Accepted to Evolving Systems, Springer-Nature

NeuDen: A Framework for the Integration of Neuromorphic Evolving Spiking Neural Networks with Dynamic Evolving Neuro-Fuzzy Systems for Predictive and Explainable Modelling of Streaming data

Iman AbouHassan* [1,2] and Nikola K. Kasabov* [3, 4,5]

1. *Technical University of Sofia, Sofia, Bulgaria.*
2. *Central Bank of Lebanon, Beirut, Lebanon.*
3. *KEDRI, SECMS, Auckland University of Technology, New Zealand.*
4. *IICT, Bulgarian Academy of Sciences, Sofia, Bulgaria.*
5. *Dalian University of Technology, Dalian, China.*

**** Contact persons:***

Nikola Kasabov (nkasabov@aut.ac.nz) , ORCID id: 0000-0003-4433-k7521

Iman AbouHassan (iabouhassan@tu-sofia.bg; iabouhassan@bdl.gov.lb) , ORCID id: 0000-0001-5552-1311

ABSTRACT:

This paper introduces a novel framework, called here 'NeuDen' for the integration of neuromorphic evolving spiking neural networks (eSNN), that learn efficiently multiple time series in their temporal association and interaction, with dynamic evolving neuro-fuzzy systems (deNFS), that learn incrementally extracted from the eSNN feature vectors, to predict future time-series values and to produce interpretable fuzzy rules. The new framework aims to make the best out of the dominant features of the two types of models. First, spike-time-dependent plasticity (STDP) learning is used in SNN to learn temporal interaction between multiple time series, connected to a dynamic eSNN (deSNN) as a regressor/classifier. Then, feature-vectors are extracted from the trained deSNN for further learning, fuzzy inference and rule extraction in a deNFS, here exemplified by DENFIS, resulting in an accurate prediction results and explainable dynamic fuzzy rules. The NeuDen, framework and model, overcomes both the explainability problems of eSNN and the limitations of deNFS to model multiple streaming time series in their temporal interaction. NeuDen surpasses both deSNN and DENFIS by providing multiple regression models and achieving higher accuracy. NeuDen is demonstrated on bench mark data and on financial and economic time series, achieving from 3 to 100 times smaller RMSE when compared with other evolving systems. The proposed framework opens a new direction for the development of more efficient evolving systems by integrating eSNN with other methods, such as other neuro-fuzzy systems, deep neural networks and quantum classifiers for specific applications.

Keywords: *NeuDen, Evolving Spiking Neural Networks, Evolving Connectionist Systems, DENFIS, NeuCube.*

1. INTRODUCTION:

In many cases, bringing two computational paradigms together in one system, leads to improved performance. Here, two evolving systems paradigms are integrated for a better predictive and explainable modelling of streaming data. And these are evolving spiking neural networks (eSNN) and dynamic evolving neuro-fuzzy systems (deNFS). Both belong to the class of evolving connectionist systems (ECOS), but each of them deals with principally different information representation and uses principally different learning algorithms. In this paper, we propose a new framework, called NeuDen, for the purpose of such integration and demonstrate that the integration leads to superior results in prediction accuracy and explainability. The paper opens a new direction for the integration of SNN, so efficient in dealing with multiple streaming data (Indiveri 2009,2010,2011; Furber 2012; Delbruck 2007; Gerstner 1995,2002; Buonomano & Maass 2009; Kasabov 2014; Koprinkova 2023), with other computational models, more efficient when dealing with vector-based data and knowledge discovery (Kohonen 1990, Carpenter & Grossberg 1991; Amari 1967,1990; Zadeh 1965,1988; Angelov et al. 2004,2010; Bengio 2009; Bezdek 1987). The proposed framework can be theoretically and technologically linked to enhance and be enhanced by other popular methods for time-space information processing (e.g., Kasabov, 2019; Chen 2013; Dey 2022; Hebb 1949; Hopfield 1982,1995; Hussain 2013; Izhikevich 2003,2004,2006; Kosko 1988; LeCun 2015; Lee 2016; Li 2021; Neftci 2011; Petro 2019; Reid 2014 a,b; Schmidhuber 2015; Song 2000; Szatmary 2010; Thorpe 2001; Verstraeten 2007; Wu 2018; Yager 1994; Yamakawa 1992; Yan 2013; Yang 2011).

ECOS, first introduced by Kasabov in 1998, represent a class of intelligent systems characterized by their dynamic, adaptive algorithms that evolve over time (Kasabov 1998). ECOS are designed and structured to incrementally learn and evolve their functionality from incoming data, supporting online learning and continuous enhancement without forgetting previously acquired information (Kasabov 2003; Tan 2020; Watts 2009). ECOS algorithms have earned significant attention of many researchers and academics for their rapid processing capabilities, proving to be highly effective in various real-world applications, often outperforming other artificial neural network approaches that have been extensively studied (Abouhassan et al. 2022; 2023a, b). This paper introduces NeuDen, an innovative ECOS-based hybrid model. NeuDen enhances the ECOS framework by leveraging fuzzy inference within spatio-temporal learning strategies for multiple streaming data using the neuromorphic spiking neural network (SNN) approach. This innovative approach enables NeuDen to efficiently process and learn from data in dynamic environments, showcasing significant advancements in the field of evolving systems.

Some hybrid models of SNN and traditional neural networks have been investigated and proposed. An architecture combining an SNN and a classical artificial neural network (ANN) for event-based optical flow estimation (Negi 2023) is intended to integrate the advantages of both SNNs and ANNs. Nonetheless, it failed to address the issue of the vanishing gradient problem and the complexity introduced by spiking layers. Another hybrid model that integrates the SNN (NeuCube) and Echo State Network (ESN) mechanisms is proposed to facilitate real-time learning and classification of EEG data (Koprinkova 2023). However, its complicated design demands meticulous hyperparameter adjustments, posing potential delays in optimization, having also problems with explainability. A hybrid deep + non-deep learning model is proposed to improve the classification of low-volume, high-dimensional data (Mavaie 2023). However, it requires more data for effective training, and it is largely dependent on the selection of feature extraction layers, and performs inconsistently across various datasets. A hybrid model for building energy optimization that integrates discrete and continuous data is proposed (Banihashemi 2017), but faces limitations in applicability across different machine learning algorithms requiring broader validation with larger data samples. To leverage the inference accuracy of graphical models with the dynamic learning capabilities of deep neural networks, the authors (Gormley 2016) of hybrid models envision a future in which these two powerful models converge, potentially creating a more efficient and accurate system for analyzing complex data. This necessitates harmonizing model structures and optimizing learning processes.

Thus, this paper introduces a new NeuDen framework and a model, inspired by the ECOS principles. NeuDen integrates neuro-fuzzy inference with spatio-temporal learning in eSNN. The rationale for developing NeuDen is to address the limitations of standard ECOS algorithms, particularly their inability to process spatio-temporal data, and to capture their temporal patterns of interaction over time.

Following this introduction, Section 2 presents the NeuDen model's proposed framework and its algorithms and illustrates the model on a simple benchmark time series data. Sections 3 and 4 exemplify the model's robustness and accuracy on real case studies from the economic and financial domains. Section 5 offers a discussion and future directions.

2. NEUDEN MODEL FRAMEWORK AND ALGORITHMS

The NeuDen framework consists of three main parts. The first part is dedicated to spatio-temporal modelling of dynamic data using SNN. The intermediary section converts spike-time data from a trained SNN into a feature-vector format, facilitating the transformation of classified samples of time series into a structured feature-vector form. The last section utilizes a dynamic evolving neuro-fuzzy system (deNFS) for learning the extracting feature-vectors to produce a fuzzified regression outcome and to report fuzzy rules. The NeuDen principal diagram is shown in Fig.1a, while Fig.1b shows the information flow.

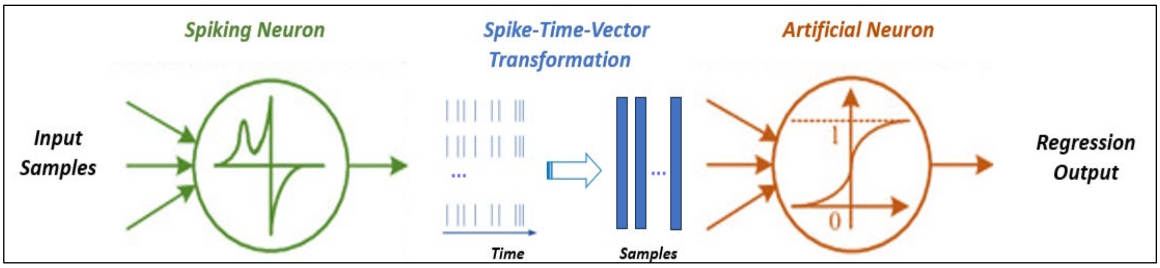


Fig.1a. The NeuDen Overall Principal Diagram

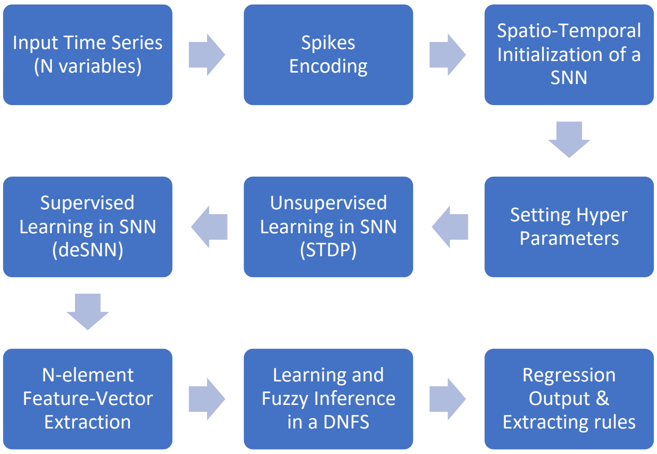


Fig.1b. Information flow in the NeuDen model

To demonstrate our model, we have selected exemplary systems based on ECOS principles (Kasabov 2019). For the spatio-temporal learning we employ the SNNcube for unsupervised learning as part of the NeuCube brain-inspired SNN architecture (Kasabov 2014); for feature-vector extraction we use the connections learned in a supervised mode in the deSNN classifier/regressor connected to the SNN cube in the NeuCube architecture; and for the neuro-fuzzy inference system we use the popular DENFIS model (Kasabov and Song 2002) (Fig. 2).

