Integrated Feature, Neighbourhood, and Model Optimization for Personalised Modelling and Knowledge Discovery

Wen Liang (Linda)

A thesis submitted to Auckland University of Technology in fulfillment of the requirements for the degree of Master of Computer and Information Science (MCIS)

2009

School of Computer and Information Science

Primary Supervisor: Professor Nikola Kasabov

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AUT:	Auckland University of Technology
AI:	Artificial Intelligence
ECF:	Evolving Classification Function
ECMC:	Evolving Clustering Method for Classification
ECOS:	Evolving Connectionist Systems
GA:	Genetic Algorithm
GAPM:	GA-Personalised Modelling
GAGSc:	GA gene selection method in terms of consistency
KEDRI:	Knowledge Engineering and Discovery Research Institute
KNN:	K-Nearest Neighbour
LS-SVM:	Least Square Support Vector Machine
LOOCV:	Leave-One-Out Cross-Validation
MLP:	Multi-Layer Perception
MLR:	Multiple Linear Regression
NSVM:	Newton Support Vector Machine
RBF:	Radial Basis Function
SVM:	Support Vector Machine
SLT:	Statistical Learning Theory
SNR:	Signal-to-Noise Ratio
SSVM:	Smooth Support Vector Machine
TWNFI:	Transductive Neural Fuzzy Inference System
WKNN:	Weighted K-Nearest Neighbour
WWKNN:	Weighted-Weighted K-Nearest Neighbour

Attestation of Authorship

"I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning".

Wen Liang (Linda)
April, 2009

Acknowledgements

I would like to extend my sincere thanks and gratitude to the following individuals who have contributed towards the completion of my master's thesis:

Huge thanks to my primary supervisor, Professor Nikola Kasabov, for taking me on board as a Master's student. Your support and comments were very important to my study and ensured the research advanced in the right direction. If it wasn't for your great ideas, your patience, your tolerance and your encouragement over the last year, I do not know where I would be today. Thank you so much!!

I also would like to extend my gratitude to the awesome students and staff members at KEDRI, who provided their time and energy to assist me for in solving various problems during my study.

Special thanks to my parents in China and parents-in-law in Taiwan. Thank you for all your great ideas and support in many practical ways during my study. Thanks also to my husband Chun and my lovely daughter Jasmine, who put up with my bad moods when I was stressed and tired.

I would like to acknowledge the generous support of Auckland University of Technology for providing me with a good study environment and condition, especially thanks to Mrs. Catriona Carruthers for helping me to proofread my paper.

*** Restrictions for using this material:**

As part of the thesis relates to partial implementation of a provisional patent: Data Analysis and Predictive Systems and Related Methodologies, Nikola Kasabov US provisional patent application No. 61/10,5,742; this implementation cannot be used for commercial applications before permission is granted.

Abstract

"Machine learning is the process of discovering and interpreting meaningful information, such as new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques" (Larose, 2005). From my understanding, machine learning is a process of using different analysis techniques to observe previously unknown, potentially meaningful information, and discover strong patterns and relationships from a large dataset. Professor Kasabov (2007b) classified computational models into three categories (e.g. global, local, and personalised) which have been widespread and used in the areas of data analysis and decision support in general, and in the areas of medicine and bioinformatics in particular. Most recently, the concept of personalised modelling has been widely applied to various disciplines such as personalised medicine, personalised drug design for known diseases (e.g. cancer, diabetes, brain disease, etc.) as well as for other modelling problems in ecology, business, finance, crime prevention, and so on. The philosophy behind the personalised modelling approach is that every person is different from others, thus he/she will benefit from having a personalised model and treatment. However, personalised modelling is not without issues, such as defining the correct number of neighbours or defining an appropriate number of features. As a result, the principal goal of this research is to study and address these issues and to create a novel framework and system for personalised modelling. The framework would allow users to select and optimise the most important features and nearest neighbours for a new input sample in relation to a certain problem based on a weighted variable distance measure in order to obtain more precise prognostic accuracy and personalised knowledge, when compared with global modelling and local modelling approaches.

Chapter 1

Introduction

1.1. Motivation

Professor Kasabov (2007b) classified computational models into three categories: *global*, *local*, and "*personalised*". The basic philosophy behind *personalised modelling* is every person is different from others, thus he/she needs and deserves a personalised model and treatment that best predicts a possible outcome for this person. A personalised model is created for every single new input vector of the problem space based on its nearest neighbours. In contrast, *global modelling* refers to a model that is created for the whole problem space rather than focusing on individual cases, thus this model has difficulty undergoing adaptation due to new input vectors. *Local modelling* refers to a model that is created to calculate the output function that is used to deal with a sub-space of the entire problem space. This approach provides a better explanation for individual vectors, and any further new input vectors are much more easily studied as well. These three modelling approaches have been successfully applied to deal with a variety of classification and prediction problem tasks, such as handwriting recognition (e.g. segment the text into individual characters and classify them), face detection (e.g. give an image to classify as face or not face), as well as weather prediction (e.g. predict the weather for the next several days) and climate prediction (e.g. temperature and soil moisture).

"*Personalised modelling*" is an emerging approach which has been applied for numerous decades to evaluate and deal with a variety of modelling problems. For instance, in the field of *personalised healthcare,* the knowledge discovered by this approach has significantly contributed to prediction, diagnosis and therapy for individual patients' diseases. This approach has also resulted in improved patient safety (Iakovidis, 2007; Baek et al., n.d.). In the articles by Ginsburg and McCarthy (2001) and TEMU (2008), it has been mentioned that providing a personalised therapy for an individual patient during the diagnosis timeframe has proved to be very efficient and helpful. Furthermore, given the current advances in networking technologies, *personalised mobile service* provides a more efficient service, which in turn also benefits business (Lankhorst, Kranenburg, Salden, & Peddemors, 2002). Most recently, *personalised web* is an emerging technique that provides users with personalised search

and browsing systems (McGowan, Kushmerick & Smyth, 2002; Magoulas & Dimakopoulos, 2005).

Nowadays, the concept of *personalised medicine* is becoming a leading trend in medicine, health care and life science. As presented by Lesko (2007) who is from the U.S. Food and Drug Administration, "Personalized medicine can be viewed…as a comprehensive, prospective approach to preventing, diagnosing, treating, and monitoring disease in ways that achieve optimal individual health-care decisions." Personalised medicine brings many benefits and possibilities to different disciplines. For instance, for patients and clinicians, they receive more effective, precise and safer diagnosis and treatment; for the pharmaceutical industry, the benefits are efficiently improving productivity and the efficiency of product lines; and for society as a whole, the benefits are receiving more focused applications of valuable health care resources. As stated by the Personalized Medicine Coalition (2008), traditional medicine is primarily based on the visible symptoms of the disease, but recently doctors can integrate an individual patient's molecular profile to characterise various forms of cancers (e.g. breast cancer, brain cancer, and liver cancer, etc) to make a decision about treatment. Furthermore, according to Ginsburg and McCarthy (2001), the objective of personalised medicine is to determine a patient's disease at the molecular level, so the right therapies are able to be applied on the right people at the right time. Multiple examples have significantly proved that the traditional form of medicine is declining in favour of more accurate marker-assisted diagnosis and treatment. In contrast, personalised medicine is escalating, being primarily based on an individual patient's molecular profile. The concept of personalised modelling is worth further investigation as it has a vast potential.

During my Master's study at Auckland University of Technology (AUT), I had an opportunity to talk with Professor Kasabov who is the Director and Chief Scientist of KEDRI. As a result, I started to take a further look at the area of "*Personalised Modelling*".

1.2. Research Scope and Focus

A good research study should be educational, informative, meaningful and useful. To this end, it is essential to specify the research scope and focus. *Global modelling*, *local modelling*, and *personalised modelling* are the three important categories of learning

models that can be utilized in the area of data analysis and decision support in general, and in the area of medicine and bioinformatics in particular (Kasabov, 2007b). In this research, I particularly concentrated on personalised modelling, especially focusing on developing a novel framework and system for personalised modelling by integrating it with the Genetic Algorithm that would include: selecting and ranking the most important features and nearest neighbours of a new input sample in relation to a certain problem based on a weighted variable distance measure. The main two reasons for this are: (1) To provide more accurate and effective accuracy when compared with the global modelling and local modelling approaches, and (2) To provide more precise personalised knowledge and a better understanding of meaningful information. The research scope and focus are depicted in Figure 1.1.

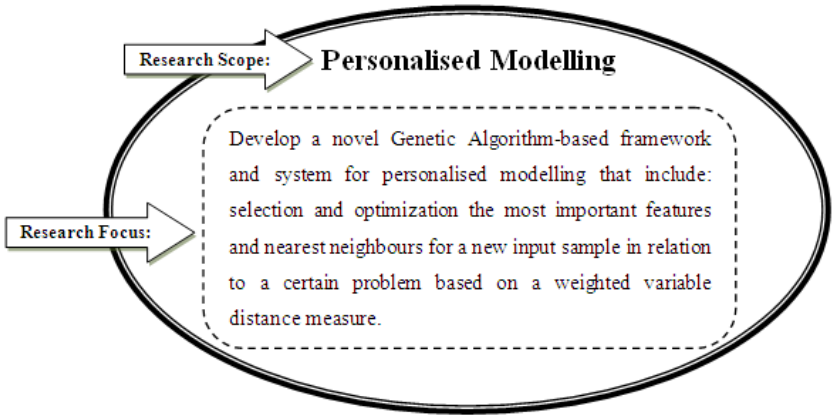


Fig.1.1: Research scope and focus.

1.3. Research Objective

The major objective of this research is to develop a novel personalised modelling framework and system to select and optimise the most significant features and the optimal number of nearest neighbours for a single input sample, corresponding to a certain problem, based on a weighted variable distance measure. The novel framework and system might provide more accurate performance and more precise personalised knowledge when compared with the global modelling and local modelling approaches.

Additionally, a list of opening questions which need to be addressed and studied in this study are as follows:

Q 1: Can a GA-based system select optimal nearest neighbours for every new input vector?

The major reason to define an appropriate number of nearest neighbours is to help researchers significantly improve classification or prediction accuracy. It remains

a challenging opening question that needs to be considered and addressed.

Q2: What features are significant for every new input vector?

Feature selection is defined as a simple pre-processing technique for choosing the most significant features when creating models. The reasons for addressing the problem of selecting an optimal number of features are: (1) In this way we can reduce the number of features and only concentrate on the most important ones, thus the possible noise in the data set is significantly reduced and better accuracy can be achieved; (2) It allows the building of a model that generalizes better to unseen points, and (3) It avoids over-fitting and improves the model's performance.

1.4. Thesis Contribution to Information Science

Genetic algorithm (GA) is defined as an optimization technique to solve complex optimization problems, which are primarily based on the principle of Darwin's "survival of the fittest". GA is able to deal with a large problem space efficiently, as well as to achieve an optimal or close to optimal solution after a number of iterative computations. As a result, based on the traditional personalised modelling algorithms, GA is adopted as a method of optimizing the following parameters in order to deal with the questions proposed above:

- ✧ Selection of an optimal number of nearest neighbours for every new input vector.
- ✧ Selection of an optimal set of features that best contribute to the classification and prediction tasks.

Thus, a novel GA-based personalised modelling (GAPM) system might provide better classification and prediction results when compared with global and local modelling approaches. It also might provide more precise personalised knowledge and a better understanding of meaningful information.

Moreover, the novel GAPM system is applied for knowledge discovery on a real-world pest-related climate data set. This data set contains information on pest establishment in numerous regions of the world. This data set was successfully presented in 2004 as a technical report to the National Centre of Research Excellence in Bioprotection, which is operated by Lincoln University in New Zealand.

This paper was presented as a poster at the 15th International Conference on Neuro-Information Processing of the Asia Pacific Neural Network Assembly in 2008. This paper will also be presented at the 16th International Conference on Neuro-Information Processing of the Asia Pacific Neural Network Assembly in 2009.

1.5. Thesis Content

The entire study is organized into the following chapters:

Chapter 2 is a literature review of methods of personalised modelling, which includes a comparison between inductive modelling and transductive modelling approaches, how they work, and their applications. In addition, the two opening questions proposed above (section 1.3) are considered in light of implementation of the transductive inference approach. Both inductive inference and transductive inference methods are studied further, including a detailed literature review on global, local and personalised modelling approaches.

Chapter 3 reviews a number of techniques involved in GAPM, including an overview of feature selection methods, cross-validation techniques, and optimization methods.

