsk=rsk_all(i);
        pos=find(rsk_all < rsk);
        pos1=find(expected_rt(pos) > expected_rt(1,i));
        if find(i==selected_list,1)
            pick(p,:)=[expected_rt(1,i) rsk];
            rt_out(:,p,t)=rt(i,:,t); %rt(t,:,i);
            asset_index(t,p)=i;
```

```
            p=p+1;
        end
    end
    if wo~=1
        for i=1:n_crypto % reformat
            rt_out(:,p,t)=rt(n_stock+i,:,t);
            asset_index(t,p)=n_stock+i;
            p=p+1;
        end
    end
    exist pick;
    if ans ~= 0
        pick_st=sort(pick);
        plot(pick_st(:,2),pick_st(:,1),'+-'); hold off;
    else
        exist pick;
    end
end
```

```
function [wkdy_exp_ret,wkdy_asset_index,wkdy_ret_stocks_test,
    wkdy_ret_crypto_test,...
          wkdy_date] = fn_select_ex_ret_weekday(rt_out,asset_index
              ,...
          ret_stocks_test,ret_crypto_test,date_stocks_test,
              date_crypto_test,...
          n_stock,n_crypto)

T=size(date_crypto_test,1);
wd=1;
for d=1:T
    weekday=find(date_stocks_test==date_crypto_test(d));
    if weekday
        wkdy_exp_ret(:,:,wd)=rt_out(:,:,wd);
        wkdy_asset_index(wd,:)=asset_index(wd,:);
        wkdy_ret_stocks_test(wd,:)=ret_stocks_test(weekday,:);
        wkdy_ret_crypto_test(wd,:)=ret_crypto_test(d,:);
        wkdy_date(wd,1)=date_stocks_test(weekday,1);
        wd=wd+1;
    end
end
```

```
function f=fnMLE_EGARCH_MixNrLfRtGPD_empiri(x0)

global rto
rt=rto;
xil=x0(1);
xir=x0(2);
lambda1=x0(3);
lambda2=x0(4);
omega=x0(5);
gam=x0(6);
delta=x0(7);
nu=x0(8);
p=[0.8;0.1;0.1];

```

```
betar=(1-xir)*(1-(2*xir))^0.5; %  beta
betal=(1-xil)*(1-(2*xil))^0.5; %  beta

n=size(rt,1);
z1=zeros(n,1); hts=zeros(n,1);
hts(1,1)=omega;
z1(1,1)=(rt(1,1) - lambda1 - (lambda2*exp(hts(1,1))))/sqrt(exp(hts(1,1)));

for t=2:n
    hts(t,1)=omega + gam*z1(t-1,1) + delta*abs(z1(t-1,1)) + nu*(hts(t-1,1));
    z1(t,1)=(rt(t,1) - lambda1 - (lambda2*exp(hts(t,1))))/sqrt(exp(hts(t,1)));
end

et=z1.*sqrt(exp(hts));
%normal
lik(:,1)= pdf('normal',et,0,sqrt(exp(hts)));
lik(:,2)=pdf('Generalized Pareto',et,xir,betar.*sqrt(exp(hts)),0);
lik(:,3)=pdf('Generalized Pareto',-et,xil,betal.*sqrt(exp(hts)),0);

lik_w=lik.*p';
f=sum(lik_w,2);
f= -1*sum(log(f));

end
```

```
lb=zeros(nAssets,1);up=zeros(nAssets,1)+1;        % x bound
rg=up-lb;
% gen random 0 1
v = cadeia(crm,Ll*nAssets,0,0,0);

[N,L] = size(v); ger = 300; p_cross = 0.5; p_mutate = 0.03;
tol=1e-05;break_even=2;
vzp = eval(f1);
PRINT = 1;
% General parameters & Initial operations
sol = 1; %
vmfit = [];
vx = [];
itr = 1;
C = [];
sol_fit=[];
port_exp_ret=[];
for i=1:nAssets
    bgn=1+(i-1)*Ll;
    ed=bgn + Ll - 1;
    x(:,i) = decode(v(:,bgn:ed),lb(i),rg(i),Ll);
end
x=x./sum(x,2);
port_exp_ret=mean(R*x');
fit=R*x';
var_fit = -prctile(fit, alpha);
parfor abc=1:N
```