2.1. Spike-time learning in the eSNN

SNN have been evolving in the last decade or so as efficient models for temporal data modelling (Bothe, Maass, Gerstner). Evolving SNN (eSNN) are first introduced in (Kasabov 2007) and then further developed and used in many applications (see Kasabov 2019, Schliebs 2014). Dynamic eSNN (deSNN) are introduced in (Kasabov 2013) and also used as classifier/regressor in the NeuCube architecture (Kasabov 2014). The NeuCube architecture consists of four main parts:

- spike encoding, where continuous time series are encoded into sequences of spikes;
- unsupervised learning in a 3D structured SNNcube using a spike time algorithm, such as spike-time dependent plasticity (STDP);
- supervised learning in a classifier/regressor deSNN (Kasabov et al. 2013);
- model parameter optimization.

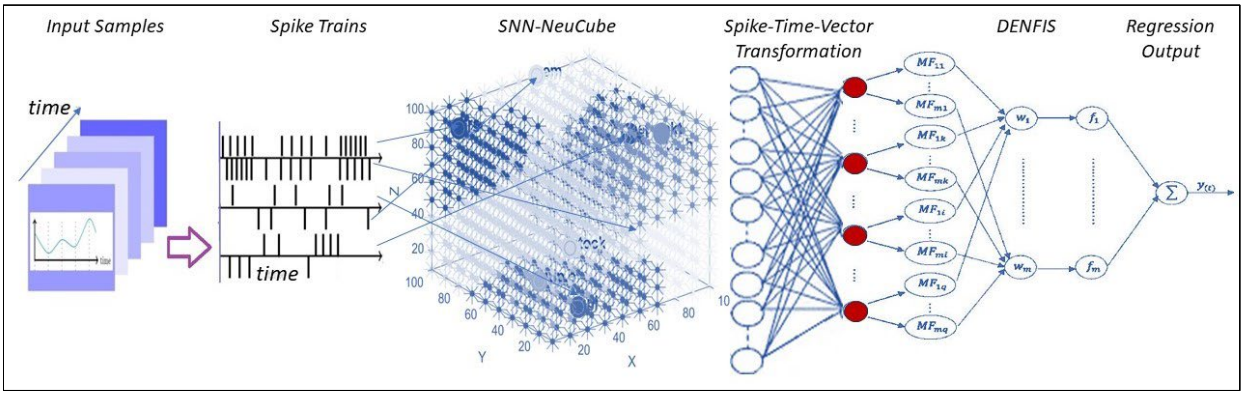


Fig.2: NeuDen Exemplified by the use of the NeuCube eSNN and DENFIS.

STDP is a Hebbian form of plasticity that involves long-term potentiation (LTP) and depression (LTD) based on the timing difference between pre- and post-synaptic spikes (Song 2000) between two connected spiking neurons. STDP allows neurons to learn consecutive temporal associations from data and adjusts synaptic weights based on the relative timing of spikes, which can lead to the formation of meaningful patterns of neural connectivity reflecting the temporal association between input time series. When a pre-synaptic neuron fires a short time before the post-synaptic neuron ($\Delta t > 0$), the synaptic weight increases (Long-Term Potentiation or LTP). Conversely, if the post-synaptic neuron fires before the pre-synaptic neuron ($\Delta t < 0$), the synaptic weight decreases (Long-Term Depression or LTD).

$$\boldsymbol{\Delta\omega} = \begin{Bmatrix} +A^{+}e^{\frac{-\Delta t}{\tau+}} & \text{for } \Delta t > 0 \\ -A^{-}e^{\frac{+\Delta t}{\tau-}} & \text{for } \Delta t < 0 \end{Bmatrix} \tag{1}$$

where: Δw is the change in synaptic weight, $\Delta t = t_{post} - t_{pre}$ represents the time difference between the post-synaptic spike time (t_{post}) and the pre-synaptic spike time (t_{pre}), A+ and A− are the maximum amount of potentiation and depression, $\tau+$ and $\tau-$ are time constants.

2.2. Dynamic Evolving SNN (deSNN):

The Dynamic Evolving Spiking Neural Network (deSNN) (Kasabov et al. 2013) model enhances the traditional Spike-Time Dependent Plasticity (STDP) by incorporating dynamic synapses using Rank-Order (RO) learning and the Spike Driven Synaptic Plasticity (SDSP) learning rules.

The output neurons here are evolved to capture the activity of the SNNcube during the propagation of a whole input sample of many time points of all input temporal variables). The RO learning rule sets an initial weight configuration based on the earliest spikes, which are considered to carry the most significant information of the input pattern.

$$\omega_{ij} = \alpha . mod^{order(ij)} \tag{2}$$

Here, ω_{ij} is the synaptic weight between a neuron *i* and output neuron *j*. The *order* parameter represents the rank order of spikes to this neuron, α is a learning parameter and *mod* is a constant between 0 and 1. The larger *mod* is, the more importance is assigned to the first coming spikes.

The Spike Driven Synaptic Plasticity (SDSP) learning introduces a weight drift mechanism where weights can incrementally increase or decrease over time, allowing for a more adaptable representation of temporal patterns. Once the synaptic weight ω_{ji} is initialized, it becomes dynamic and adjusts its weight through the SDSP algorithm.

$$\Delta\omega_{ij}(\mathrm{t}) = e_i(t). D \tag{3}$$

where, $e_i(t)$ indicates the presence of a spike at synapse *i* at time t, and D is the drift parameter.

DeSNN performs adaptive, evolving incremental learning. As new patterns are presented, new neurons and connections can be formed, or the connection weights of existing ones can be adjusted, allowing the network to continually evolve and refine its structure and weights throughout its lifetime.

After supervised learning is completed in the deSNN for one (whole) temporal input sample, its connection weights are extracted to form a feature vector for a followed neuro-fuzzy inference in DENFIS.

2.3. Dynamic evolving neural-fuzzy inference system (DENFIS): learning and rule extraction

DENFIS (Kasabov and Song 2002) uses fuzzy inference, evolving clustering methods, and adaptive rule updating. It allows efficient and flexible modeling of complex temporal sequences. The learning mechanism in DENFIS online model includes:

(a) Evolving clustering and formulation of fuzzy rules

DENFIS is first clustering the input data using evolving clustering method (ECM) and then employs Takagi-Sugeno fuzzy inference mechanism (Takagi 1985). Each rule is defined to capture the data within one cluster by a set of fuzzy propositions forming the rule's antecedents and a polynomial function as the consequent. For instance, if x1 is R11, and x2 is R12, ..., and xq is R1q, then y is f1(x1, x2, …, xq), where x1, x2, …, xq are the input variables. The rules are formulated based on fuzzy membership functions, with inputs being converted into membership degrees for each fuzzy set.

$$\mu(x) = \mathrm{mf}(\mathrm{x}; \mathrm{a}, \mathrm{b}, \mathrm{c}) = \left\{ \begin{array}{cc} 0 & if\ x \le a \\ \frac{x-a}{b-a} & if\ a < x \le b \\ \frac{c-x}{c-b} & if\ b < x < c \\ 0 & if\ x \ge c \end{array} \right\} \tag{4}$$

In one implementation, DENFIS uses triangular-type functions for all fuzzy membership functions (Kasabov 2006), depending on three parameters: a, b, and c. These parameters help in determining the shape and position of the fuzzy membership function (eq. 4), defining the degree to which each input belongs to a fuzzy set.

(b) Updating of Rule Consequents through Regression Methods

Rules are created and updated with each new data pair. A new fuzzy rule is established if a new cluster center is identified through the online ECM. The rule's antecedents are formed using the position of the cluster center, and the consequent regression functions are updated or created using a linear least-square estimator (LSE) based on the learning data. These functions can be linear or non-linear, including a neural network model.

2.4. The overall NeuDen model algorithm

(a) Spatio-temporal Spike Encoding Algorithm (SE)

Within the NeuDen model, a Temporal Contrast algorithm is employed to transform real-world input data into spike representations. This algorithm operates on a threshold-based principle (Delbruck 2007), which effectively captures significant changes in signal intensity that surpass a specified threshold. The spikes generated can be either positive or negative, depending on the sign of these changes:

$$S_i = SE\ (X(t)) \tag{5}$$

where S_i represents the encoded spike trains associated with neuron i, generated from input data X(t) at time t.