Chapter 4 presents a novel GA-based framework and system for personalised modelling based on transductive modelling in order to study and address the opening questions raised in chapter 1. This chapter clearly explains: (1) the motivation behind developing this novel framework and system; (2) the workings of this novel system with WKNN and WWKNN as base models, and (3) the knowledge discovery arising from the novel GAPM system.

Chapter 5 firstly presents a graphical user interface (GUI) that demonstrates how the novel GAPM system runs using MATLAB. Secondly, an experiment run on a benchmark data set (e.g. Sonar) is performed using NeuCom and the novel GA-based personalised modelling system to compare the classification accuracy of different algorithms. The results with their detailed analysis are described in the last section.

Chapter 6 offers a detailed comparative analysis of global and local modelling approaches against the personalised modelling approach of GAPM on the leukaemia cancer data set. This chapter contains the following sections: (1) A problem specification section that introduces the basic concepts of leukaemia cancer and the

reasons for studying a data set related to this area; (2) A data set section that gives a description of the data set and the data pre-processing stages; (3) An experimental setup section that introduces the two pieces of software used in this study, each step of the experiments as well as the methodology of each step, and (4) A section that presents the experimental results with detailed analysis.

Chapter 7 presents a detailed comparative analysis of global and local modelling approaches against the personalised modelling approach using a real world pest-related climate data set. This chapter has the same structure as Chapter 6.

Finally, Chapter 8 presents the conclusions of this study as well as suggestions for future work.

Chapter 2

Methods for Personalised Modelling: A Literature Review

2.1. Introduction

Before a detailed literature review on methods for personalised modelling is presented, firstly a comparison between inductive modelling and transductive modelling approaches, including the theory behind these two approaches, as well as their applications are introduced. Secondly, both inductive inference and transductive inference methods are studied further, including a detailed literature review on global, local and personalised modelling approaches.

2.2. Inductive versus Transductive Reasoning Approaches

Up until now, most learning models in the area of artificial intelligence (AI) are developed and implemented, especially those employing neural fuzzy inference methods, based on either inductive inference or transductive reasoning approaches. Figure 2.1 graphically presents the differences between these two reasoning approaches. It can be seen that the transductive inference method is associated with both training and testing data in a problem space, while the inductive inference method has to induce a function from the training data first and then deduct the function and use it to predict the testing data (Vapnik, 2005). Further comparisons between these two reasoning approaches are studied in the following section.

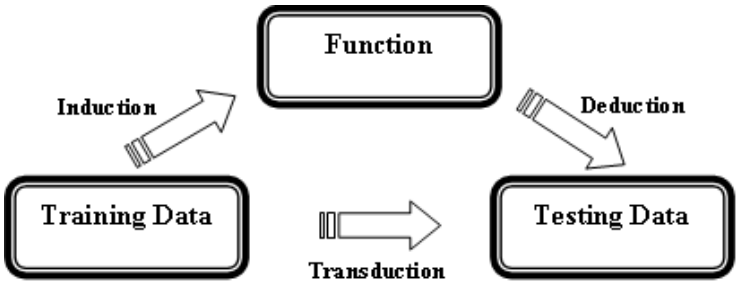


Fig.2.1: The overall differences between inductive inference and transductive inference methods.

2.2.1. Inductive Inference Method

The theory of inductive inference was originally introduced by Ray Solomonoff around 1960. It is defined as a process of inferring a general rule or law from the observations of a particular example (Angluin & Smith, 1983). For instance, for a given binary string "*100, 111100, 11000, 1110, 1100*", the following rule can be inferred; "*any number of 1s followed by any number of 0s*". In general, the

inductive inference method is concerned with the creation of a model (generally a global model), where the model is created from the all available data. It focuses on the whole problem space. This model can be further adapted to investigate new input vectors. Once a global model is created, no new information about a new input vector is taken into account, and the error is calculated to measure how well the new input vector can fit into the model. Figure 2.2 presents an overview of an inductive inference method: where a global model M is created built on the data set D, and the model is then recalled for every new input vector V_i. Finally, the output (Y_i) of each new input vector is calculated using the model.

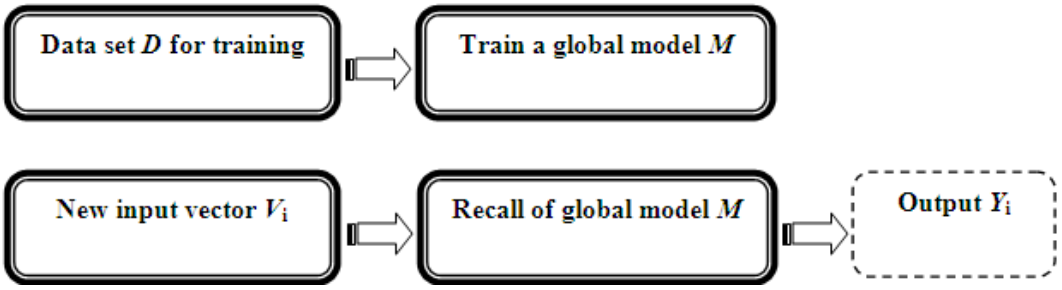


Fig.2.2: Overview of an inductive inference method.

The inductive inference approach has already been widely applied to develop abstract models of the process by which a child acquires its native language, or the process of scientific inquiries (Gold, 1967; Putnam, 1975; Wexler & Culicover, 1980). Most recently, a variety of proposals for the inductive inference method have been successfully applied to practical systems. One such inductive inference method based system is the "Lindenmayer system **(**L-system)" that utilizes tools from formal language theory to represent changes in biological organisms over a given time period (Doucet, 1974; Feliciangeli & Herman, 1977). The inductive inference method is also potentially useful in automatic program synthesis applications and helpful in specifying programming languages. For instance, Biermann and Krishnaswamy (1976) developed a synthesis system that uses trace information that is initially provided by users. Shaw, Swarvout and Green (1975) also proposed an interactive system for synthesizing the LISP programming language. The inductive inference method also has applications in the field of pattern recognition where the method helps in text categorization and recognition. It determines whether a given input pattern belongs to the specified class exactly as according to the given grammar structure.

From the above, it can be seen that the inductive inference approach has been successfully applied to a variety of disciplines. However, one of the major drawbacks of this approach is that it is only concerned with the evaluation of a model based on data from the entire problem space and this can be a difficult task and is not really necessary in most cases.

2.2.2. Transductive Inference Method

The transductive inference approach was originally proposed by Vapnik in 1998. It is defined as a method that evaluates the potential value of a model for only an individual point of the problem space by using additional information related to that single point. In contrast to the inductive inference approach, the transductive inference approach is more concerned with solving an individual given problem rather than solving a general problem (Bosnic et al., 2003). Figures 2.3 and 2.4 present an overview of a transductive inference approach: every new input vector V_i requires investigation for a classification or prediction task which is primarily based on its nearest neighbours. Thus, the nearest neighbours form a sub-data set D_i that is derived from the original training data set D. A new local model M_i is dynamically created based on these vectors and further adapted to estimate the output Y_i for every new input vector V_i.

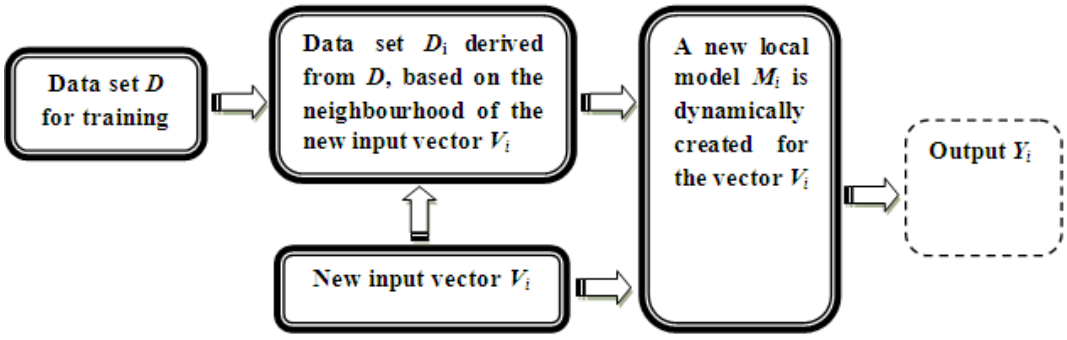


Fig.2.3: Overview of a transductive inference method (a) (modified from Song & Kasabov, 2005).

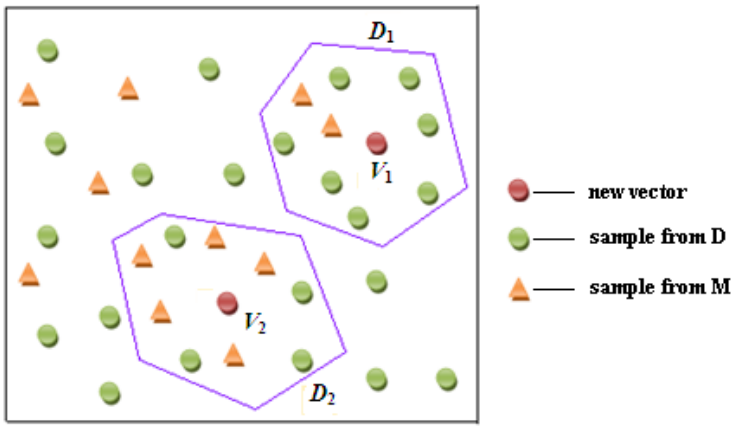


Fig.2.4: Overview of a transductive inference method (b):V_1 and V_2 represent the two new input vectors which are surrounded by a number of nearest neighbours selected from the training data set *D* and generated from an existing model *M* (modified from Song & Kasabov, 2005).

The transductive inference approach has been widely applied to applications very much related to clinical or/and medical fields due its focus being primarily individual patients. For instance, this method has been successfully used in the area of personalised clinical care, where each patient's medication is based on their personal information and medical condition (Spang, 2003; Williams, 2003; Kriete, 2004; Angrist, 2005). Furthermore, this approach has been widely applied in the area of medical disease prediction (Nevins et al., 2003; Pittman et al., 2004; Tyrer, Duffy & Cuzick, 2004). In the article by Weston et al. (2003), the transductive inference approach is also thought to be potentially useful in dealing with a variety of prediction tasks, such as predicting whether a given drug will bind to a target site, as well as providing additional measures to discover the reliability of predictions made in medical diagnosis (Kukar, 2003). This method can also be used for solving a variety of classification tasks, such as image classification (Proedrou et al., 2002), text classification (Joachims, 1999; Chen, Wang & Dong, 2003), heart disease diagnostics (Wu et al., 1999), digit and speech recognition (Joachims, 2003), and micro-array gene expression classification (Wolf & Mukherjee, 2004).

Most recently, the transductive inference method has been successfully applied in the field of bioinformatics using support vector machine (SVM) and the experimental results prove that it provides better accuracy than the inductive inference method (Kasabov & Pang, 2004). The main reason for this is the transductive inference method exploits the structural information of unlabeled

data. However, one of the drawbacks of the transductive inference approach is that it is only efficient when the size of the data set is small. In addition to this, there is a list of opening questions which are raised when implementing the transductive modelling approach, including: "*How many nearest neighbours should be selected for every new input vector?*" and "*What features are significant for every new input vector?*". These two major questions will be studied and addressed in this research.

2.3. Global, Local and Personalised Modelling

2.3.1. Global Modelling

In global modelling, a global model is created from the entire data set for the whole problem space rather than focusing on individual vectors. This model is usually very difficult to be adapted on new incoming input vectors. Examples of global modelling are: *Support Vector Machine* (SVM), *Multiple Linear Regression* (MLR), and *Multi-Layer Perception* (MLP).

In this study, SVM is presented as a popular algorithm for comparison with local and personalised modelling algorithms. One main reason is that it is a fast optimization algorithm that can obtain high-quality classification accuracy with few training samples. However, in dealing with a large, high-dimensional data set, the kernel computation time for training the SVM classifier is long.

2.3.1.1. SVM

SVM is a supervised learning algorithm based on small-sample *Statistical Learning Theory*, which was originally proposed by Vapnik (1998) and his co-workers. It has been widely applied to deal with classification and regression problems. In addition, it has been successively extended by several other researchers, such as *V-SVM* (Schölkopf & Snola, 2000), *Smooth Support Vector Machine* (SSVM) (Lee & Mangasarian, 2001), *Newton Support Vector Machine* (NSVM) (Fung & Mangasarian, 2004), and *Least Square Support Vector Machine* (LS-SVM) (Suykens & Vandewalle, 1999).