```
    ft=fit(:,abc);
    sol_fit(abc,1) = mean(ft(ft < -var_fit(abc)));
end

p_top = 0.5; p_tail = 0.1; p_rest = 1 - (p_top + p_tail);
n_top = round(p_top * N); n_tail = round(p_tail * N); n_rest = round(p_rest * N);
if (n_top + n_tail + n_rest) ~= N
   dif = N - (n_top + n_tail + n_rest);
   n_rest = n_rest + dif;
end
c=3;
% Generations
sol_pre=1; ct_break=0;
clear gap;
while itr <= ger
   % Reproduction
   port_exp_ret=mean(R*x');
   gap=[sol_fit abs(sol_fit-riskport) -abs(port_exp_ret'./sol_fit)];
   [rw,ind] = sort(gap(:,c));
   ft_sort=[sol_fit(ind) port_exp_ret(ind)' abs(port_exp_ret(ind)'./sol_fit(ind))];
   vtemp = [v(ind(1:n_top),:); v(ind(end-n_tail+1:end),:); v(2:n_rest+1,:)];

   %  Crossover
   rd=rand(N,1);
   C(:,1) = rd <= p_cross;
   C(:,2) = round(cross_pos.*rd)+1;
   I = find(C(:,1) == 1);
   IP = [I,C(I,2)];
   for i = 1:size(IP,1)
      v(IP(i,1),:) = [vtemp(IP(i,1),1:IP(i,2)) vtemp(1,IP(i,2)+1:end)];
   end

   %    Mutation
   M = rand(N,L) <= p_mutate;
   M(1,:) = zeros(1,L);
   v = v - 2 .* (v.*M) + M; % mutate at M = 1 except for 1st row of v because highest chance

   % Results
  parfor i=1:nAssets
        bgn = 1+(i-1)*Ll;
        ed  = bgn + Ll - 1;
        x(:,i) = decode(v(:,bgn:ed),lb(i),rg(i),Ll);
  end
    x=x./sum(x,2);
    fit=R*x';
    var_fit = -prctile(fit, alpha);
    parfor abc=1:N
        ft=fit(:,abc);
        sol_fit(abc,1) = mean(ft(ft < -var_fit(abc)));
    end
```

```
    port_exp_ret=mean(R*x');
    gap=[sol_fit abs(sol_fit-riskport) -abs(port_exp_ret'./sol_fit)];
    [rw_f,ind_f] = sort(gap(:,c));
    indb=ind_f(1);
    sol_pre=sol;
    portReturns = R * x(indb,:)';
    sd=std(portReturns);
    var = -prctile(portReturns, alpha);
    sol = -mean(portReturns(portReturns < -var));
     v(1,:) = v(indb,:);
     x(1,:) = x(indb,:);
     sol_fit(1,:)=sol_fit(indb,:);
     media = mean(sol_fit);
     vx = [vx sol]; vmfit = [vmfit media];
     itr = itr + 1;
     if abs(sol-sol_pre) < tol
         ct_break=ct_break+1;
         if ct_break > break_even
             brk_code(lp)=1;
             break;
         end
     else
         ct_break=0;
     end
end
vx=[vx zeros(1,ger-size(vx,2))];
vmfit=[vmfit zeros(1,ger-size(vmfit,2))];
```

```
clear;

port_benchmark=Port_net_ret_wo_crypto;
% ----------------------------------------
% Opt: GA stock wt crypto
% obj fun: min CVaR at alpha = 0.99
% ----------------------------------------

alpha =1; % = alpha =0.99
run Empirical_fore_Opt_GA_CVaR_AR_garch_evt_t_cop_wt_crypto.m