(b) Spatio-temporal Initialization and Mapping:

Initialize the spiking neurons in a 3D cube structure where each node represents a neuron. Let N_i be the neuron *i* in the cube. The encoded spike trains are then mapped to corresponding neurons in the cube. Each neuron receives input from a subset of the encoded data. The connection between each encoded spike train to corresponding neurons in the cube is based on feature similarity or spatial proximity. This connectivity can be controlled by a small radius r, determining how close in terms of spatial distance other neurons must be to connect.

$$N_i = Map\ (S_i, r) \tag{6}$$

where N_i is the neuron, and r is the radius parameter.

(c) Unsupervised training using Spike-Time Dependent Plasticity (STDP):

Apply the STDP rule to all connected neuron pairs in the NeuCube to adjust their synaptic weights based on the relative timing of spikes between pre-synaptic and post-synaptic neurons.

$$w_{ij}\,(t+1) = w_{ij}\,(t) + \Delta w \tag{7}$$

where w_{ij} (t) is the weight between neurons i and j at tim t; Δw is the change in synaptic weight.

(d) Regression Module using Dynamic Evolving Spiking Neural Network (deSNN):

Supervised learning utilizes the principles of deSNN (Kasabov 2013) to adapt the network structure and synaptic weights of the model based on the temporal dynamics of the neurons in the SNNcube. Initial weights are adjusted based on the order of spikes using Rank-Order (RO) rule, and further adjustments for the weights based on incoming spike patterns using the Spike Driven Synaptic Plasticity (SDSP) learning rule.

$$\boldsymbol{\omega}_{\boldsymbol{ji}} = \alpha.mod^{order(ji)} \tag{8}$$

$$\Delta\boldsymbol{\omega}_{\boldsymbol{ji}}(\mathrm{t}) = e_j(t). \tag{9}$$

(e) Extracting Feature-Vectors:

This is based on the learned connections in the deSNN that reflect the temporal dynamics of the neurons in the SNNcube. One feature vector is extracted for every input temporal sample of multiple time series within a time window. This transformation reflects the spike rate of each input neuron in the SNNcube over a window of time:

$$F(S) = ExtractFeatures\ (S) \tag{10}$$

where F is the feature vector extracted from input spike sequence S.

(f) Fuzzy Inference System:

Applying DENFIS to the extracted feature-vectors, so that DENFIS will cluster the transformed data F(S) in the input space using its ECM and will form automatically fuzzy rules. This will involve evolving cluster centers in the new feature space and their radius which is limited to be D_{thr} at maximum.

(g) Fuzzy membership function:

For each input feature in the transformed data F(S), a fuzzy membership function will map this feature to a membership value in [0, 1], indicating the degree to which each feature belongs to pre-defined fuzzy sets:

$$\mu x = \mathrm{mfF}(S);\mathrm{a, b, c} \tag{11}$$

(h) Regression Output:

The regression output is formulated as the weighted average of rule weights:

$$y_0 = \frac{\sum_{i=1}^{m} \omega_i \cdot f_i(\boldsymbol{F}(\boldsymbol{S}))}{\sum_{i=1}^{m} \omega_i} \tag{12}$$

where y_0 is the rule output; $f_i(F(S))$ are the polynomial output functions of the i^{th} rule for the transformed data, F(S) is the extracted feature vector from spikes S, and ω_i are the rule weights calculated based on the membership degrees.

(i) The NeuDen Training Algorithm

The NeuDen Training Algorithm is shown in Box 1.

2.5. A NeuDen model design, illustrated on a benchmark time series data

Here we use a simple, classical time series benchmark data, 'Gas Furnace' to demonstrate that even on a single time series prediction, the benefit of using the proposed method is significant. Our method of using NeuCube + DENFIS achieved RMSE of 0.0397, which is 27 times lower than the RMSE of 1.09 achieved by the use of NeuCube (SNNcube + deSNN), and 13 times lower when using DENFIS only (RMSE=0.5328). Details are explained below.

Box 1:
Begin Algorithm:

1: Set NeuDen hyperparameters:
   a. SNN parameters: spike-encoding threshold, SNN dimensions, connectivity radius, learning/validating ratio, STDP and DeSNN learning rates (α), (D).
   b. DNFS parameters: D_{thr}, clustering parameters, membership function, fuzzy rule parameters.

2: Input features Xi(t), where i=1,...,n (number of spatial features), and t=1, ..,T (number of temporal features)

3: Output: predicted output Y

4: STEPS:

Step 1: Data Preprocessing: generate samples from the input stream $X_i(t) \rightarrow M_i(t')$. t'=1 … T' (number of samples < T)

Step 2: Encode input data $M_i(t')$ into spike sequences $\mathbf{S_j}$ using a spike encoding algorithm SE.

Step 3: Initialize the 3D-SNN structure with neurons $\mathbf{N_j}$ positioned based on the input parameters.

Step 4: Map the encoded spike sequence $\mathbf{S_j}$, into the defined input neurons in the SNNcube.

Step 5: Unsupervised learning: apply STDP rule to each pair of connected neurons (i & j) in the SNNcube and adjust synaptic weights based on their spike timing: $w_{ij}(t+1) = w_{ij}(t) + \Delta w$

Step 6: Supervised learning: Adjusting synaptic weights based on the deSNN learning (RO & SDSP).

Step 7: Transform spikes to feature vector: Extract relevant features $F_i(t')$ from spike patterns as vector format.

Step 8: Initialize the fuzzy system with initial rules and membership functions.

Step 9: Input the transformed feature vector $F_i(t')$ into neuro-fuzzy model.

Step 10: Update neuro-fuzzy model, partition the input space for training and incremental learning, and update or create fuzzy rules as needed based on the evolving clustering method.

Step 11: Calculate the output as a weighted average of the fuzzy rule outputs. $Y = f(F_1, F_2, \ldots, F_n, \epsilon)$

Step 12: Calculate the training and test error of the model as a Root Mean-Square Error (RMSE) as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(\hat{y}_k^i - y_k^i\right)^2}{N}}$$

where y and $\hat{y}$ represent the original and predicted values, respectively, and N is the total number of samples. An RMSE value closer to 0, denotes the higher accuracy the model has.

End Algorithm

The 'Gas Furnace' dataset comprises 296 ordered records of methane (CH_4) and carbon dioxide (CO_2) emissions from a furnace (Box 1967). The model is to predict the output variable CO_2 at time (t) according to the following formula:

$$CO_{2(t)} = f\left(CH_{4(t-4)}, CO_{2(t-1)}\right) \tag{13}$$

The original dataset is segmented into 192 uniform samples with a sliding step of one-time step. Each sample consists of 100 time points and 2 spatial features representing input variables.

Employing a Threshold-Based Representation method, spike trains are generated from real data, at a spike rate of 26%. These spikes denote variations exceeding the threshold of 0.5 between consecutive input values (Fig. 3).

Subsequently, these spike trains are divided 70/30 for training and incremental testing, and are transferred to 1000 spatially allocated neurons in a 3D SNNcube, initialized with the use of a small world connectivity of radius 2.5 to control the initial connection weights between neighboring neurons.

Followed is the unsupervised learning in the SNNcube using parameters set as 0.01 for STDP learning rate, potential leak rate of 0.002, firing threshold of 0.5, and a refractory of 6 units of time. The neuronal connections between the hidden neurons show the temporal associations between the spiking activity of neurons where blue connections reflect the positive spikes while red for negative spikes, and most of the neurons are highly active (Fig.4).

Next, spikes data is propagated to the supervised learning stage which is controlled by the deSNN learning parameters, set at 0.4 and 0.05 for the mod and drift respectively. When using deSNN as a regressor as part of the NeuCube architecture, the RMSE is 1.09 (Fig.5). Then, feature vectors are extracted from the trained deSNN for training and inference in DENFIS. The fuzzy inference learning stage is performed using three membership functions with two learning rounds and a threshold set at 0.1. The regression output achieved in the NeuDen model shows a much lower RMSE of 0.0397 (Fig.6), which is also lower than the RMSE of 0.5328 when DENFIS was used as a single regression model using the original input features (Fig.7).

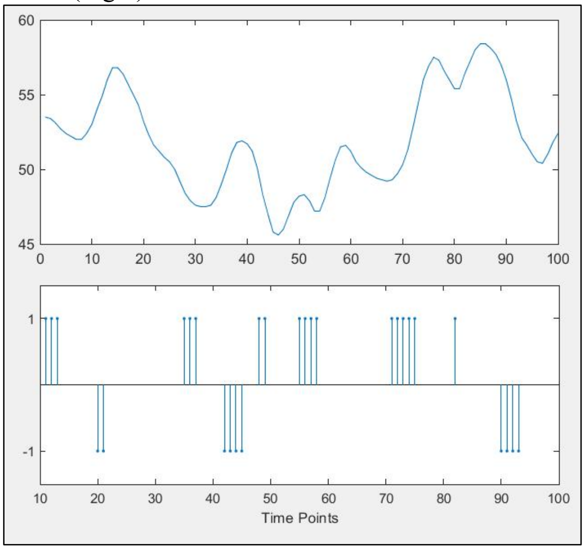


Fig. 3: CO_2 benchmark real data versus CO_2 encoded spikes in the NeuDen.