SVM is a powerful tool for separating a set of binary labeled data in a feature space using an optimal hyperplane. The two major types of SVM

used far and wide, are *linear SVM* (Vapnik & Lerner, 1963) and *non-linear SVM* (Aizerman & Braverman, 1964). In cases where the data is linearly separable, SVM separates a given set of training data with a hyperplane, thus the distance from the hyperplane to the data is maximized (also known as "the maximum margin hyperplane"). If the data is non-linearly separable, SVM can work in combination with the non-linear "kernel function" that can automatically map the data onto a feature space (possibly a high-dimensional feature space). As a result, the hyperplane in the high-dimensional feature space corresponds to a non-linear decision boundary in the original input space. Figure 2.5 presents an overview of the SVM process: exploring an optimal hyperplane to split a set of vectors in such a way that vectors within one category are placed on one side of the plane, while vectors within other category are placed on the other side of the plane. As stated by Noble (2006), the SVM algorithm has been adopted increasingly in a wide variety of applications such as the automatic classification of micro-array gene expression profiles, as well as identifying handwritten digits through studying a large group of scanned images of handwritten zeroes, ones, etc.

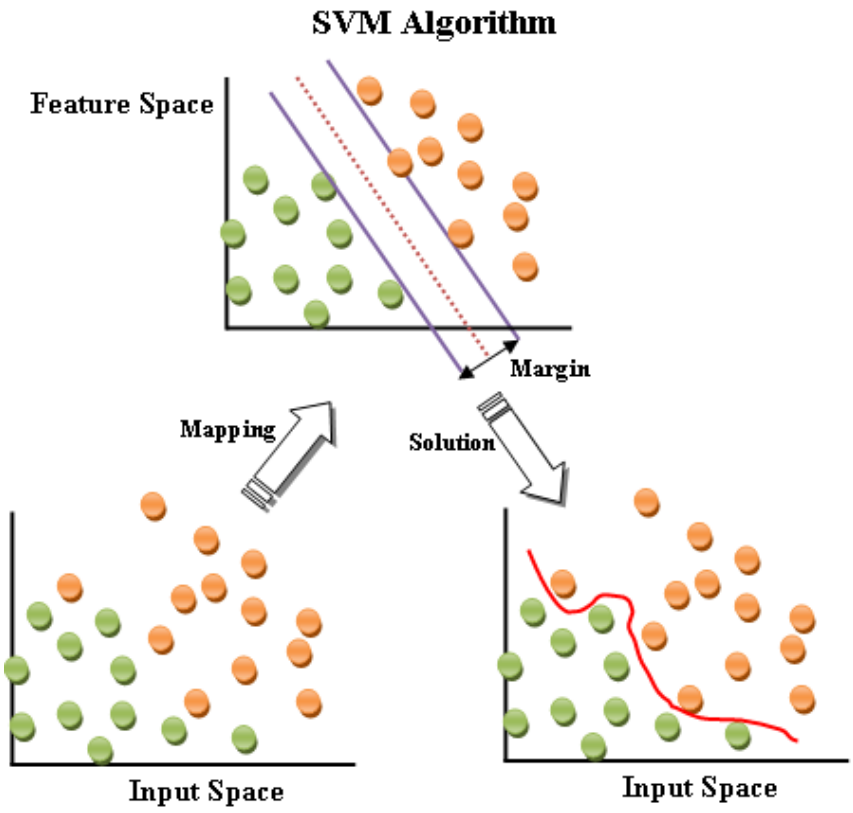


Fig.2.5: Overview of a simple SVM process.

Mathematically, the SVM can be formulated as the following equation (Gunn, 1998): suppose we have a two-class classification task

$$D = \{((x_{1,} y_1),...(x_{i,} y_i)) \mid x \in R^n, y \in \{-1,1\}\}_{i=1}^{m} \quad (2.1)$$

where D is the given training data set, x is the n-dimensional vector, and y is the class label which indicates which class the x belongs to. Figure 2.6 illustrates that when the data are linearly separable, the optimal hyperplane is defined as:

$$w^T_{(w_1,...w_n)} * x + b = 0 \quad (2.2)$$

where w is the weight vector, and b is the scalar. Therefore, the optimal hyperplane separates those vectors belonging to two different classes. Furthermore, both w and b can be constrained such that:

$$W(\Lambda) = \min L(w, b, \Lambda) \quad (2.3)$$

where L is the Lagrange function, and Λ is the Lagrange multiplier. If we want to choose the w and b to maximize the margin, the hyperplane in Equation 2.2 can be re-defined as:

$$w^T_{(w_1,...w_n)} * x + b = 1 \quad (2.4)$$

$$w^T_{(w_1,...w_n)} * x + b = -1 \quad (2.5)$$

It can be seen that if the distance between the vectors belonging to the two different classes is maximized, those vectors are optimally separated by the hyperplane in Equations 2.4 and 2.5. With Equation 2.3, the parameters w, Λ and the optimal hyperplane are given as:

$$w = \sum_{i=1}^{n} \Lambda_i x_i y_i \quad (2.6)$$

Therefore, the classifying function can be defined as:

$$f(x) = w^T_{(w_1,...w_n)} * x + b \quad (2.7)$$

Finally, the result (either 1 or -1) calculated in Equation 2.7 can be further adopted to determine the class which x belongs to.

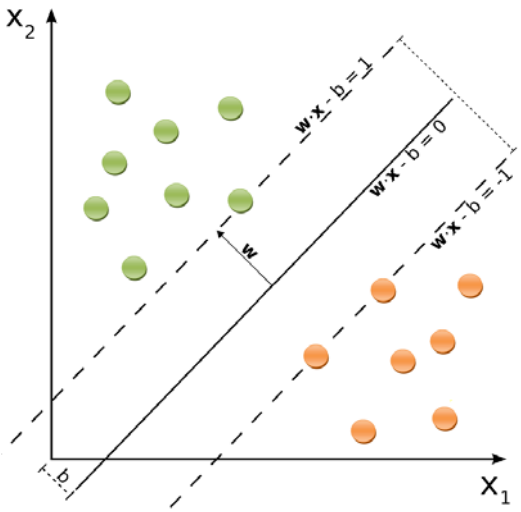


Fig.2.6: Overview of a simple linearly separable SVM.

2.3.2. Local Modelling

A local model is created to evaluate an output function that is able to deal with a sub-space of a problem space. In other words, the local modelling approach has the ability to provide a better explanation and knowledge about individual vectors than global modelling. Moreover, the subsequent new input vectors are much easier to be investigated using this model. Local modelling includes *Radial Basis Function* (RBF), *Evolving Classification Function* (ECF), and *Evolving Clustering Method for Classification* (ECMC).

In this study, ECF is presented as a popular algorithm for comparison with global and personalised modelling algorithms. The main reason for this choice is that it has two special characteristics: (1) It allows fast incremental and online learning, and (2) The dynamic allocation of rule nodes helps users to easily understand and even verify the model's functionality (Kasabov, 2007).

2.3.2.1. ECF

As stated by Arbib (2003), traditional neural network models do not allow researchers to discover new patterns from the data as they are seen as "black boxes". As a result, Kasabov introduced a novel type of neural network model in 2003 called evolving connectionist systems (ECOS) that allows for fast incremental, online learning, as well as rule extraction and rule adaptation. According to Kasabov (2007a), "*Evolving connectionist system* (ECOS) is a connectionist architecture that facilitates modelling of an

evolving process and knowledge discovery", which represents a new piece of "neural network" knowledge. The concept of evolving classifier function (ECF) is a typical implementation of ECOS which has been widely applied to pattern classification tasks. Theoretically speaking, ECF is composed of four layers of nodes: (1) input variables, (2) fuzzy membership functions, (3) a set centers of data in the input space, and (4) classes (Kasabov, 2007a). ECF can produce rule nodes in a multi-dimensional input space and each rule node is identified by its *radius*, *center* as well as the *class* it belongs to. Figure 2.7 demonstrates an example of the classification task of clusters of data: where *c* is the class, v_i is the *i*-th data vector, o_j is the centre of *j*-th node, and r_j is the radius of *j*-th node.

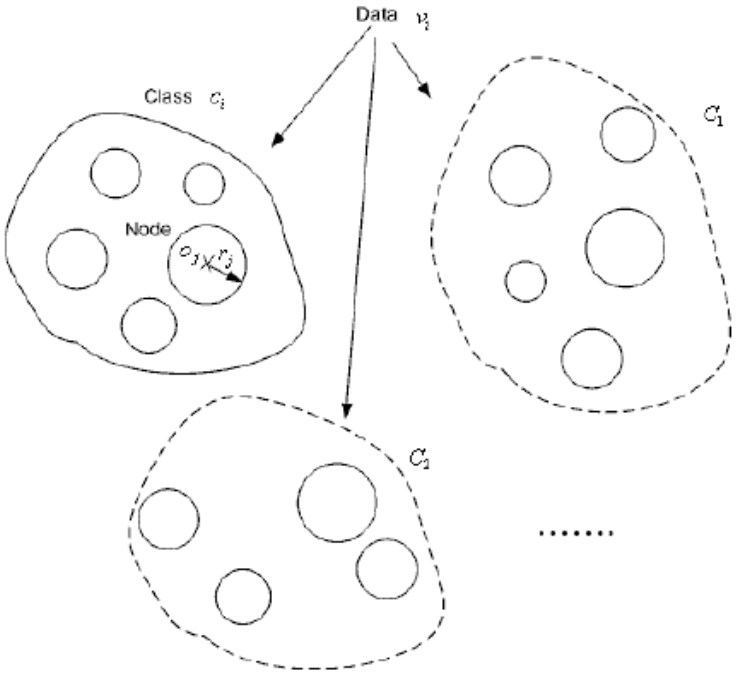


Fig.2.7: An example of clusters evolved in ECF for a classification task in robotics (Huang, Song, & Kasabov, 2005).

2.3.3. Personalised Modelling

The personalised modelling approach is one type of local modelling that is created for every single new input vector of the problem space based on its nearest neighbours using the transductive reasoning approach (Kasabov, 2007). *K-nearest neighbor* (KNN) is the simplest personalised modelling algorithm and has been successfully extended, as *Weighted K-Nearest Neighbour* (WKNN) (Dudani, 1976) and *Weighted-Weighted K-Nearest Neighbour* (WWKNN) (Kasabov, 2007), which will be studied further here.

KNN is simple, quick, and often effective. There are many cases in which its performance is at least as good as other more sophisticated algorithms. Based on the KNN algorithm, WKNN is created as robust to noisy learning samples as it takes the weighted average of *k*-nearest neighbours to the testing samples, and so it can easily smooth out the impact of isolated learning samples (Duda & Hart, 1973). In addition, WKNN is able to model complex functions by using a collection of less complex approximations (Wagacha, 2003). However, one major drawback of WKNN is that if the distribution of the class labels in the problem space is unbalanced, this algorithm may tend to favour the larger class, resulting in poor results (Hand & Vinciotti, 2003). WWKNN is a personalised profile of the variable importance that can be derived for every new input vector that represents a new piece of personalised knowledge. However, there are a number of opening questions that need to be considered when implementing the personalised WWKNN algorithm, such as defining the optimal number of nearest neighbours and the optimal number of features.

2.3.3.1. KNN

KNN is a supervised learning algorithm that has been successfully used for classifying sets of samples based on nearest training samples in a multi-dimensional feature space, and was originally proposed by Fix and Hodges in 1951. The basic idea behind the KNN algorithm is:

- ✧ Firstly, a set of pairs features (e.g. $(x_1, y_1), \ldots, (x_n, y_n)$) are defined to specify each data point, and each of those data points are identified by the class labels $C = \{c_1, \ldots, c_n\}$.

- ✧ Secondly, a distance measure is chosen (e.g. Euclidean distance, or Manhattan distance) to measure the similarity of those data points based on all their features.

Euclidean distance:	**Manhattan distance:**
$D(x,y)=\sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}$	$D(x,y)=\sum_{i=1}^{n}\lvert x_i-y_i\rvert$

(2.8)

where $D(x, y)$ is the distance between the two objectives x and y; x_i and y_i are the values of attributes i in cases x and y, respectively; $i = 1$ to n is a set of attributes in both objectives.

✧ Finally, the *k*-nearest neighbours are found for a target data point by analyzing similarity and using the majority voting rule to determine which class the target data point belongs to.