% #### THIS IS SAVED OUTPUT AREA. ###

%%
clear;
alpha =1;

cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\Port_Opt\GA'

load('Port_Pre_covid_GAmin_CVaR_icapm_egarchmixRvine_itrp_wt_crypto','itrp')
load('Port_Pre_covid_GAmin_CVaR_icapm_egarchmixRvine_vx_out_wt_crypto','vx_out')
load('Port_Pre_covid_GAmin_CVaR_icapm_egarchmixRvine_vmfit_out_wt_crypto_','vmfit_out')

```

```
load('Port_Tbill3mns_preco','Tbill3mns_preco')
Tbill3mns=Tbill3mns_preco;
load('
    Port_Pre_covid_rvine_icapm_egarch_mix_wkdy_asset_index_wt_crypto',
    'wkdy_asset_index')
asset_index=wkdy_asset_index;
load('
    Port_Pre_covid_GAmin_CVaR_icapm_egarchmixRvine_port_ret_cvar_wt_crypto_alpa99_fun_
    ','port_ret_cvar')
load('
    Port_Pre_covid_GAmin_CVaR_icapm_egarchmixRvine_wt_crypto_alpa99_fun_selection
    ','out_ga_cvar')
load('
    Port_Pre_covid_GAmin_CVaR_icapm_egarchmixRvine_Port_net_wt_crypto_alpa99_fun_selec
    ','GA_cvar_Port_net')
load('
    Port_Pre_covid_GAmin_CVaR_icapm_egarchmixRvine_tim_ga_wt_crypto_alpa99_fun_select
    ','tim_ga_cvar')
% cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\
    Port_Opt\GA'
% load('Port_egarchmixRvine_wkdy_asset_index_wt_crypto_fun_selection
    ','wkdy_asset_index')
% asset_index=wkdy_asset_index;
% cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\
    Port_Opt\CLONAL_AIS'
a1=squeeze(out_ga_cvar);

port_ret=port_ret_cvar;
% out=[mean(port_ret) std(port_ret)]

var_assets = -prctile(port_ret, alpha); % = 99% confidence level
cvar_assets = -mean(port_ret(port_ret < -var_assets));
% out1=[cvar_assets var_assets]

Ratio_SR=mean((port_ret-Tbill3mns)/std(port_ret));
pos=find(port_ret<0);
Ratio_Sortino=mean((port_ret-Tbill3mns)/std(port_ret(pos)));
% out2=[Ratio_Sortino Ratio_SR]

nAssets=size(find(asset_index(1,:) > 0),2);
side_buy=0;
side_sell=0;
count_no_turnover=0;
day=size(a1,1);
for i=2:day
    for j=5:4+nAssets
        if a1(i,j) > a1(i-1,j)
            side_buy=side_buy+1000*(a1(i,j) - a1(i-1,j));
        elseif a1(i,j) < a1(i-1,j)
            side_sell=side_sell+1000*(a1(i-1,j) - a1(i,j));
        else
            count_no_turnover=count_no_turnover+1;
        end
    end
end
side_buy=side_buy/day-1;
```

```
side_sell=side_sell/day-1;
if side_sell < side_buy
    avg_turnover=side_sell/mean(GA_cvar_Port_net);
else
    avg_turnover=side_buy/mean(GA_cvar_Port_net);
end

out=[mean(port_ret) std(port_ret) Ratio_Sortino Ratio_SR cvar_assets
    var_assets mean(port_ret)/cvar_assets ...
     mean(port_ret)/var_assets avg_turnover GA_cvar_Port_net(1,end)
        tim_ga_cvar/246]
%%
% ----------------------------------------
% Opt: GA wt crypto
% obj fun: min VaR at alpha= .99
% ----------------------------------------
% cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\
    Port_Opt\GA'
alpha=1; %99 confident interval
run Empirical_fore_Opt_GA_VaR_AR_garch_evt_t_cop_wt_crypto.m