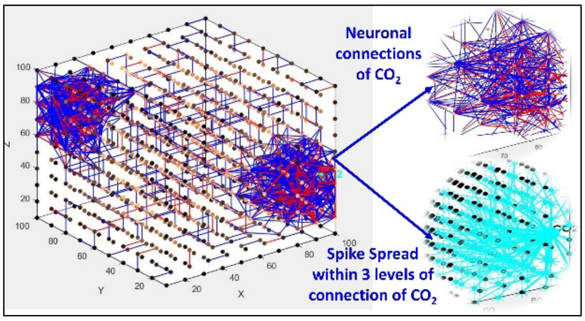


Fig. 4: The neuronal connections and spike spread activity of the two input neurons in the SNNcube.

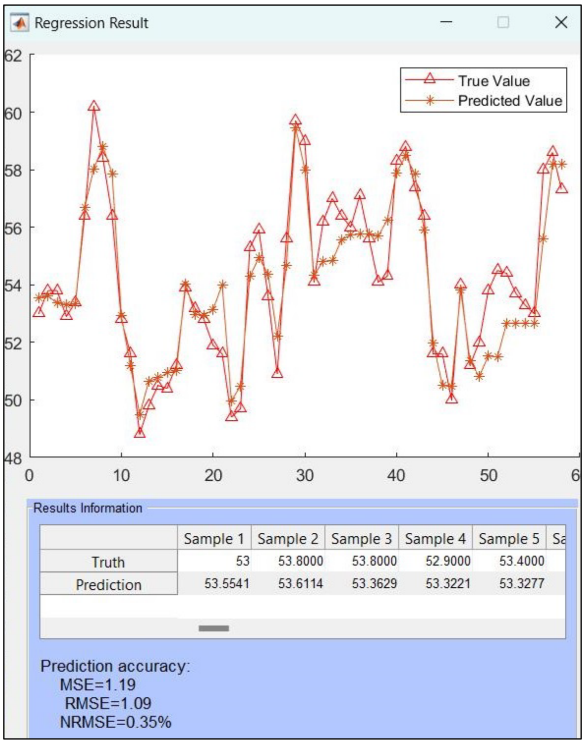


Fig. 5: The spatio-temporal regression test output after unsupervised SNN learning and supervised learning using deSNN as a regressor in the NeuCube architecture, results in an RMSE of 1.09.

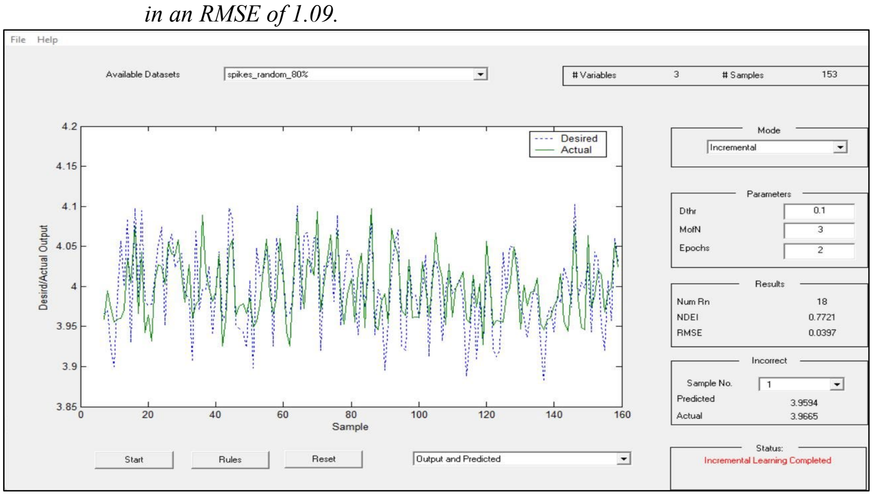


Fig. 6: The regression output of the NeuDen model after SNN learning, feature-vector extraction from deSNN and incremental learning and fuzzy inference in DENFIS, with the lowest error of RMSE 0.0397.

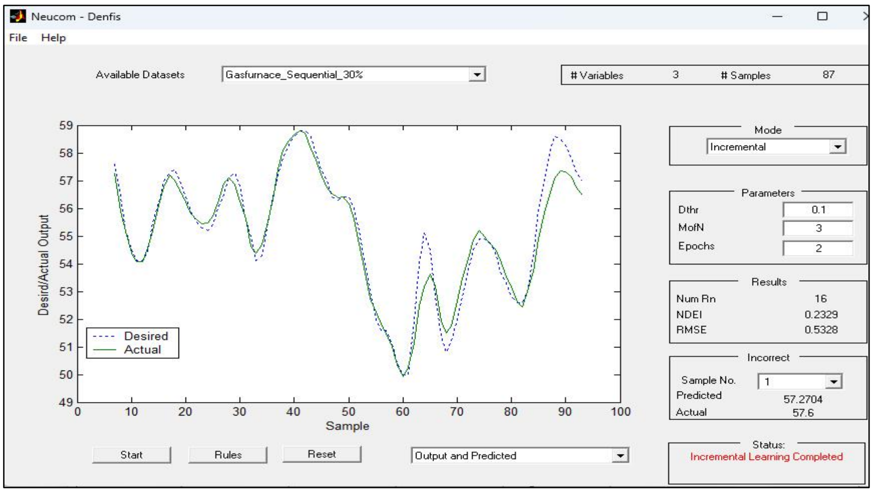
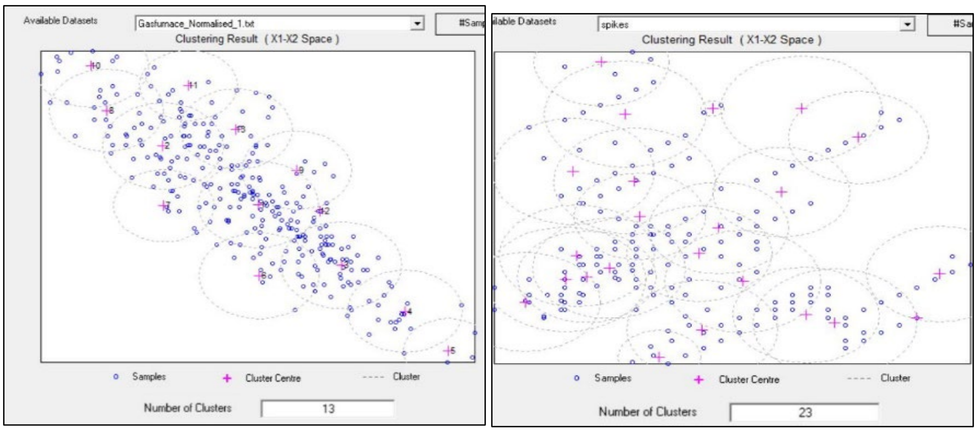


Fig.7. The testing RMSE of the DENFIS model used alone, incrementally trained /tested on the Gas Furnace data. RMSE is 0.5328.

The ability of DENFIS to extract local rules based on evolved clusters is a significant advantage of NeuDen vs NeuCube. Comparing the clusters evolved when DENFIS was used on raw time series data (Fig.8a) vs the clusters evolved in the DENFIS model on the extracted from the SNN feature vectors in the NeuDen model (Fig.8b), it is noticed that the latter clusters are spread wider in the problem space and thus the increased predictive accuracy. The extracted fuzzy rules from the above clusters are also different, incorporating the interaction between the input time series over time in the case of NeuDen clusters.



(a) DENFIS *(b) NeuDen*

Fig.8: (a) Clusters of raw time series data learned in the DENFIS model; (b) clusters of feature-vectors in the NeuDen model.

3. A NEUDEN MODEL FOR MULTIPLE STOCK TIME SERIES PREDICTION

In this section, we aim to further evaluate the NeuDen model on the stock price benchmark dataset, previously utilized for validating the NeuCube model. Using the same data for training and testing as in NeuCube, a NeuDen model results in an RMSE of 0.02, which is 6.5 times smaller than when using NeuCube (SNNcube + deSNN) or SNNcube + SVM (RMSE of 0.13) and 4 times smaller when using DENFIS (RMSE of 0.08) as a single regression model.

3.1. Data Description and pre-processing

In this case study, our experiments focus on time series regression analysis using a share price benchmark demo dataset (HTTP NeuCube). The original dataset comprises 150 ordered time sequences of daily closing prices for six stock indices: Apple Inc. (AAPL), Google (GOOGL), Intel Corp (INTC), Microsoft (MSFT), Yahoo (YHOO), and NASDAQ (QQQX), each time series sample constating of 100 daily prices of the 6 stocks and the objective is to predict the next day NASDAQ stock price.

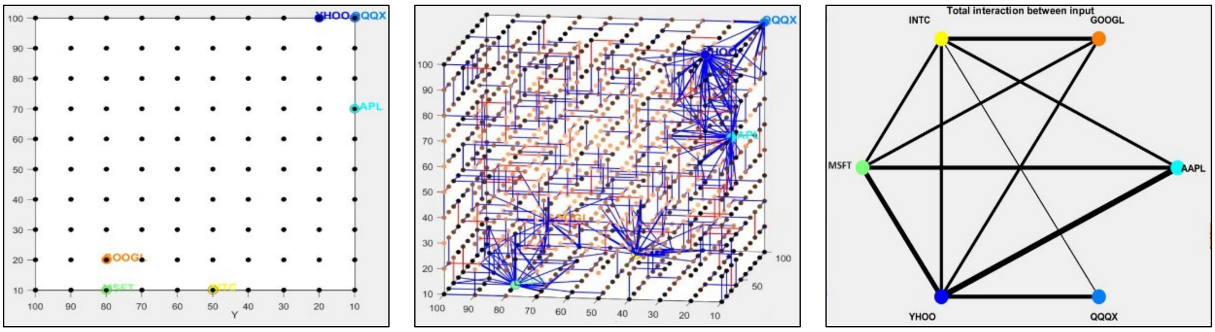
3.2. Design of a NeuDen Model

The NeuDen model processes the generated samples in two distinct stages. Firstly, it undergoes spatiotemporal learning, where the samples are transformed into trains of spikes utilizing a Threshold-Based method with a 0.5 level to govern the generation of corresponding spikes. During this initial stage, the NeuDen model is configured to train on 50% of the generated spike sequences and validated on the following 50%.