Figure 2.8 illustrates an overview of KNN: if $k = 5$, the class of the target vector *v* (represented as ▲) is determined by identifying its five nearest neighbours which are classified as ■ (where ■ and ● represent two classes, respectively).

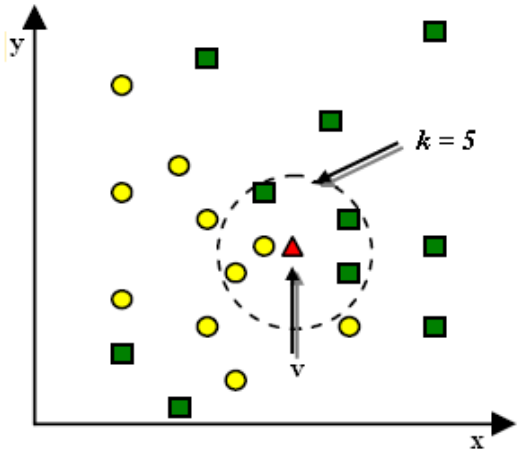


Fig.2.8: An example of the KNN classification task. Each vector is represented by a two-dimensional point within a Euclidean space.

2.3.3.2. WKNN

WKNN is designed based on the transductive reasoning approach, which has been widely used to evaluate the output of a model focusing on solely an individual point of a problem space using information related to this point (Vapnik, 1998). In the WKNN algorithm, each single vector requires a local model that is able to best fit each new input vector rather than a global model, thus each those new input vector can be matched to an individual model without taking any specific information about existing vectors into account. In contrast to the KNN algorithm, the output of a new input vector is calculated not only dependent upon its *k*-nearest neighbour vectors, but also upon the distance between the existing vectors and the new

input vector which is represented as a weight vector *w*, this being the basic idea behind the WKNN algorithm.

Mathematically, the WKNN algorithm can be formulated with the equation:

$$Output = \sum_{j=1,\ldots,k_i} w_j y_j \tag{2.9}$$

where k_i represents the number of nearest neighbours; w_j denotes the weight that is calculated based on the distance from the new input vector:

$$w_j = \left[\max(d) - (d_j - \min(d)) \right] / \max(d) \tag{2.10}$$

where $d = \left[d_{1,\ldots,} d_{k_i} \right]$ represents the distance between the new input vector and k_i ; the parameters max(*d*) and min(*d*) represent the maximum and minimum values in *d*, respectively.

WKNN has been successfully applied to medical and clinical applications to diagnose an individual patient in order to provide an accurate individual treatment. In addition, it is widely applied to the stock market to predict a stock index for a single target day.

2.3.3.3. WWKNN

WWKNN is a novel personalised modelling algorithm which was proposed by Professor Kasabov in 2007. The basic idea behind this algorithm is: the output of each new input vector is measured not only dependent upon its *k*-nearest neighbours, but also upon the distance between the existing vectors and the new input vectors, and also the power of each vector which is weighted according to its importance within the sub-space (local space) to which the new input vector belongs. If we assume that all the variables from a data set are used and the distance of vectors is calculated in a *V*-dimensional space with all input variables having the same impact on the output variables. However, the different variables might vary in importance when classifying vectors into classes if these variables are ranked by their discriminative power in classifying vectors over the entire *V*-dimensional Euclidean space. As a result, it can be seen that variables might have a

different ranking when we measure the discriminative power of the same variables for a sub-space of the problem space. The output of each new input vector can be calculated by using this type of ranking within the neighbourhood of *k*-nearest neighbour vectors.

The WWKNN algorithm is based on the following formulas:

$$d_j = \sqrt{\sum_{l=1\ldots n}^{k} c_{i,l}(x_l - x_{j,l})^2} \tag{2.11}$$

$$C_i = (c_{i,l}, c_{i,l}, \ldots c_{i,n}) \tag{2.12}$$

where d_j is the distance between the new input vector x_i and its nearest neighbour vector x_j, k represents as the number of nearest neighbours, whereas the parameter $c_{i,l}$ is the coefficient weighing variable x_l in relation to its nearest neighbour vector x_i which is calculated using the Signal-to-Noise-Ratio (SNR) supervised method to rank each variable across all vectors in the neighbourhood data set D_i:

$$C_{i,l} = S_1 / \sum S_1 (l = 1, 2, \ldots, n) \tag{2.13}$$

$$S_1 = \left| x_1^{(class1)} - x_1^{(class2)} \right| / (Std_1^{(class1)} + Std_1^{(class2)}) \tag{2.14}$$

where the parameters $x_1^{(class1)}$ and $x_1^{(class2)}$ represent the mean values of variable x_l for the samples from Class 1 and Class 2, respectively. In addition, the parameters $Std_1^{(class1)}$ and $Std_1^{(class2)}$ represent the standard deviation in data set D_i belonging to Class 1 and Class 2, respectively.

2.4. Personalised Knowledge Discovery Through Personalised Modelling

The literature suggests that inductive modelling is concerned with the creation of a global model which is derived from an entire problem space. The model obtained is then recalled for application to every new input data. In most cases, a global model is developed based on the inductive modelling approach that covers the entire problem space and is denoted as a single function.

In contrast, transducive modelling is used to create a local model for every new input data based on the nearest neighbours within the existing problem space. The local model in this case indicates a sub-space (local space) of the given problem space. In general, personalised modelling is one type of local modelling, where the personalised model is developed only for an individual vector of the problem space. The basic philosophy behind the personalised modelling approach is that every person is different from all others, thus he/she will benefit from having a personalised model and personalised treatment. Examples of personalised modelling are: KNN (*K-nearest neighbor*), WKNN (*Weighted K-Nearest Neighbour*), and WWKNN (*Weighted-Weighted K-Nearest Neighbour*). KNN is defined as a method for classifying a set of samples based on nearest neighbours in a multi-dimensional feature space. In contrast, in the WKNN algorithm, the output of a new input vector is calculated not only dependent upon its nearest neighbours, but also upon the distance between the existing vectors and the new input vector (weight vector *w*). The WKNN algorithm has recently been successfully extended to the WWKNN algorithm (Kasabov, 2007a), where there is one more weight is involved, which is the power of each vector which is weighted according to its importance within the local space to which the new input vector belongs.

In this study, the personalised modelling is integrated with the GA, which is defined as a technique that mimics biological evolution as a problem-solving strategy. GA is able to manipulate many parameters simultaneously, such as the optimal number of parameters for: number of threshold, number of nearest neighbours and significant features that need to adopted for every personalised model. The novel GAPM system will be applied to a comparative analysis of classification accuracy between GA-based personalised modelling (WKNN and WWKNN) and global (SVM) and local modelling (ECF) on several data sets.

As mentioned above, under this hypothesis it is assumed that GA-based personalised modelling will provide better accuracy than the global and local modelling approaches. In addition, the GA-based personalised modelling will provide more precise personalised knowledge and a better understanding of meaningful information.

2.5. Summary

In this chapter, a comparison between the inductive and transductive inference approaches was presented, including an introduction to the basic theory behind both

approaches, how they work, their areas of application and a description of the various algorithms based on these approaches. Furthermore, a number of opening questions that need to be considered when implementing transductive inference models were brought forward and these questions will be studied further in the following chapter.

Chapter 3

Feature Selection, Cross-Validation, and Optimization Methods: A Review

3.1. Introduction

Chapter 2 introduced a literature review of personalised modelling methods which included a comparison of inductive modelling and transductive modelling approaches, how they work, and their applications. In this chapter, a number of other techniques involved in this study are reviewed, including an overview of the feature selection procedures, cross-validation techniques, and genetic algorithm.

3.2. Overview of Feature Selection Methods

During the last few years, feature selection techniques in the machine learning field have motivated much study, and have become increasingly necessary to various bioinformatics applications especially. Nowadays, there are a growing number of applications for this technique in many different fields, such as data mining (Chen, Han & Yu, 1996; Provost & Kolluri, 1999), pattern recognition (Stearns, 1976; Ferri et al., 1994), and text learning (Yang & Pedersen, 1997). In general, a feature selection technique is defined as a fundamental step of the data mining process to find an optimal set of features, using certain learning algorithms from a given set of features. The primary goals of this technique are described as follows:

- ✧ to improve classification or prediction accuracy
- ✧ to speed up and reduce the cost of learning stages
- ✧ to avoid over-fitting and improve classification or prediction model performance
- ✧ to reduce the dimensionality of the feature space and to indentify the relevant features to be applied for a successful classification or prediction task.

In order to efficiently and properly achieve the goals, the choice of an appropriate feature selection model, to describe a learning system and evaluate the performance of a feature subset, is regarded as an important decision in the domain of machine learning. In general, feature selection techniques are organized into two common models, depending on whether the machine learning algorithm is adopted as a part of the selection method: *filter* and *wrapper,* which are introduced in the following sections.

✧ **Filter Model**

In the filter model, feature selection and the classifier learning are separated in a feature subset, which means features are first selected and then the classification model is induced, based on the selected features. This type of feature selection approach is independent of any machine learning algorithms. Figure 3.1 presents the basic structure of a simple filter model, where the feature selection process starts with a given training set characterized by the full feature set, and then various feature subsets are generated and evaluated by using the feature subset generator and evaluator. The final evaluation of a specific feature subset is accomplished by training and testing a specific classification model. Finally, ultimate classification accuracy is estimated based on the test set.

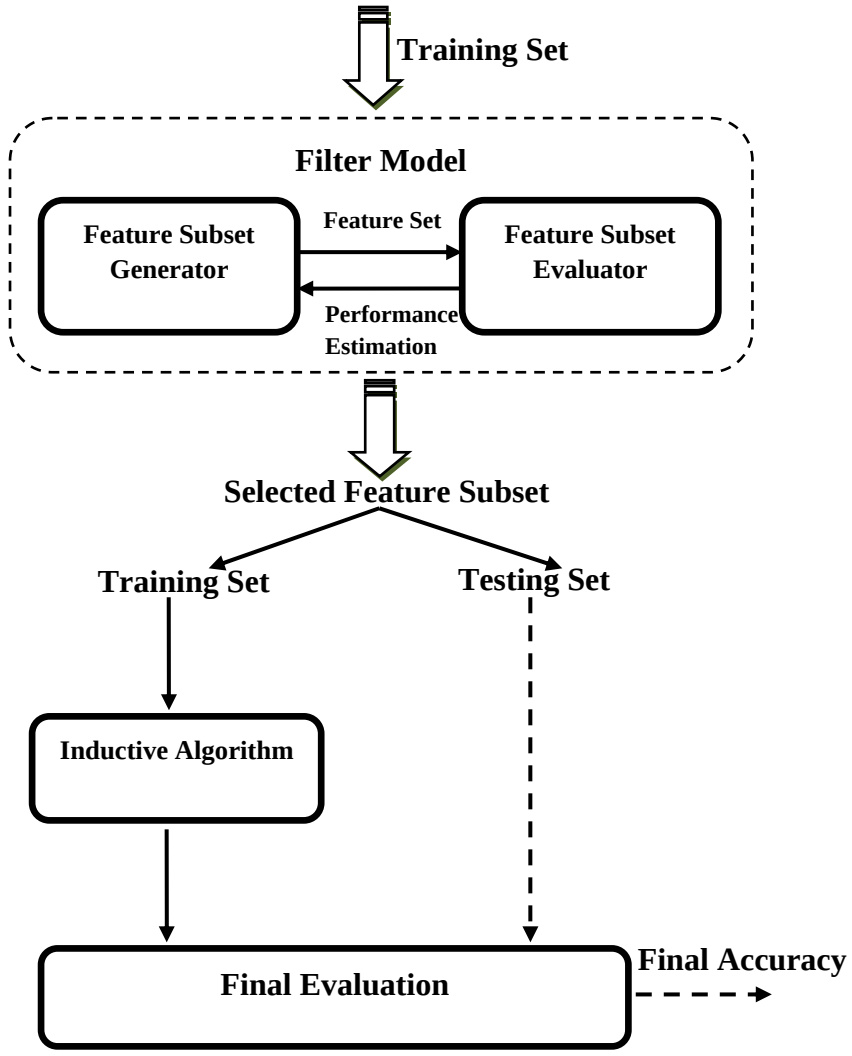


Fig.3.1: Basic structure of a simple filter model.

The filter model is one of the simplest and most commonly used feature selection techniques in microarray literature. The advantages of this model are that there is no

machine learning process involved while feature selection occurs, and time consumption is much lower than the wrapper model. However, a major drawback of this model is that it ignores interaction with classifiers, thus classification performance is not optimal.