% #### THIS IS SAVED OUTPUT AREA. ###

clear;
alpha =1;

cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\
    Port_Opt\GA'

load('
    Port_Pre_covid_GA_min_VaR_icapm_egarchmixRvine_itrp_wt_crypto_alpa99_fun_selection
    ','itrp')
load('
    Port_Pre_covid_GA_min_VaR_icapm_egarchmixRvine_vx_out_wt_crypto_alpa99_fun_select
    ','vx_out')
load('
    Port_Pre_covid_GA_min_VaR_icapm_egarchmixRvine_vmfit_out_wt_crypto_alpa99_fun_sele
    ','vmfit_out')

load('Port_Tbill3mns_preco','Tbill3mns_preco')
Tbill3mns=Tbill3mns_preco;
load('
    Port_Pre_covid_rvine_icapm_egarch_mix_wkdy_asset_index_wt_crypto',
    'wkdy_asset_index')
asset_index=wkdy_asset_index;
load('
    Port_Pre_covid_GA_min_VaR_icapm_egarchmixRvine_port_ret_cvar_wt_crypto_alpa99_fun_
    ','port_ret_var')
load('
    Port_Pre_covid_GA_min_VaR_icapm_egarchmixRvine_out_wt_crypto_alpa99_fun_selection
    ','out_ga_var')
load('
    Port_Pre_covid_GA_min_VaR_icapm_egarchmixRvine_Port_net_wt_crypto_alpa99_fun_sele
    ','GA_var_Port_net')
```

```
load('
    Port_Pre_covid_GA_min_VaR_icapm_egarchmixRvine_tim_ga_wt_crypto_alpa99_fun_selecti
    ','tim_ga_var')
% cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\
    Port_Opt\GA'
% load('Port_egarchmixRvine_wkdy_asset_index_wt_crypto_fun_selection
    ','wkdy_asset_index')
% asset_index=wkdy_asset_index;
% cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\
    Port_Opt\CLONAL_AIS'
a1=squeeze(out_ga_var);

port_ret=port_ret_var;
% out=[mean(port_ret) std(port_ret)]

var_assets = -prctile(port_ret, alpha); % = 99% confidence level
cvar_assets = -mean(port_ret(port_ret < -var_assets));
% out1=[cvar_assets var_assets]

Ratio_SR=mean((port_ret-Tbill3mns)/std(port_ret));
pos=find(port_ret<0);
Ratio_Sortino=mean((port_ret-Tbill3mns)/std(port_ret(pos)));
% out2=[Ratio_Sortino Ratio_SR]

nAssets=size(find(asset_index(1,:) > 0),2);
side_buy=0;
side_sell=0;
count_no_turnover=0;
day=size(a1,1);
for i=2:day
    for j=5:4+nAssets
        if a1(i,j) > a1(i-1,j)
            side_buy=side_buy+1000*(a1(i,j) - a1(i-1,j));
        elseif a1(i,j) < a1(i-1,j)
            side_sell=side_sell+1000*(a1(i-1,j) - a1(i,j));
        else
            count_no_turnover=count_no_turnover+1;
        end
    end
end
side_buy=side_buy/day-1;
side_sell=side_sell/day-1;
if side_sell < side_buy
    avg_turnover=side_sell/mean(GA_var_Port_net);
else
    avg_turnover=side_buy/mean(GA_var_Port_net);
end

out=[mean(port_ret) std(port_ret) Ratio_Sortino Ratio_SR cvar_assets
    var_assets mean(port_ret)/cvar_assets ...
     mean(port_ret)/var_assets avg_turnover GA_var_Port_net(1,end)
        tim_ga_var/246]

%%
% ----------------------------------------
% Opt: GA with crypto
```

```
% obj fun: Max Sharpe Ratio
% ----------------------------------------
%
alpha=1; % not impact to the algor just cal var and cvar at given max sharpe ratio
run Empirical_fore_Opt_GA_Sharpe_AR_garch_evt_t_cop_wt_crypto.m