Given that financial datasets lack spatial coordination, the NeuDen model autonomously assigns spatial locations for the input neurons in a SNNcube of 1000 spiking neurons using a graph-matching algorithm that allocates input neurons based on how similar their time series are (HTTP NeuCube) – Fig.9a. The SNNcube connections are initialized with small-world connectivity of radius 2.5 to regulate the initial connection weights between neighboring neurons. The unsupervised learning phase is regulated by specific learning parameters, including a Spike-Time Dependent Plasticity (STDP) set at 0.001, a leaking rate of 0.002, a firing threshold of 0.5, a refractory period of 6-time units, and three training iterations.

After training, the SNNcube in the NeuDen model effectively captures the temporal relationships that are present in the original data through its interconnected nodes. The input mapping algorithm reveals a strong correlation between the Nasdaq and the Yahoo index, placing them very close to each other and suggesting significant mutual influence between them. This is further demonstrated in Fig. 9b, which shows denser connections between the Nasdaq and the Yahoo index compared to other features. In this figure only connections with weights above 0.08 are highlighted, with blue connections being positive and red – negative. The dynamics of the daily interaction between the input times series is captured in a feature interaction network graph (Fig. 9c).

In the supervised learning stage, of the deSNN in the NeuDen model, parameters are set to 0.9 and 0.01 for the mod and drift, respectively. After feature-vectors are extracted, a DENFIS model is run on them to perform the prediction and reveal fuzzy rules. The achieved test RMSE on 50% of the data selected for testing is 0.02.



(a) *(b)* *(c)*

Fig. 9: Mapping and neuronal connections in the eSNNcube. (a) Mapping of input features in a 2D view, (b) Synaptic connections following unsupervised training, with blue connections having positive weight and red - having negative weight, (c) Feature Interaction Network with neuron connection weights, defining the level of information exchange between the input variables after spike encoding.

3.3. Comparative analysis of the performance of the NeuDen model vs other regression models

To compare the performance of the NeuDen model, other regression models were applied to the same data and results are shown in Table 1 and in Fig.S1 of Supplementary materials. The NeuDen model results in a RMSE of 0.02, while DENFIS RMSE is 0.08 and other methods resulted after optimization in a RMSE of 0.13.

The NeuDen model allows also for local rule extraction using the new six features. This is based on the evolving clustering method as part of DENFIS (see Fig.S2 from the Supplementary materials). An exemplar rule is given in Table 2.

TABLE 1

THE ACCURACY OF DIFFERENT REGRESSORS MEASURED BY THE RMSE

Model + Regressor	No optimization	After Optimization
SNNcube + deSNN	0.17	0.13
SNNcube + Linear	0.20	0.13
SNNcube + SVM	0.17	0.13
DENFIS	0.08	
NeuDen (SNNcube + DENFIS)	0.02	

TABLE 2

FUZZY RULE SAMPLE EXTRACTED FROM
THE NEUDEN MODEL FOR STOCK VALUE PREDICTION

Rule 1:
if
X1 is GaussianMF (0.50, 0.50) &
X2 is GaussianMF (0.49, 0.72) &
X3 is GaussianMF (0.49, 0.81) &
X4 is GaussianMF (0.44, 0.67) &
X5 is GaussianMF (0.50, 0.20) &
X6 is GaussianMF (0.50, 0.57)
then
Y = 1.78 + 0.45*X - 0.92*X2 - 0.53*X3 + 0.49*X4 + 0.06*X5 + 0.15*X6

4. NUEDEN MODEL DEVELOPMENT AND COMPARATIVE TESTING FOR NOMINAL EFFECTIVE EXCHANGE RATE (NEER) PREDICTION BASED ON MULTIPLE ECONOMIC TIME SERIES

In this section, we develop and validate in a comparative manner a NeuDen Model for predicting Nominal Effective Exchange Rate (NEER). NEER is a measure of a country's currency value relative to a basket of foreign currencies, with weights assigned based on trade importance. It reflects how a country's currency performs against its major trading partners. The predictive test error of the NeuDen model (RMSE of 0.03) is about 100 times smaller than when using a NeuCube model (SNNcube + deSNN) (RMSE of 3.06).

4.1. Data Description

When NEER increases, it signifies an appreciation of the domestic currency, while a decrease indicates depreciation. The NEER formula, as defined in equation (14), sums up the product of exchange rates ($xrate_i$) between the domestic currency and each foreign currency in the basket of top trade partners, multiplied by their respective trade weights ($trdw_i$).

$$NEER = \sum(xrate_i * trdw_i) \tag{14}$$

A NeuDen model is developed in this section to predict NEER data focusing on Lebanon, a country facing an extraordinary crisis. Since 2019, Lebanon has been experiencing an unprecedented sovereign, banking, and currency crisis. The Lebanese 'lira' has lost 98 per cent of its value, and inflation has soared into triple-digits.

Developing a NeuDen model for the prediction of NEER in the context of Lebanon's severe crisis is a challenging test of the model's efficiency, as it seeks to emphasize the model's robustness and ability to provide valuable insights into the dynamics of a vulnerable economy. Until the crisis in 2019, NEER and foreign assets of the Banque du Liban (henceforth ‘BdL’) were generally moving together which is expected as BdL used its reserves to influence the exchange rate (Fig. 10). After that, assets were significantly reduced as a result of subsidizing major imports to meet basic needs at the official exchange rate.

4.2. Data Pre-processing

In this study, we will analyze NEER data in conjunction with several economic aspects relevant to Lebanon. The dataset comprises 152 ordered records of monthly NEER data, spanning from January 2010 to August 2022. They were extracted from the International Financial Statistics (IFS) Dataset Portal (HTTP IFS).

Our analysis involves using a number of independent variables as input features for our NeuDen model. These variables include Gross Domestic Product (GDP), Current Account Balance (CAB), Net Foreign Assets (FA) of Lebanon's Central Bank ‘BdL’, and the Coincident Indicator (COIND) (Table 3).

Annual GDP data is sourced from Lebanon's Central Administration of Statistics (http CAS) and is converted into a constant monthly distribution for modelling purposes. Monthly statistics for CAB, FA, and COIND are obtained from BdL statistics (http BDL). The Current Account Balance (CAB) is a major component of the Balance of Payments, reflecting the balance of trade in goods and services, as well as primary and secondary income balances. In Lebanon's case, CAB has historically been in deficit primarily due to trade deficit. Foreign assets represent the reserves held by BdL, while the Coincident Indicator is a composite measure adopted by BdL, offering insights into economic trends.

TABLE 3.
THE ECONOMIC VARIABLES AS INPUT FEATURES FOR THE PREDICTION OF NEER

Feature	Description
NEER	Nominal Effective Exchange Rate (units)
CAB	Current Account Balance (US$ billions)
FA	Central Bank’s Foreign Assets (US$ billions)
COIND	Coincident Indicator (units)
GDP	Gross Domestic Product (US$ billions)

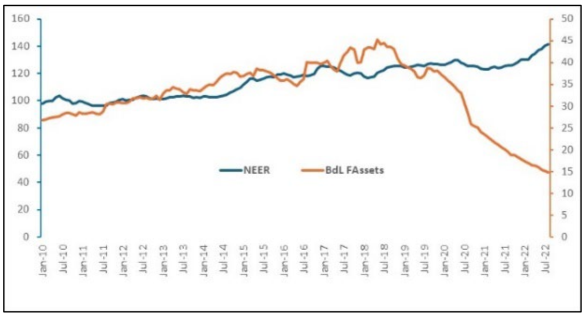


Fig. 10: NEER and BdL Foreign Assets trends.

To enhance the dataset for modelling, we employ a window sampling technique without normalizing the original data. This method generates 140 overlapping samples, each with a window size of 12 months, incorporating both temporal and spatial features, resulting in 60 data points per sample. This sampling method significantly increases the dataset size, facilitating more accurate predictions by capturing continuous temporal fluctuations. In this case study, we aim to predict NEER, considering it as the target feature, based on its historical values, CAB, FA, COIND, GDP, and ε is the residual term constant, as expressed in equation (15).

$$NEER_{t+1} = f\{CAB_t, FA_t, COIND_t, GDP_t, NEER_t, \varepsilon_t\} \tag{15}$$

4.3. Spatio-Temporal Modeling

The generated samples from the input data stream are encoded to the NeuDen model using a Threshold-Based Representation method, where the threshold is set at 0.7, producing a spike frequency of 0.14. These spike sequences are subsequently used to incrementally train and validate the SNNcube, which is divided equally at 50% for training and testing the model. A SNNcube structure is set up with a configuration of 10 neurons across each of its x, y, and z axes. This configuration employs a Leaky-Integrate and Fire neuron model (LIF) to process the incoming spike activity. Each spiking neuron is connected to its neighboring neurons within a maximum distance of a small connectivity radius (SWR = 2.5). The unsupervised learning phase utilizes the Spike-Timing-Dependent Plasticity (STDP) rule with two iterations. The STDP learning rate is set to 0.01, the potential leaking rate to 0.002, the firing threshold to 0.5, and a refractory period of 6 units of time. This stage transfers spikes in time and adjusts connection weights between neurons based on the relative timing of input and output spikes.