The principal type of filter model is the *Signal-to-Noise Ratio* (SNR) ranking procedure. SNR is a supervised method, which is defined as a calculated ranking number for each variable to identify how well this variable distinguishes two different classes. Moreover, it is able to efficiently reduce the dimensionality of a data set. The basic idea behind this approach is that begins with the evaluation of an individual gene and iteratively examines the informative genes in the rest of data set in terms of statistic criterion. Mathematically speaking, it can be formulated with the following equation:

$$S_1 = \left| x_1^{(class1)} - x_1^{(class2)} \right| / \left(Std_1^{(class1)} + Std_1^{(class2)} \right) \quad (3.1)$$

where the parameters $x_1^{(class1)}$ and $x_1^{(class2)}$ represent the mean values of variable x_1 for the samples from Class 1 and Class 2, respectively. In addition, the parameters $Std_1^{(class1)}$ and $Std_1^{(class2)}$ represent the standard deviation in an available data set that belong to Class 1 and Class 2, respectively.

Most recently, SNR has been successfully applied in the area of molecular classification to evaluate the informativeness of each individual gene. Furthermore, the implementation of this approach has been widely used in various novel approaches, such as a hybrid method (Goh, Song et al., 2004) and a univariate ranking method (Lai et al., 2004). In this study, the SNR ranking procedure is applied on SVM and ECF feature selection process.

✧ **Wrapper Model**

In the wrapper model, a feature subset procedure is defined, and various feature subsets are generated and evaluated using a feature subset generator and evaluator. The evaluation of a specific feature subset is accomplished by training and testing with a specific classification model. A search algorithm is then wrapped around the classification model to search the space of all feature subsets. Figure 3.2 demonstrates the basic structure of a simple wrapper model.

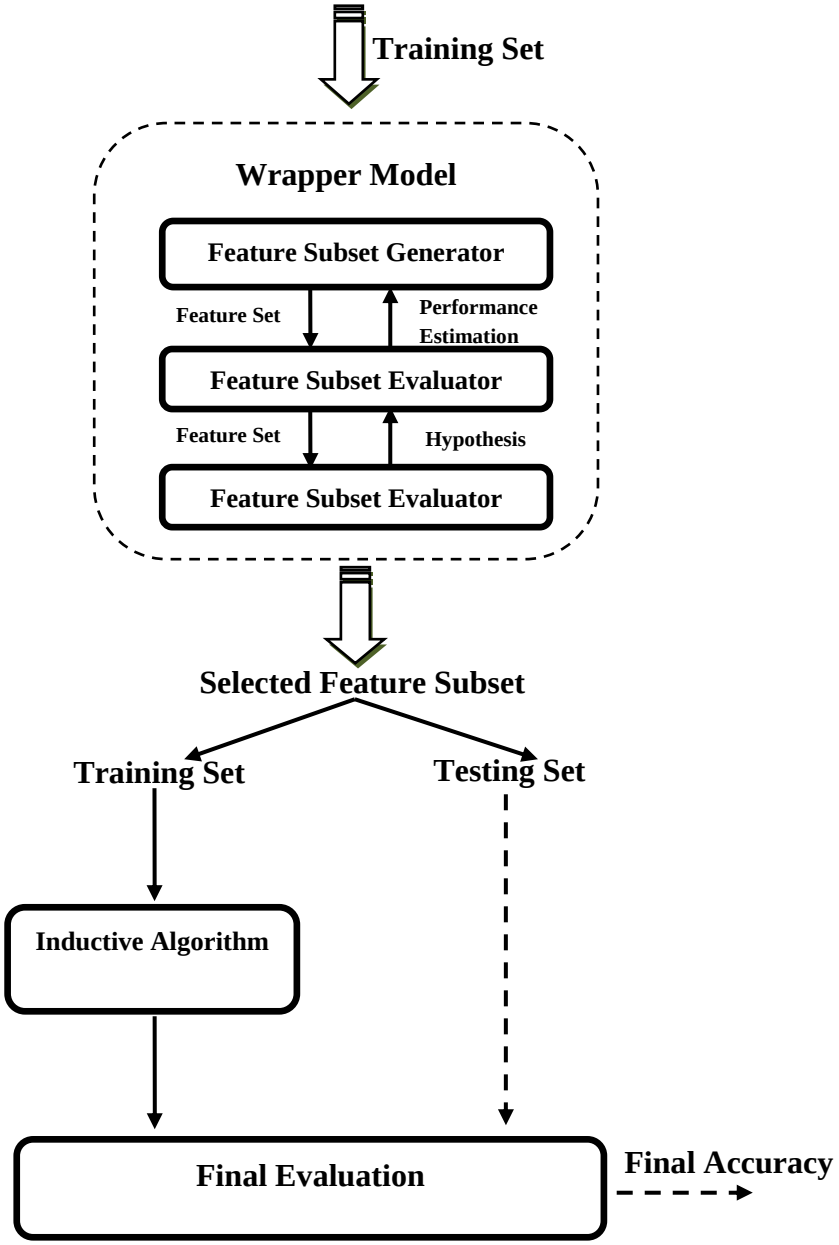


Fig.3.2: Basic structure of a simple wrapper model.

The wrapper model is one of the simplest and most commonly used feature selection techniques in machine learning applications. In contrast to the filter model, the advantages of the wrapper model are that it has interactions with the classifier while selecting features, as well as providing more accurate performance than the filter model. However, the disadvantages of this model are that it is very computationally expensive when compared with the filter model, and the evaluation results heavily depend on the inductive algorithm (also known as the central machine learning algorithm).

3.3. Overview of Cross-Validation Techniques

The choice of an appropriate data splitting/sampling strategy is critical for the verification of final experimental results (Braga-Neto, Hashimoto, Dougherty, Nguyen et al., 2004; Allison, Cui, Page, & Sabripour, 2006). Up to now, cross-validation is the most popular data splitting method which has been successfully applied in microarray data analysis, investigating the performance of neural networks, and estimating the generalization ability of a classifier (also known as generalization error). Cross-validation (also called "rotation estimation") is defined as an optimal method for measuring how well the results of a statistical analysis can generalize to an independent data set. The main idea behind this method is to split the available training set into two parts: one is a training set used to train the model, another is a testing set used for estimating the performance of the trained model. The primary goal of this method is to reduce generalization error and the possibility of over-fitting that is generally accomplished by sequentially leaving out parts of the original sample in the available data set and then performing a multi-variable analysis. This process goes on till all the samples in the data set have been estimated (Ransohoff, 2004).

In GAPM, this method is adopted to collaborate with the WKNN and WWKNN algorithms in order to decrease the generalization error in the classification stages, thus ensuring the models can provide the best accuracy throughout the experiments. A brief overview of two common cross-validation techniques is described below:

✧ ***K*-fold Cross-validation**

In *K*-fold cross-validation, the entire data set is roughly divided into *K* equal-sized subsets. For each of *K* experiments, an individual sub-sample serves as the testing data for testing the model, while the remaining $K-1$ sub-samples serve as training data. The process of cross-validation is repeated by *K* times/folds (commonly 10-fold is used) with each of the *K* sub-samples being estimated exactly once as the testing data (Figure 3.3 shows a general *K*-fold cross-validation process). Once all samples have been estimated, the overall generalization error is calculated as the average error rate across all *K* times experiments.

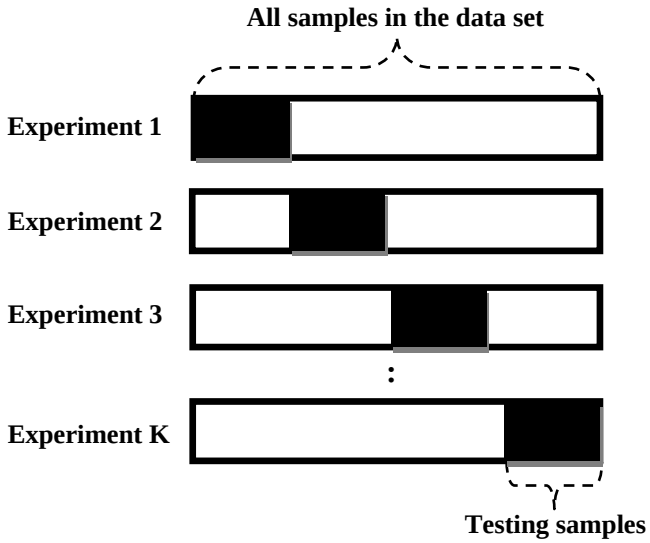


Fig.3.3: Overview of a general *K*-fold cross-validation process.

The advantage of this method is that all samples are used for both training and testing, and each sample is used for validation exactly once. On the other hand, the disadvantage of this method is that the training process needs to be repeated by *K* times computations to make an evaluation.

✧ ***Leave-One-Out Cross-Validation*** **(LOOCV)**

The leave-one-out cross-validation algorithm was originally proposed by Craven and Wahba in 1979, and defined as an almost unbiased validation schema for the optimal generalization ability of a classifier. Leave-One-Out Cross-Validation is in point of fact a type of *K*-fold cross-validation, where the number of folds (*K*) equals the number of samples (*N*) in an available data set. The basic idea behind this algorithm is to use *N*-1 samples for training and the remaining sample for testing each experiment. The process of LOOCV is repeated by *N* times, until every sample in the available data has been estimated, with all samples being used for training except one which is left out for testing (Figure 3.4 shows the general leave-one-out cross-validation process). Finally, the overall result is calculated by taking the average performance of all *N* times experiments.

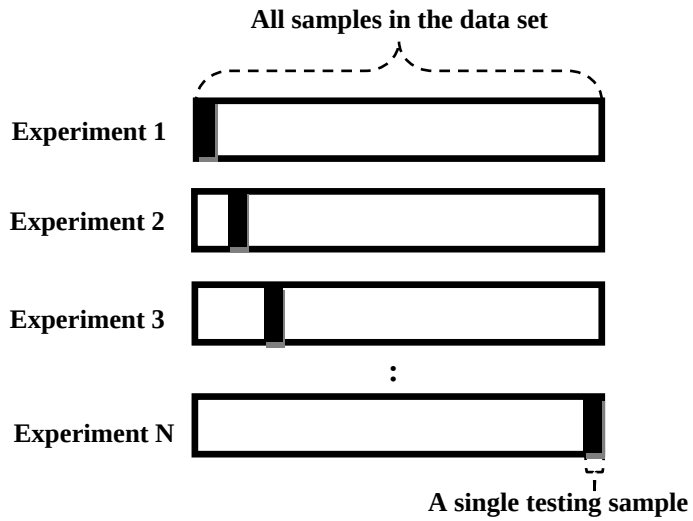


Fig.3.4: Overview of a general leave-one-out cross-validation process.

The advantage of this method is it makes good use of the available data as each pattern used is used both as training and testing data. However, the disadvantage of this algorithm is that it is very computationally expensive for use in neural networks due to the large amount number of times the training process is repeated.

3.4. Genetic Algorithms as Optimization Methods

The Genetic Algorithm (GA) is defined as an optimization and machine learning technique, which was primarily derived from the principles of natural selection and genetics (following the biological evolution theory originally proposed by Charles Darwin). This algorithm has been widely studied, experimented and applied to various fields by John Holland in the late 1950s and early 1960s, and one of his students, David Goldberg, at the University of Michigan in the 1960s and 1970s (Goldberg, 1989).

GA is an optimal method for solving optimization problems by manipulating from a population of *chromosomes* (e.g. strings of "0's" and "1's") to a new population using the principle of *natural selection* in cooperation with genetic operators like *crossover* and *mutation*. In other words, GA investigates a set of points called the *population*, and various biological genetic operators like *selection*, *crossover* and *mutation* are applied to the chromosomes in the population in order to provide better output solutions. In general, each chromosome is assigned a *fitness* value in the current population, which depends on how well that chromosome solves the problem. Figure 3.5 presents a flowchart of the basic structure of a typical genetic algorithm: given an initial population of chromosomes, GA solves an optimization problem by randomly selecting

chromosomes as parents based on their fitness function, the chromosomes with higher fitness are more likely to be selected as parents. Once the parent chromosomes are selected, the parents are combined to create offspring, thus *n* offspring are created through recombination/crossover of *n* parents. The *n* offspring are randomly mutated and survive to replace the *n* parents in the population. The process of reproduction and replacement goes on until one or more termination criterions are met (as described in section 4.3).