% #### THIS IS SAVED OUTPUT AREA. ###

%%
clear;
alpha =1;

cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\Port_Opt\GA'

load('Port_Pre_covid_GA_max_sharpe_icapm_egarchmixRvine_itrp_select_asset_fun_selection','itrp')
load('Port_Pre_covid_GA_max_sharpe_icapm_egarchmixRvine_vx_out_select_asset_fun_selectio','vx_out')
load('Port_Pre_covid_GA_max_sharpe_icapm_egarchmixRvine_vmfit_out_select_asset_fun_selec','vmfit_out')

load('Port_Tbill3mns_preco','Tbill3mns_preco')
Tbill3mns=Tbill3mns_preco;
load('Port_Pre_covid_rvine_icapm_egarch_mix_wkdy_asset_index_wt_crypto','wkdy_asset_index')
asset_index=wkdy_asset_index;
load('Port_Pre_covid_GA_max_sharpe_icapm_egarchmixRvine_port_ret_sharpe_select_asset_fun','port_ret_sharpe')
load('Port_Pre_covid_GA_max_sharpe_icapm_egarchmixRvine_out_select_asset_fun_selection','out_ga_sharpe')
load('Port_Pre_covid_GA_max_sharpe_icapm_egarchmixRvine_Port_net_select_asset_fun_select','GA_sharpe_Port_net')
load('Port_Pre_covid_GA_max_sharpe_icapm_egarchmixRvine_tim_select_asset_fun_selection','tim_ga_sharpe')
% cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\Port_Opt\GA'
% load('Port_egarchmixRvine_wkdy_asset_index_wt_crypto_fun_selection','wkdy_asset_index')
% asset_index=wkdy_asset_index;
% cd 'C:\Users\rkhantha\OneDrive - AUT University\AUT\PhD\Matlab\Port_Opt\CLONAL_AIS'
a1=squeeze(out_ga_sharpe);

port_ret=port_ret_sharpe;
% out=[mean(port_ret) std(port_ret)]
```

```
var_assets = -prctile(port_ret, alpha); % = 99% confidence level
cvar_assets = -mean(port_ret(port_ret < -var_assets));
% out1=[cvar_assets var_assets]

Ratio_SR=mean((port_ret-Tbill3mns)/std(port_ret));
pos=find(port_ret<0);
Ratio_Sortino=mean((port_ret-Tbill3mns)/std(port_ret(pos)));
% out2=[Ratio_Sortino Ratio_SR]

nAssets=size(find(asset_index(1,:) > 0),2);
side_buy=0;
side_sell=0;
count_no_turnover=0;
day=size(a1,1);
for i=2:day
    for j=5:4+nAssets
        if a1(i,j) > a1(i-1,j)
            side_buy=side_buy+1000*(a1(i,j) - a1(i-1,j));
        elseif a1(i,j) < a1(i-1,j)
            side_sell=side_sell+1000*(a1(i-1,j) - a1(i,j));
        else
            count_no_turnover=count_no_turnover+1;
        end
    end
end
side_buy=side_buy/day-1;
side_sell=side_sell/day-1;
if side_sell < side_buy
    avg_turnover=side_sell/mean(GA_sharpe_Port_net);
else
    avg_turnover=side_buy/mean(GA_sharpe_Port_net);
end

out=[mean(port_ret) std(port_ret) Ratio_Sortino Ratio_SR cvar_assets
    var_assets mean(port_ret)/cvar_assets ...
     mean(port_ret)/var_assets avg_turnover GA_sharpe_Port_net(1,end)
         tim_ga_sharpe/246]
```

```

crypto_name_new= {'BTC-USD',...
'ETH-USD',...
'USDT-USD',...
'BNB-USD',...
'XRP-USD'};

[sz_crypto,n_crypto]=size(ret_crypto_train);
[sz_stock,n_stock]=size(ret_stocks_train);

exist fun_select
fun_select=ans;
[rt_out_wt_crypto, asset_index_wt_crypto]=fn_select_asset_fundamental
    (rt,n_stock,n_crypto,wo,fun_select);

[wkdy_exp_ret,wkdy_asset_index,wkdy_ret_stocks_test,
    wkdy_ret_crypto_test,...
                  wkdy_date]=fn_select_ex_ret_weekday...
```

```
                    (rt_out_wt_crypto,asset_index_wt_crypto,
                       ret_stocks_test...
                    ,ret_crypto_test,date_stocks_test,date_crypto_test
                       ...
                    ,n_stock,n_crypto);
```