Inheriting the faculties of the NeuCube, the proposed model offers visualization of the connection between the neurons after training. A snapshot of the connections above 0.08 weight is shown in Fig. 11. The big dots are the input neurons. The blue lines correspond to positive weight connections and the red lines for the negative.

4.4. Feature-Vector Extraction

The trained SNNcube is recalled on each sample to train in a supervised mode a deSNN regressor. This phase incorporates modulation and drift parameters set at 0.8 and 0.009, respectively. The feature extraction process captures the spike activity of the input neurons. It extracts them in a vector format for each input variable, with a size of the generated new data equal to the number of samples in the input stream (eq. 10). This feature extraction step allows to convert the spatio-temporal information learned by the SNNcube model into a format that can be effectively utilized within neuro-fuzzy learning and inference. In this process, the extracted features differ from those in the original dataset; instead, they represent the spike rates in the input neurons of the SNNcube. The only exception to this extracted spike data is the target feature, which maintains the same values as those in the input data stream. Afterwards, a normalization step is applied to this variable using the natural logarithmic method to ensure its compatibility with the other independent features (eq. 16).

$$\ln(NEER_{t+1}) = f\{F(CAB_t), F(FA_t), F(COIND_t), F(GDP_t), F(NEER_t), \varepsilon'_t\} \tag{16}$$

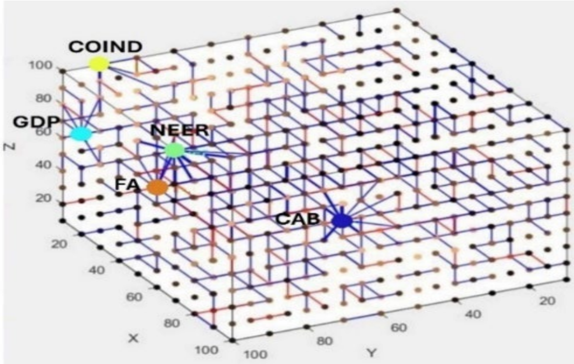


Fig.11. Spatio-temporal connections above 0.08 weight between neurons in a trained eSNNcube as part of NeuDen model for the NEER case study. Big dots indicate input features, blue lines indicate positive connections and red lines – negative connection weights.

4.5. Experimental Results and Analysis

The graphical representation in Fig.13 shows the regression results following spatio-temporal learning in the NeuCube model (SNNcube + deSNN as a regressor), achieving an RMSE of 3.06.

In contrast, Fig.14 indicates the regression output post fuzzy logic training of the spike rates, significantly improving the model's accuracy, achieving RMSE of 0.03.

4.6. Exploring explainability options in the NeuDen model

The correlation matrix, shown in Fig.15, provides a visual and quantitative depiction of the relationships between the new features in the NeuDen model. A moderate positive correlation exists between NEER and BDLFA, suggesting a likely increase in BDLFA when NEER increases. Conversely, a negative correlation between NEER and CAB suggests that an increase in NEER might be associated with a decrease in the CAB. White or light-colored squares show little to no correlation.

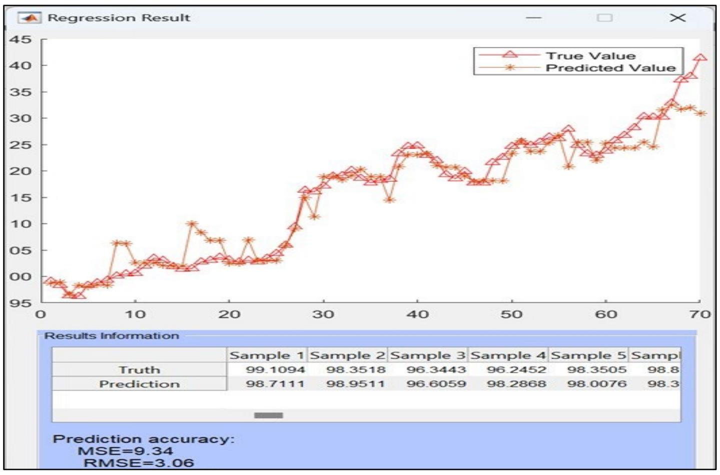


Fig.13. Regression outcomes from spatio-temporal learning and testing phases in NeuCube of the NEER data. RMSE is 3.06.

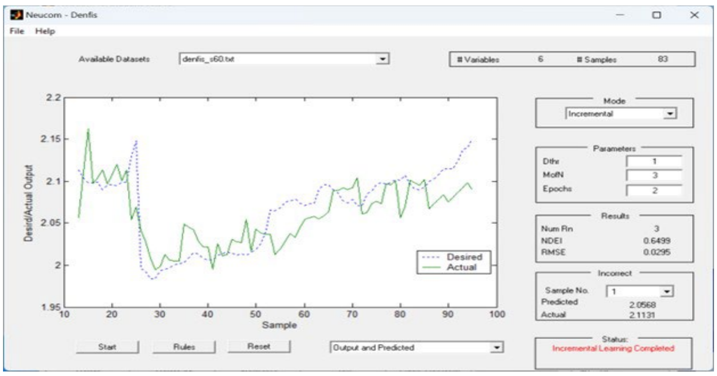


Fig.14. Regression outcomes post fuzzy logic learning and incremental testing phase in the NeuDen model on the NEER data. RMSE is 0.03.

Fig.16 shows the neuron proportion after unsupervised learning in the SNNcube, with the NEER variable as a predominantly active node, accounting for 40% of the overall spike activity. This is followed by Fixed Assets at 36%, and CAB at 16%.

The proposed NeuDen model shows potential in understanding exchange rate dynamics. It considers a combination of economic factors and lagged values to predict NEER at time t. Figure 17 shows the feature Interaction Network after spatio-temporal learning in the SNNcube. The numbers in percentages shown on the graph are the coefficients extracted from the fuzzy regression output and plotted on the graph by the authors for better representation of the model's capacity. The regression output suggests that economic growth and an increase in foreign assets held by monetary authorities tend to strengthen the currency, while larger deficits are associated with currency depreciation.

$$NEER (t) = 1.89 + 0.073333 * GDP - 0.50667 * |CAB| + 0.21 * BDLFA - 0.02 * COIND - 0.18333 * NEER (t-1) \quad (17)$$

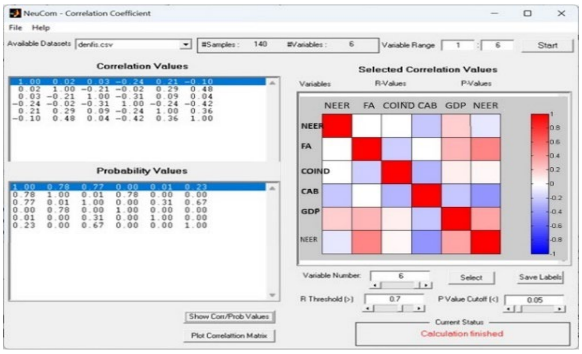


Fig.15. Correlation matrix between economic variables.

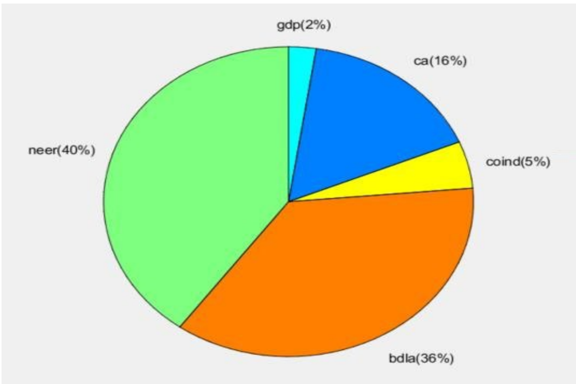


Fig.16. Neuronal activation proportion after unsupervised learning in the eSNNcube as part of a NeuDen model.

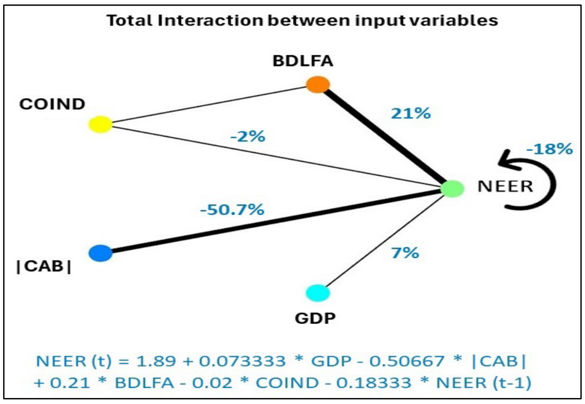


Fig.17. Feature Interaction Network after spatio-temporal learning and fuzzy inference in a NeuDen model. The attached weights to the graph arcs are the coefficients in the derived regression formula (eq. 17).