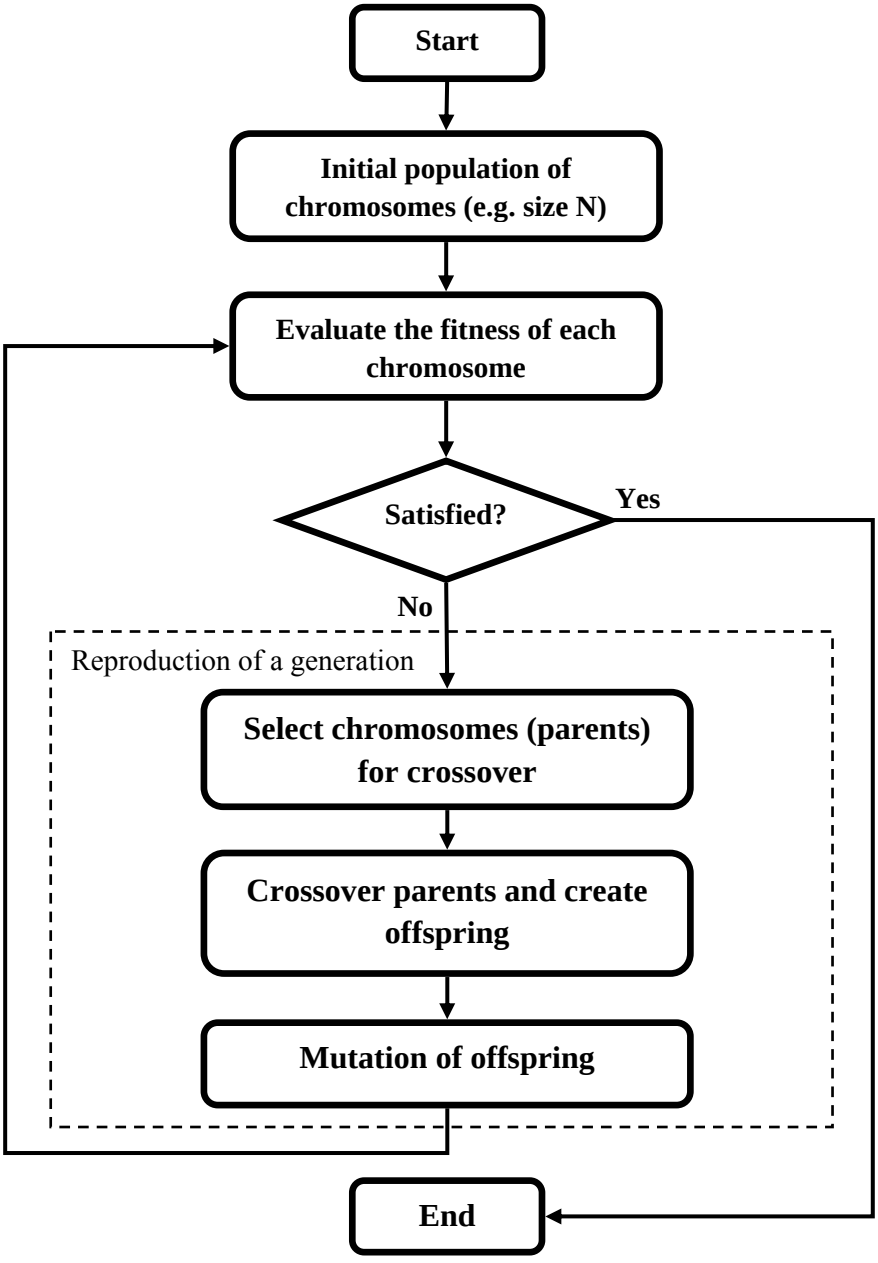


Fig.3.5: The basic structure of a simple genetic algorithm.

GA has been widely applied to various disciplines, such as science, engineering, economics, and political science to solve complex optimization problems. For instance,

GA has been widely utilized in game theory to evolve strategies for the Prisoner's Dilemma (which is a simple two−person game developed by Merrill Flood and Melvin Dresher in the 1950s), because it is seen as an optimal method for solving real world phenomena (Axelrod, 1984; Axelrod & Dion, 1988). In addition, it has been successfully applied in computer science to efficiently evolve sorting networks in the 1980s (Hillis, 1992). The reasons why GA has been applied in various fields are:

- ✧ GA performs well with various data, such as experimental data, numerically generated data, or analytical functions.
- ✧ GA is able to manipulate a large number of parameters simultaneously, including continuous and discontinuous parameters.
- ✧ GA is parallel, because it has multi-offspring, it can search the output solutions in many directions.
- ✧ By reason of parallelism, GA can perfectly estimate numerous schema at once, thus it performs especially well in solving problems (e.g. non-linear) where the space of all potential output solutions are too huge to search exhaustively in any reasonable amount of time.

As stated above, GA can be seen as an appropriate and popular method for dealing with complex optimization problems. It cannot only reduce the computational complexity and dimensions of the feature space, but it can also increase the performance of the classifiers. As a result, in this study, a typical GA is applied to serve as an optimal method to maximize the classification ability of GA-based personalised modelling (GAPM) by selecting an optimal number of parameters for: *number of threshold*, *number of nearest neighbours* and *significant features that need to be adopted for every personalised model*. In addition, it creates models that can provide the best accuracy using a combination of these optimal parameters, and it provides more precise personalised knowledge and a better understanding of meaningful information.

3.5. Summary

This chapter reviewed a set of methods involved in the GAPM, such as feature selection procedures, the cross-validation techniques, and genetic algorithm optimization technique. Feature selection refers to a fundamental step in the data mining process which selects an optimal set of features under certain learning algorithms. In this way we can efficiently reduce the number of features and only focus on the most important ones, thus achieving better accuracy and providing more precise personalised knowledge. Cross-validation is defined as a method to measure how well the results of

a statistical analysis can generalize to an independent data set. This technique is seen as a most popular data splitting method which has been successfully applied in various fields. GA is primarily derived from the principles of natural selection and genetics, which is seen as an appropriate and popular method for dealing with complex optimization problems.

Chapter 4

A Novel Framework and System for Personalised Modelling

4.1. Introduction

Chapter 2 presented a review of the transductive modelling approach, including how it differs from inductive modelling, how it works, its applications in various areas, as well as several opening questions that need to be considered. In this chapter, a novel GA-based framework and system for personalised modelling based on the transductive modelling approach is introduced in order to study and address the opening questions raised in Chapter 2. This novel system allows users to select and optimise the most important features and nearest neighbours for a single sample in relation to a certain problem based on a weighted variable distance measure. Thus, it can ensure greater accuracy and personalised knowledge when compared with the global modelling and local modelling approaches.

This chapter begins with the motivation behind the development of this novel framework and system. Secondly, a novel system called *GA-based Personalised Modelling* (GAPM) is introduced, and the settings and working of this system with WKNN and WWKNN as base models are explained. Finally, the knowledge discovery from the novel GAPM system is presented.

4.2. Motivation

As stated in Chapter 2, the transductive approach has been successfully implemented in medical and clinical decision support systems, time-series prediction problems, etc., where a personalised model is created for a single new input vector. This approach provides good accuracy for personalised models. However, there are a number of questions that need to be considered when implementing the transductive inference approach, such as "*how many nearest neighbours should be selected?*", and "*what features are important for a specific input vector?*" A novel framework and system for personalised modelling are developed to study and address these opening questions based on the existing method (see Figure 4.1). As presented in Figure 4.1, in the novel GAPM system, the models, selected features, and the numbers of nearest neighbours are integrated into one chromosome and optimized by using genetic algorithms in order to significantly improve the accuracy of personalised modelling when compared with the

global modelling and local modelling approaches. This can provide a better understanding of personalised knowledge. Theoretically speaking, the accuracy of a personalised model largely relies on some specific parameters that might have different values for every new input vector, such as the number of nearest neighbours, and number of selected features. As a result, it is essential to optimise those parameters in order to effectively improve the accuracy of a personalised model, as well as correctly derive personalised knowledge. Most recently, the genetic algorithm has been successfully adopted as an appropriate procedure to efficiently optimise those parameters for solving various classification or prediction tasks. In the next section, the novel system called *GA-based Personalised Modelling* (GAPM) is introduced in detail, and the working of this system with WKNN and WWKNN as base models is explained.

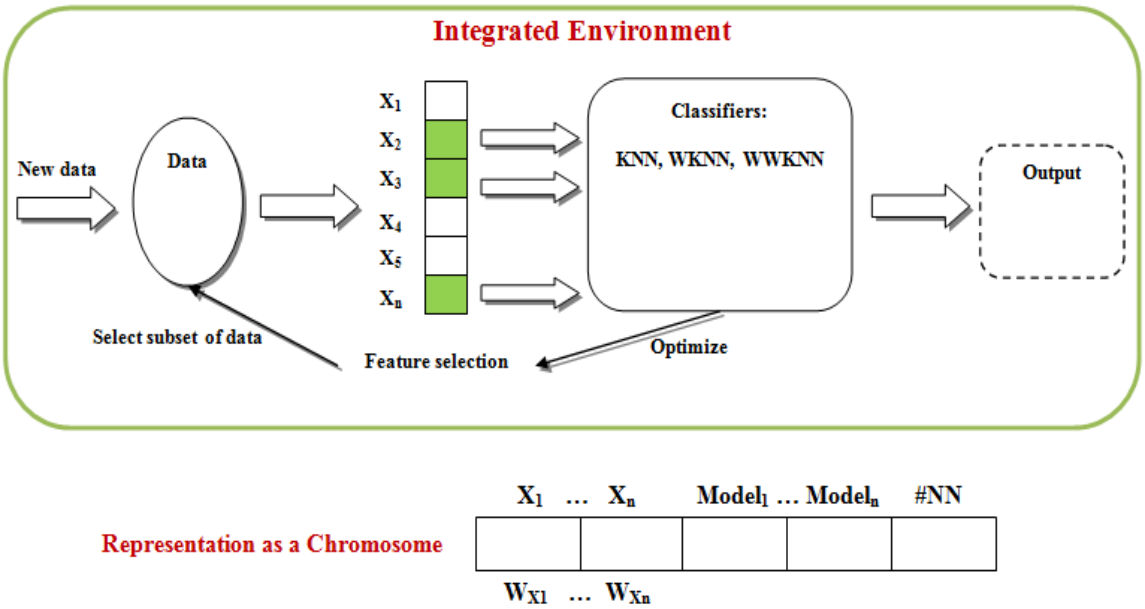


Fig.4.1: An overview of the novel GAPM system.

4.3. Settings and Operations of a Novel GAPM Framework and System

Settings of the Novel GAPM Framework and System

Before a description of the novel system and its working are given, the five basic components of GA in GAPM (*chromosome*, *fitness function*, *selection/reproduction*, *genetic operators (e.g. crossover and mutation)*, and *termination criteria*) are briefly introduced.

(1) Chromosome

To solve an optimization problem, GA usually begins by defining a population of chromosomes (also called genomes), which identifies a possible output solution to the problem that GA tries to solve. A chromosome is a set of parameters, and the parameter

set is to be coded as a finite sequence of values. In general, the chromosome is represented as strings of 0's and 1's (e.g. 0010110011 – a binary chromosome of length 10). Good coding is possibly the most important factor for the performance of a GA. Up until now, there have been various coding strategies proposed, but generally binary encoding is one of the most common and popular techniques.

In this study, the threshold (*T*), the total number of samples (*K*) and the total number of features (*F*) are used as the input parameters needing to be optimized using the proposed GAPM. Figure 4.2 shows the binary chromosome denotes the genotype of these three parameters: where g_T, g_K, g_F indicate the model parameters; n_T, n_K, n_F indicate the number of bits of the threshold, *k*-nearest neighbours and feature mask, respectively.

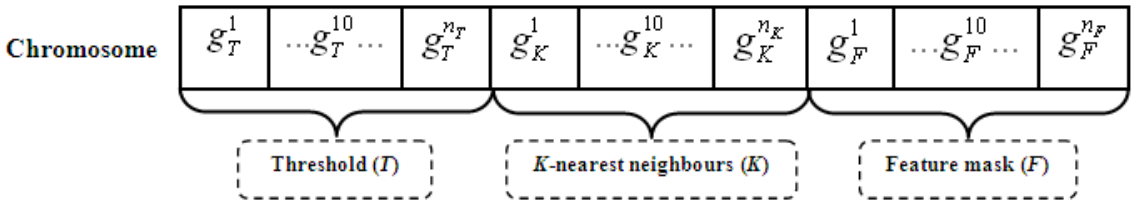


Fig.4.2: The chromosome comprising three parts: *T*, *K*, and *F*.