5. DISCUSSIONS AND FURTHER DEVELOPMENTS

Traditional time series models use predominantly single machine learning techniques that would limit their performance in both accuracy and/or explainability when dealing with complex and multiple time series. The proposed NeuDen model incorporates modeling of spatio-temporal relationship in time series data along with precise fuzzy inference and fuzzy rule extraction.

The NeuDen model distinguishes itself from the well-known models NeuCube and DENFIS primarily through its novel approach to integrating spatiotemporal dynamics and fuzzy logic. Compared to DENFIS, which excels in dynamic environments but lacks in capturing spatiotemporal patterns, the NeuDen model integrates spatiotemporal data processing with a fuzzy logic inference system, providing a significant advantage in capturing and analyzing complex patterns in time series data. Notably, NeuDen's capacity to generate interpretable regression formulas further differentiates it from NeuCube, which lacks this capability. This distinction underscores NeuDen's holistic approach to dealing with the complex nature of time series data, leveraging both spatial and temporal correlations alongside the ability to handle uncertainty through fuzzy logic.

Based on the results of the 3 case studies presented in this paper, we can claim that the integration of eSNN and deNFS is feasible, effective, and promising for multiple time series predictive modelling. The accuracy tests in the 3 case studies result in a much smaller RMSE, of 3 to 100 times, depending on the case study and on the complexity of the data.

Our proposed NeuDen model represents a new paradigm of adaptive and evolving systems. It leverages the capabilities of eSNN models, potentially enhancing the ability to manage complex patterns and dynamics in data, analyze and visualize data connectivity, and establish regression outputs using fuzzy inference methods.

There are many aspects of how the NeuDen approach can be further developed and improved, e.g.:

- Selecting a more specialized spike encoding algorithm (Petro 2019; Nuntalid 2011);
- Directly extracting feature-vectors from the SNNcube, rather than from the deSNN;
- Using other learning rules in the SNNcube;
- Using other classifiers/regressors, such as:
    - fuzzy and neuro-fuzzy systems (Zadeh 1965,1988; Angelov & Filev 2004);
    - self-organizing maps, Adaptive Resonance Theory and other neural network models (Kohonen 1990, Carpenter & Grossberg 1991);
    - explainable deep neural networks (Angelov 2019);
    - quantum classifiers (Ravi 2023).

The NeuDen framework can make a better use of the already created neuromorphic devices (e.g. Delbruck, Indivery) and can be easily implemented in a neuromorphic hardware, such as SpiNNaker (Furber), Loihi (Dey), TrueNorth (Diehl), and many other. The neuromorphic hardware systems offer a scalability (e.g. several millions of spiking neurons), high speed, and low energy consumption.

REFERENCES:

(Abouhassan 2022): Abouhassan I, Kasabov N, Popov G, Trifonov R (2022) “Why Use Evolving Neuro-Fuzzy and Spiking Neural Networks for incremental and explainable learning of time series? A case study on predictive modelling of trade imports and outlier detection,” 1-7, https://doi.org/10.1109 /IS57118.2022.10019673.

(Abouhassan 2023a): Abouhassan I, Kasabov N, Jagtap V, Kulkarni P (2023) “Spiking neural networks for predictive and explainable modelling of multimodal streaming data with a case study on financial time series and online news,” Scientific Reports, https://doi.org/13.10.1038/s41598-023-42605-0.

(Abouhassan 2023b): Abouhassan I, Kasabov N, Bankar T, Garg R, Bhattacharya B (2023) “PAMeT-SNN: Predictive Associative Memory for Multiple Time Series based on Spiking Neural Networks with Case Studies in Economics and Finance,” preprint, https://doi.org/10.36227/techrxiv.24063975

(Amari 1967): Amari S (1967) “A theory of adaptive pattern classifiers.” IEEE Transactions on Electronic Computers, vol. EC-16, no. 3, pp. 299-307, June 1967, https://doi.org/10.1109/PGEC.1967.264666.

(Amari 1990): Amari S (1990) “Mathematical Foundations of Neurocomputing.” Proc. IEEEE, 78, 1143-1163.

(Angelov 2004): Angelov P, Filev D (2004) "An approach to online identification of Takagi-Sugeno fuzzy models," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 34, no. 1, pp. 484-498.

(Angelov 2010): Angelov P, Filev D, Kasabov N (eds) (2010) “Evolving Intelligent Systems,” IEEE Press and Wiley, 2010.

(Angelov 2019): Angelov P, Soares E (2019) “Towards Explainable Deep Neural Networks (xDNN).” ArXiv. /abs/1912. 02523.

(Banihashemi 2017): Banihashemi S, Ding G, Wang J (2017) “Developing a Hybrid Model of Prediction and Classification Algorithms for Building Energy Consumption,” Energy Procedia, vo. 110, 371-376, ISSN 1876-6102, https://doi.org/10.1016/j.egypro.2017. 03.155.

(Bengio 2009): Bengio Y (2009) “Learning Deep Architectures for AI.” Foundations and Trends® in Machine Learning, Vol. 2: No. 1, pp 1-127. http://dx.doi.org/10.1561/2200000006.

(Bezdek 1987): Bezdek J ed. (1987) “Analysis of Fuzzy Information,” vols. 1,2,3, CRC Press, Boca Raton, Florida.

(Bohte 2000): Bohte SM, Kok JN, Poutré HL (2000) “SpikeProp: backpropagation for networks of spiking neurons,” The European Symposium on Artificial Neural Networks, https://api.semanticscholar.org/CorpusID:14069916.

(Bohte 2002): Bohte SM, Kok JN, Poutre HL (2002) “Error-backpropagation in temporally encoded networks of spiking neurons.” Neurocomputing, 48, 17–37.

(Box 1967): Box G, Jenkins M (1976) “Time series analysis: forecasting and control,” Revised, Holden-Day.

(Buonomano & Maass 2009): Buonomano DV, Maass W (2009) “State-dependent computations: spatiotemporal processing in cortical networks.” Nat Rev Neurosci. 2009 Feb;10(2):113-25. https://doi.org/10.1038/nrn2558. Epub 2009 Jan 15. PMID: 19145235.

(Carpenter & Grossberg 1991): Carpenter GA, Grossberg S (1991) “Pattern recognition by self-organizing neural networks,” MIT Press, 1991.

(Chen 2013): Chen Y, Jin HL, Kasabov N, Hou ZL, Cheng L (2013) “NeuroCubeRehab: A Pilot Study for EEG Classification in Rehabilitation Practice Based on Spiking Neural Networks,” International Conference on Neural Information Processing, https://doi.org/10.1007/978-3-642-42051-1_10.

(Diehl 2016): Diehl PU, Pedroni BU, Cassidy A, Merolla P, Neftci E, Zarrella G (2016) "TrueHappiness: Neuromorphic emotion recognition on TrueNorth," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 2016, pp. 4278-4285, doi: 10.1109/IJCNN.2016.7727758.

(Delbruck 2007): Delbruck T, Christian B, Longinotti L (2007) "Real time sensory-motor processing for event-based sensors and systems," http://sourceforge.net/p/jaer/wiki/Home/.

(Dey 2022): Dey S, Dimitrov A (2022) “Mapping and Validating a Point Neuron Model on Intel’s Neuromorphic Hardware Loihi,” Front. Neurosci, 2022.

(Furber 2012): Furber S (2012) “To Build a Brain,” IEEE Spectrum, vol.49, Number 8, 39-41.

(Gerstner 1995): Gerstner W (1995) “Time structure of the activity of neural network models,” Phys. Rev 51

(Gerstner 2002): Gerstner W, Kistler WM (2002) “Spiking Neuron Models: Single neurons, populations, plasticity.” Cambridge University Press. https://doi.org/10.1017/CBO9780511815706.

(Gormley 2016): Gormley M (2016) “Hybrid Graphical Models & Neural Networks”. Probabilistic Graphical Models 10-708, Spring 2016.

(Hebb 1949): Hebb D (1949) “The Organization of Behavior.” New York, John Wiley and Sons.

(Hopfield 1982): Hopfield JJ (1982) “Neural networks and physical systems with emergent collective computational abilities.” PNAS USA, vol.79, 2554-2558.

(Hopfield 1995): Hopfield J (1995) “Pattern recognition computation using action potential timing for stimulus representation.” Nature, 376

(HTTP BDL) : Banque du Liban, https://www.bdl.gov.lb/economicandfinancial datasub,php?docId=192&code=10& filecode=10.

(HTTP CAS): Central Administration of Statistics, http://www.cas.gov.lb/index. php/national-accounts-en.

(HTTP IFS): International Financial Statistics (IFS), https://data.imf.org/regular. aspx? key=61545850.

(HTTP KE): http://www.knowledgeengineering.ai.

(HTTP NeuCom): http//www.theneucom.com

(HTTP NeuCube): http//www.kedri.aut.ac.nz/neucube.

(Hussain 2013): Hussain AJ, Tawfik H (2013) “A Spiking Neural Network for Financial Prediction.” IEEE Proceedings of International Joint Conference on Neural Networks, (pp. 3111-3118), Dallas, Texas, USA.

(Indiveri 2009): Indiveri G, Chicca E, Douglas RJ (2009) “Artificial cognitive systems: From VLSI networks of spiking neurons to neuromorphic cognition. Cognitive Computation, 1(2), 119–127. https://doi.org/10.1007/s12559-008-9003-6.

(Indiveri 2010): Indiveri G, Stefanini F, Chicca E (2010) “Spike-based learning with a generalized integrate and fire silicon neuron.” In: 2010 IEEE International Symposium on Circuits and Systems (ISCAS 2010), Paris, FR, 30 May 2010 - 2 June 2010. Institute of Electrical and Electronics Engineers Corporation (IEEE), 1951-1954.

(Indiveri 2011): Indiveri G, Linares-Barranco B, Hamilton T, Van Schaik A, Etienne-Cummings R, Delbruck T, Liu S, Dudek P, Hafliger P, Renaud S et al. (2011) “Neuromorphic silicon neuron circuits.” Frontiers in Neuroscience, 5.

(Izhikevich 2003): Izhikevich EM (2003) “Simple Model of Spiking Neurons.” IEEE Transactions on Neural Networks, 14(6), 1569-1572.

(Izhikevich 2004): Izhikevich EM (2004) "Which model to use for cortical spiking neurons?" IEEE TNN, 15(5)

(Izhikevich 2006): Izhikevich EM (2006) "Polychronization: computation with spikes." Neural Comput. 2006 Feb;18(2):245-82. https://doi.org/10.1162/089976606775093882. PMID: 16378515.

(Kasabov 1998): Kasabov N (1998) "The ECOS framework and the ECO learning method for evolving connectionist systems," Journal of Advanced Computational Intelligence, 2(6):195–202.

(Kasabov 2002): Kasabov N, Song Q (2002) "DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and Its Application for Time-Series Prediction," IEEE Transactions on Fuzzy Systems, vol. 10, 144-154, https://doi.org/10.1109/ 91.995117.

(Kasabov 2003): Kasabov N (2003) "Evolving Connectionist Systems," Spinger, 2007 (first edition 2003).

(Kasabov 2006): Kasabov N (2006) "Adaptation and interaction in dynamical systems: Modelling and rule discovery through evolving connectionist systems", Applied Soft Computing, 6, 307–322, https://doi.org/0.1016/ j.asoc.2005.01.006

(Kasabov 2010): Kasabov N (2010) "To Spike or Not to Spike: a probabilistic spiking neuron model." Neural Netw. 2010 Jan;23(1):16-9. https://doi.org/10.1016/j.neunet.2009.08.010. Epub 2009 Sep 6. PMID: 19783402.

(Kasabov 2013): Kasabov N, Dhoble K, Nuntalid N, Indiveri G (2013) "Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition," Neural Networks, 41:188-201. https://doi.org/10.1016/j.neunet.2012.11.014.

(Kasabov 2014): Kasabov N (2014): "NeuCube: A Spiking Neural Network Architecture for Mapping, Learning and Understanding of Spatio-Temporal Brain Data," Elsevier, Neural Networks, Vol. 52, pp. 62-76, https://doi.org/10.1016/j.neunet.2014.01.006.

(Kasabov 2019): Kasabov N (2019) "Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence," vol.7, Springer, https://doi.org/10.1007/978-3-662-57715-8.

(Kohonen 1990): Kohonen T (1990) "The Self-Organizing Map," Proceedings of the IEEE, 78, 1464-1480. http://dx.doi.org/10.1109/5.58325.

(Koprinkova 2023): Koprinkova-Hristova P, Penkov D, Nedelcheva S, Yordanov S, Kasabov N (2023) "On-line Learning, Classification and Interpretation of Brain Signals using 3D SNN and ESN," IJCNN, https://doi.org/ 10.1109/IJCNN54540.2023.10191974.

(Kosko 1988): Kosko B (1988) "Bidirectional Associative Memories." IEEE Transactions on Systems, Man, and Cybernetics. 18 (1)

(LeCun 2015): LeCun Y, Bengio YY, Hinton G (2015) "Deep Learning." Nature, 521(7553), 436-444.

(Lee 2016): Lee JH, Delbruck T, Pfeiffer M (2016) "Training Deep Spiking Neural Networks using Backpropagation." Frontiers in Neuroscience, vol. 10, ISSN=1662-453X, https://doi.org/10.3389/fnins.2016.00508.

(Li 2021): Li H, Shen D, Wang L (2021) "A Hybrid Deep Learning Framework for Alzheimer's Disease Classification Based on Multimodal Brain Imaging Data." Frontiers in Neuroscience, 15, 625534.

(Mavaie 2023): Mavaie P, Holder L, Skinner MK (2023) "Hybrid deep learning approach to improve classification of low-volume high-dimensional data". BMC Bioinfor. 24:419, https://doi.org/10.1186/s12859-023-05557-w.

(Neftci 2011): Neftci E, Chicca E, Indiveri G, Douglas R (2011) "A systematic method for configuring VLSI networks of spiking neurons," Neural computation, 23(10):2457-97, https://doi.org/10.1162/NECO_a_00182

(Negi 2023): Negi S, Sharma D, Kosta A K, Roy K (2023) "Best of Both Worlds: Hybrid SNN-ANN Architecture for Event-based Optical Flow Estimation," https://doi.org/10. 48550/arXiv.2306.02960

(Nuntalid 2011): Nuntalid N, Dhoble K, Kasabov N (2011). "EEG Classification with BSA Spike Encoding Algorithm and Evolving Probabilistic Spiking Neural Network." In 18th International Conference on Neural Information Processing. Shanghai, China, Springer, Heidelberg. LNCS 7062, 451-460.

(Petro 2019): Petro B, Kasabov N, Kiss R (2019) "Selection and optimisation of spike encoding methods for spiking neural networks," algorithms, IEEE Transactions of Neural Networks and Learning Systems, April 2019, https://doi.org/10.1109/TNNLS.2019.2906158, vol.31, Issue 2, 358-370.

(Ravi 2023): Ravi, Kasabov N et al (2023). "From Quantum Computing to Quantum-inspired Computation for Neuromorphic Advancement – A Survey." TechRxiv. Preprint, https://doi.org/10.36227/techrxiv.24053250.v1

(Reid 2014a): Reid D, Hussain AJ, Tawfik H (2014) "Financial Time Series Prediction Using Spiking Neural Networks." PLOS ONE, 9, 1-13.

(Reid 2014b): Reid D, Hussain AJ, Tawfik H, Ghazali R (2014) "Prediction of Physical Time Series Using Spiking Neural Networks." Springer 10th International Conference ICIC 2014, LNAI 8589, (pp. 816-824).

(Schliebs 2014): Schliebs S, Kasabov N (2014) "Computational modeling with spiking neural networks," Springer handbook of bio-/neuro informatics, 625-646.

(Schmidhuber 2015): Schmidhuber J (2015) "Deep Learning in Neural Networks: An overview," Neural Networks, vol 61, 85-117, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2014.09.003.

(Song 2000): Song S, Miller K, Abbott L (2000) "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," Nature Neuro- science, vol. 3, 919-926, https://doi.org/10.1038/78829.

(Szatmary 2010): Szatmary B, Izhikevich E (2010) "Spike-Timing Theory of Working memory," PLoS Comput Biol 6 (8)

(Tan 2020): Tan C, Šarlija M, Kasabov N (2020) "Spiking Neural Networks: Background, Recent Development and the NeuCube Architecture," https://doi.org/10.1007/s11063-020-10322-8.

(Takagi 1985): Takagi T, Sugeno M (1985) "Fuzzy identification of systems and its applications to modeling and control," IEEE transactions on systems, man and cybernetics, (1), 116-132.

(Thorpe 2001): Thorpe S, Delorme A, et al. (2001) "Spike-based strategies for rapid processing." Neural Netw.,14(6-7),715-25.

(Verstraeten 2007): Verstraeten D, Schrauwen B, D'Haene M, Stroobandt D (2007) "An experimental unification of reservoir computing methods," Neural Networks, 20(3):391-403. https://doi.org/10.1016/j.neunet.2007.04.003. Epub 2007 Apr 29. PMID: 17517492.

(Watts 2009): Watts MJ (2009) "A Decade of Kasabov's Evolving Connectionist Systems: A Review," IEEE transactions on systems, man and cybernetics, part C, applications and reviews, vol. 39, issue 3.

(Wu 2018): Wu Y, Deng L, Li G, Zhu J, Shi L (2018) "Spatio-temporal Backpropagation for Training High-Performance Spiking Neural Networks." Frontiers in Neuroscience, 12, 737-744, https://doi.org/10.3389/fnins.2018.00331.

(Yager 1994): Yager RR, Filev D (1994) "Generation of fuzzy rules by mountain clustering," J. of Intell. and Fuzzy Syst., 2, 209-19, 1994.

(Yamakawa 1992): Yamakawa T, Uchino E, Miki T, Kusanagi H (1992) "A Neo Fuzzy Neuron and Its Application to System Identification and Prediction of the System behaviour." Proceedings of the 2nd International Conference on Fuzzy Logic & Neural Networks, Iizuka, Japan, 477-483.

(Yan 2013): Yan H, Zhao L, Hu L, Wang X, Wang E, Wang J (2013) "Nonequilibrium landscape theory of neural networks." Proceedings of the National Academy of Sciences of the United States of America, 110(45), 18035-18035.

(Yang 2011): Yang L, Zhongjian T (2011) "Prediction of Grain Yield Based on Spiking Neural Networks Model." IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011, (pp. 171-174).

(Zadeh 1965): Zadeh L (1965) "Fuzzy Sets." Information and Control, vol.8, 338-353.

(Zadeh 1988): Zadeh LA (1988) "Fuzzy Logic." IEEE Computer 21.