One of the most important steps in applying a GA is choosing a suitable encoding method to convert the real problem into biological terms. There are four commonly used encoding methods: binary encoding, permutation encoding, direct value encoding and tree encoding. Binary encoding is the most common and simplest method. In this method, every chromosome is represented as a string of bits, 0 or 1. In this study, the parameters {*T*, *K*, *F*} should be converted into phenotype by using the following equation: where P is the phenotype of bit string, whereas $\min_p$ and $\max_p$ represent the minimum and maximum values of the parameter respectively, *d* is the decimal value of the bit string, and *l* is the length of the bit string.

$$P = \min_p + ((\max_p - \min_p) * d) / 2^l - 1 \quad (4.1)$$

Generally, GA starts with a group of chromosomes known as the *population*. The initial population begins with a randomly selected set of bits for threshold, *K*-value and subset of features. The threshold ranges from a minimum value of 0.1 to the maximum value of 1. The *K*-value ranges from one to the maximum size of the sample in a problem space. The feature subset is initialized to one feature in each population as the starting point, thus the number of populations is equal to the number of features in a problem space to ensure that each feature has an equal opportunity of getting selected.

In order to efficiently select the most significant features, a genetic feature selection procedure is adopted in this study to solve the feature selection problem, employing the idea from Siedlecki and Slansky (1989). Each feature is represented by "0" for rejected features and "1" for selected features. GA is a powerful feature selection technique, especially when the dimensions of the original feature space are very high (Siedlecki & Sklansky, 1989). Figure 4.3 presents the number of bits of the threshold, *k*-nearest neighbours and feature mask in GAPM.

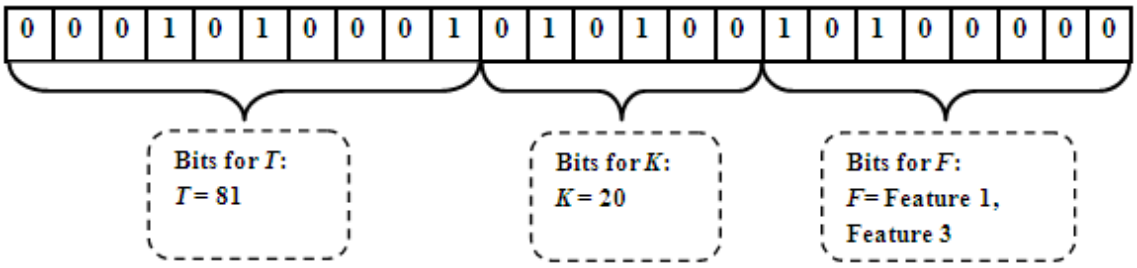


Fig.4.3: The number of bits of the threshold, *k*-nearest neighbours and feature mask in the GAPM.

(2) Fitness Function

Each of the chromosomes in a generation must be evaluated based on the fitness function. A fitness function determines how well each chromosome solves the problem. In general, the process of evaluation is accomplished by examining the classification accuracy of each chromosome, and averaging the accuracy achieved using a particular chromosome with an optimal number of threshold (*T*), *K*-value (*K*) and feature subset (*F*). A chromosome with a high fitness value has a high probability of being selected in the next generation.

(3) Selection

Selection is defined as a method to select chromosomes from the current population for reproduction. Assume that there is a population of size 4N, the selection procedure randomly selects four chromosomes to serve as parents based on their fitness values. As a result, four offspring are generated for the new population by using crossover and mutation genetic operators (described below). This selection-crossover-mutation cycle goes on until the new population contains 4N chromosomes. The chromosomes have a high fitness value and a high probability of being selected for reproduction.

In this study, the most common procedure – *roulette wheel selection* is adopted to select the individual parent chromosomes to be copied over into a new generation. In roulette wheel selection, the individuals are given a probability of being selected that is

generally taken to be directly proportional to their fitness value. Figure 4.4 illustrates the basic structure of a simple roulette wheel selection algorithm: the wheel has the same number of slots as the population size, where the size of each slot is proportional to the fitness value of the related chromosome in the population. A fit chromosome is selected by spinning the roulette wheel and noting the position of the arrow when the wheel stops.

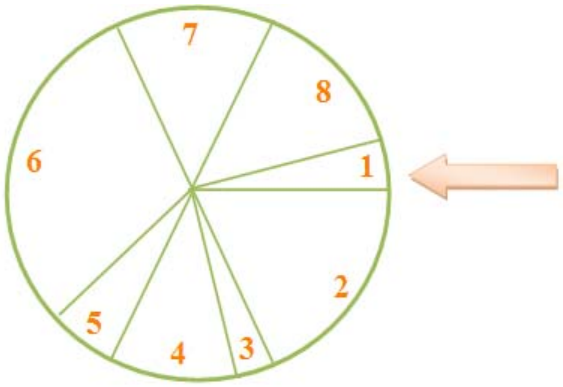


Fig.4.4: Overview of a simple roulette wheel selection algorithm.

(4) Genetic operators

Once a pair of fit chromosomes has been selected, they have to be randomly altered in order to improve their fitness for the next generation (also known as reproduction). There are two basic techniques to accomplish this task: *crossover* and *mutation*, which are described as follows:

✧ **Crossover**

The crossover operator is an important feature of GA, utilized to exchange genes between a randomly selected a pair of parent chromosomes by recombining parts of their genetic material. This operation is performed probabilistically, combining parts of two parent chromosomes to produce offspring. Generally, three types of crossover operator can be adopted to generate offspring from two randomly selected parent chromosomes: *single-point crossover*, *two-point crossover*, and *uniform crossover*.

In this study, the most common type of crossover, *single-point crossover* is used. In single-point crossover, a random point is chosen (also known as the crossover point) on the two selected parents to split the parents at this point. As shown in Figure 4.5, each child takes one part of a chromosome from each parent. Child 1 takes the head of the chromosome of Parent 1 and the tail of the chromosome of Parent 2, while Child 2 takes the head of Parent 2's chromosome and the tail of parent 1's chromosome. The

crossover rate is defined as a probability that is applied when the search algorithm uses a breeding rate to select chromosome(s) for crossover. In some cases, genetic searches begin with a low crossover rate and then increase the crossover rate if the average fitness value of the population does not significantly improve over a specified number of generations. In general, a high crossover rate may introduce new strings more quickly into the population, while a low crossover rate may sometimes cause stagnation. Therefore, it is essential to correctly define the crossover rate that will facilitate optimal performance. A default crossover rate of "0.8" is chosen in this study.

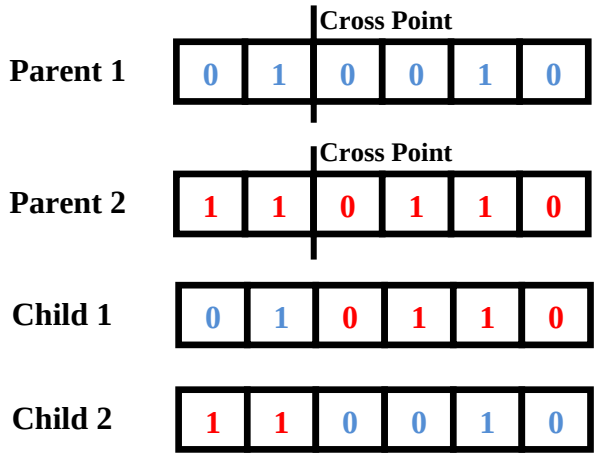


Fig.4.5: Example of a single-point crossover scheme.

✧ Mutation

Mutation is another major genetic operator utilized to generate new offspring from a single parent. This operation is critical as it ensures that genes are not all exactly the same in the new population. By looping through all alleles, if one allele is selected for mutation, it can be changed either by a small amount value or replaced with a new value. As shown in Figure 4.6, positions 3 and 7 of the chromosome have been subjected to mutation.

The mutation rate is defined as a probability. It is used when quite high since every chromosome is likely to have at least one of its genes modified through a mutation technique. In most cases, the mutation rate should be very low in order to sustain genetic diversity but not overwhelm the population with too much noise. In general, a high mutation rate may reduce convergence time, whereas a low mutation rate may avoid any bit positions getting stuck to a single value. In this study, a mutation rate of "0.01" has been chosen that has been identified as a default setting in most cases.

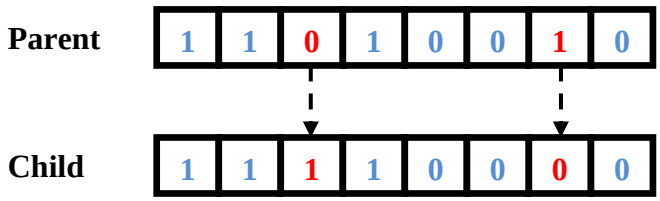


Fig.4.6: Example of a simple mutation operation.

(5) Termination Criteria

In general, the process of chromosome generation goes on until one or more of the following termination conditions are met:

- ✧ The use of an allotted amount of CPU time.
- ✧ The number of generations is greater than some pre-defined threshold.
- ✧ The number of generations has reached maximum-number, maximum-generation, and so on.
- ✧ The average *fitness* value of a population becomes more or less constant over a particular number of generations.

Once the termination criteria are met, the chromosome with the best fitness value of all generations is returned as the best population of the output solution. In this study, based on the maximization of the fitness function, GAPM proceeds via its generations to result in an optimal set of values for the number of threshold (*T*), the number of *k*-nearest neighbours (*K*) and the number of features (*F*) in a problem space. The GAPM stops running when all generations are complete.

Operation of the Novel GAPM Framework and System

WKNN and WWKNN are the base models of GAPM. The discovery of effective weight vectors for a WKNN algorithm is a difficult optimization problem with a very large search space. This is just the sort of problem that GA is good at, thus it would seem that GA working together with a WKNN algorithm is a strategy for a high-performance classification algorithm. In contrast, WWKNN is a novel personalised modelling algorithm recently proposed by KEDRI. The basic idea behind the WWKNN algorithm is quite similar to the WKNN algorithm. As a result, in this study, both WKNN and WWKNN algorithms work together with the genetic algorithm to efficiently maximize classification performance. The purpose of this novel GA-based personalised modelling system is to allow users to select and optimise the most

important features and nearest neighbours of a single sample in relation to a certain problem based on a weighted variable distance measure.

There are two major contributions made by this novel GAPM system: (1) It allows users to use GA–optimized WKNN and WWKNN algorithms to create classification models to test classification accuracy in order to provide more accurate and predictive knowledge and information for investigators, and (2) It also allows users to create personalised prediction models for each new individual input vector by using WKNN and WWKNN predication algorithms. However, one limitation is that the genetic algorithm does not collaborate with the two base algorithms to create a personalised prediction model for a single vector, thus the output of a single target vector might not be optimal.

The two contributions of this novel GAPM system are described as follows:

1) Using GAPM to Create Personalised Classification Models

Figure 4.7 presents an overview flowchart of GA-based feature selection and parameter optimization for WKNN and WWKNN algorithms in GAPM. The basic steps involved to create a personalised classification model are introduced below:

Step 1: *Data splitting*. Firstly, the entire data set is randomly split into two parts: "model creation" (e.g. 90%) for training and "model validation" (e.g. 10%) for testing using an interleave data splitting method. The training set (90%) is then loaded into the novel GAPM system, the system further randomly selects part of the data (e.g. 70%) for training and the remaining (e.g. 30%) for testing to train the classifiers. In contrast, the testing set (10%) is used to calculate final overall classification accuracy. The main advantage of data splitting is to ensure a totally unbiased verification process for all experiments.

Step 2: *Converting genotype to phenotype*. Once the data is loaded, the parameters (Threshold (T), k-nearest neighbours (K) and feature mask (F)) need to be converted from genotype to phenotype using Equation 4.1.

Step 3: *Selecting feature subset*. Once each chromosome is converted into a phenotype, an initial subset of features is established.

Step 4: *Evaluating fitness function*. For each chromosome indicating T, K, and F, the training data (e.g. 70%) is used to train the WKNN and WWKNN classifiers, while the testing data (e.g. 30%) is used to evaluate classification accuracy. Once accuracy is calculated, each chromosome is evaluated by the fitness function.

Step 5: *Meeting termination criteria*. If the termination criteria are met, the entire process is stopped; otherwise it carries on with the next generation.

Once the termination criteria are met, the optimal number of parameters $\{T, K, F\}$ is further adopted to evaluate the final output by using WKNN and WWKNN algorithms that applied to the testing set (10%).

Step 6: *Adopting genetic operators*. The GAPM investigates better output solutions by using genetic operators, such as selection, crossover, and mutation.

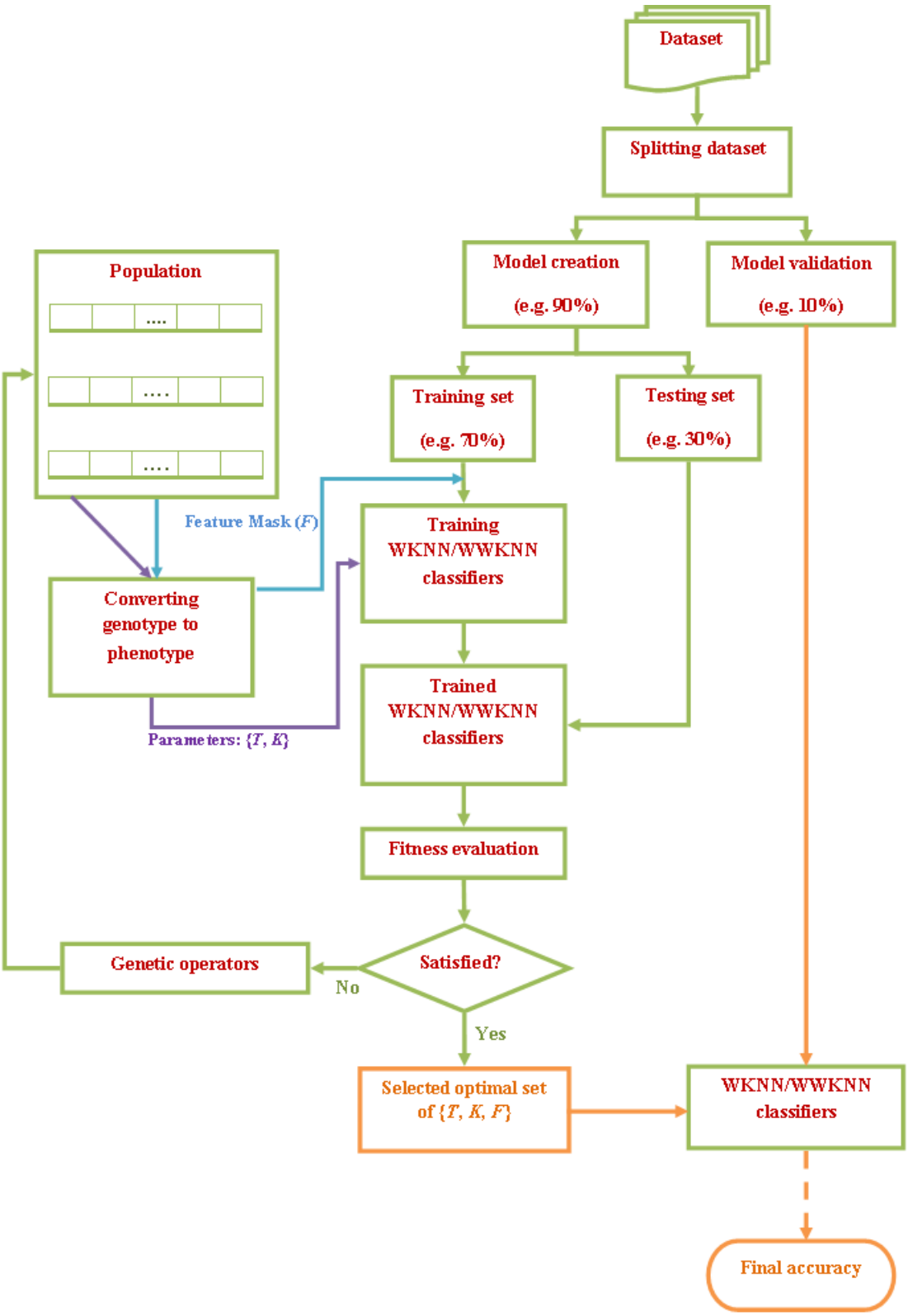


Fig.4.7: Flowchart of GA-based integrated feature selection and parameter optimization for the WKNN and WWKNN classifiers.

2) Using GAPM to Create Personalised Prediction Models

The main steps for creating personalised prediction models for an individual input vector X_i using WKNN and WWKNN algorithms are presented in Figure 4.8:

Step 1: Define the nearest neighbour N_i for a target single vector X_i in a data set D_1, meaning the target vector X_i is used for testing and the rest are used for training. Therefore, D_2 is a new data set which is derived after defining the nearest neighbour for the target vector from the data set D_1.

Step 2: The system automatically selects the most significant features V_i by using the *Signal-to-Noise Ratio* (SNR) ranking procedures (as described in section 3.2) built in the D_2 data set. Thus, D_3 is a new data set which is derived after ranking the most important features from the data set D_2.

The entire feature selection process involves several steps:

(1) Firstly, the process begins by applying SNR ranking procedures to arrange all the available features in descending order.

(2) Once all the features are ranked in correct order, the top three features are applied to train the WKNN and WWKNN prediction methods in a leave-one-out mode (randomly selecting one sample from D_3 for testing and the rest for training, but without using the target input vector X_i) to test the average accuracy of the model built in the D_3 data set. Thus, the accuracy obtained from these three features form the base classification accuracy.

(3) The next feature from the ranked set is added to the previous three features to calculate accuracy, if the accuracy is better than the base classification accuracy, then the feature is selected into the selection pool. This process goes on iteratively for the remaining features until all features are studied.

Step 3: If the accuracy calculated in the Step 2 does not satisfy users, go back to the Step 1 to find the neighbourhood of X_i and the feature selection process is repeated until the best accuracy is achieved. In contrast, If the accuracy satisfies users, then apply the WKNN and WWKNN prediction algorithm in

a leave-one-out mode once more (X_i is used for testing, and the entire D_3 data set is used for training) to calculate the output Y_i for the target input vector X_i by using the optimal number of nearest neighbours N_i and the optimal number of features V_i.

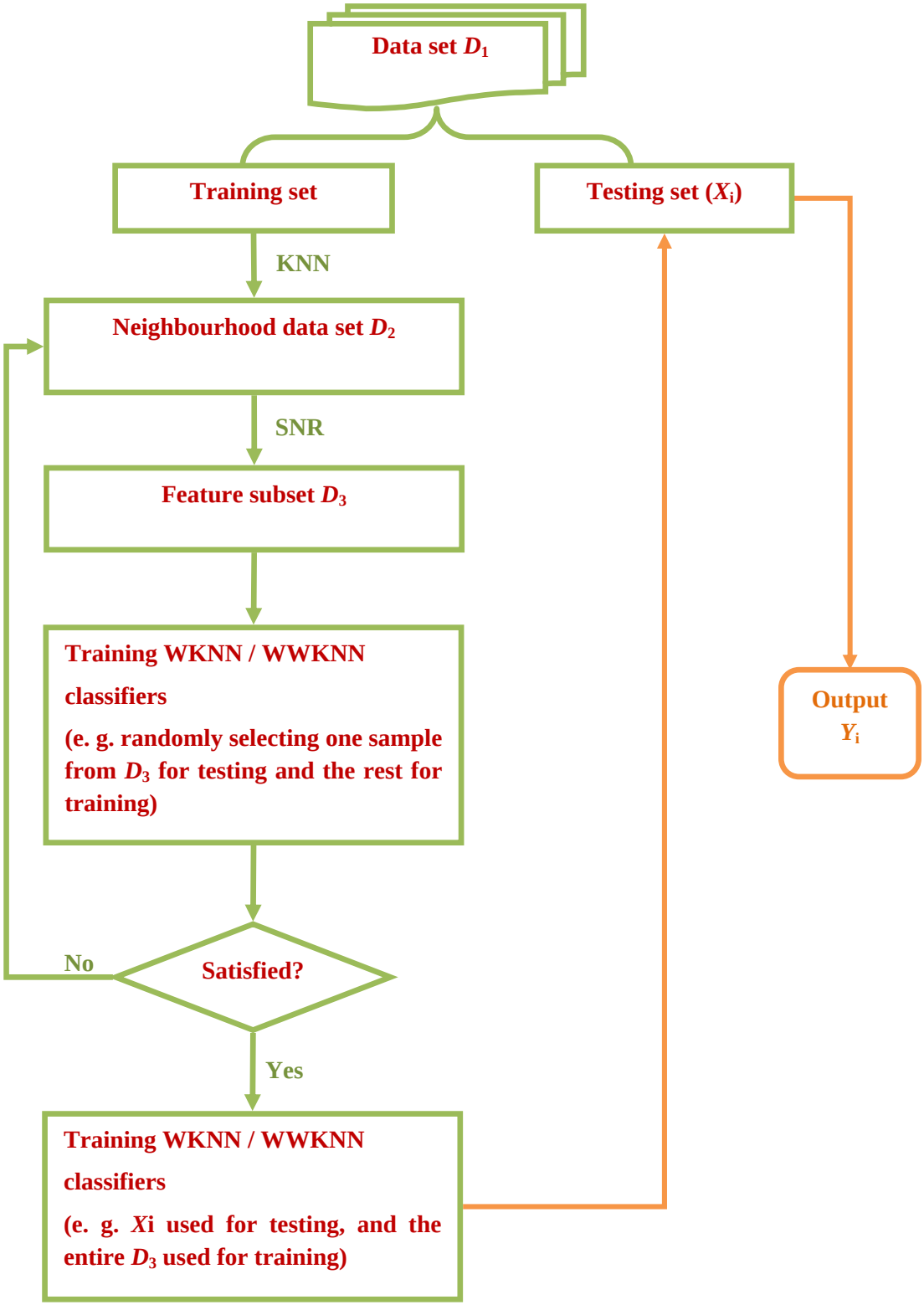


Fig.4.8: Flowchart of creating personalised prediction models for a new input vector X_i using the WKNN and WWKNN algorithms.

4.4. Knowledge Discovery from the Novel GAPM System

In this research a novel GAPM system based on the transductive reasoning approach has been developed. This approach allows selection and ranking of the most important features and nearest neighbours for a new input sample in relation to a certain problem such as a classification or a prediction task.

This novel system integrates the typical GA with weighted variable distance measure approaches, WKNN and WWKNN, to efficiently maximize classification performance. The GA has been incorporated into the personalized modelling system as it has the potential to generate an optimal number of nearest neighbours, an optimal set of features, and simultaneously perform parameter selection for WKNN and WWKNN. As mentioned above, the novel GAPM system is developed under the hypothesis that the GAPM system provides better accuracy when compared with global modelling and local modelling approaches. In addition, this novel GAPM system provides more precise personalised knowledge and a better understanding of meaningful information.

As stated previously, the crossover rate and mutation rate are two important parameters in the GA. The values are dependent upon the kind of problem given. In general, the performance of a typical GA might be significantly affected by changing the specific crossover rate and/or mutation rate. One limitation of this study is that only one default crossover rate (0.8) and mutation rate (0.01) are chosen to investigate the performance of a typical GA. As the right choice of parameter values is an important issue in the GA, future research needs to look at the relationship between the crossover and mutation rates, and how well a typical GA performs by using a different range of crossover and mutation rates.

4.5. Summary

In this chapter, a novel GA-based framework and system called GAPM was presented. This novel system allows users to select and optimise the most important features and nearest neighbours of a single sample in relation to a certain problem based on a weighted variable distance measure. The two major hypotheses held here are: (1) the novel GAPM system might provide better accuracy when compared with global modelling and local modelling approaches, and (2) this novel system might also provide more precise personalised knowledge and a better understanding of meaningful information. There is a need to look at the relationship between the crossover and

mutation rates, and how well a typical GA performs by using a different range of crossover and mutation rates.

Chapter 5

Software Implementation of the Novel GAPM System

5.1. Introduction

This chapter begins with a presentation of the MATLAB implementation of the novel GAPM system. This is followed by an experiment run on a benchmark data set (e.g. Sonar) using NeuCom and the novel GA-based personalised modelling system to compare the classification accuracy of different algorithms. The results with detailed analysis are described in the final section.

5.2. MATLAB Implementation of the Novel GAPM System

The novel framework and system was developed using MATLAB, based on the transductive inference approach and the evolutionary algorithms of genetic algorithm for parameter optimization. MATLAB is a high-performance and easy-to-use language that has been widely applied to various areas, and it is a standard instructional tool for introductory and advanced courses in mathematics, industry, engineering, and science. Figures 5.1 and 5.2 present the main graphical user interface (GUI) for GA-optimized WKNN and WWKNN algorithms, respectively.

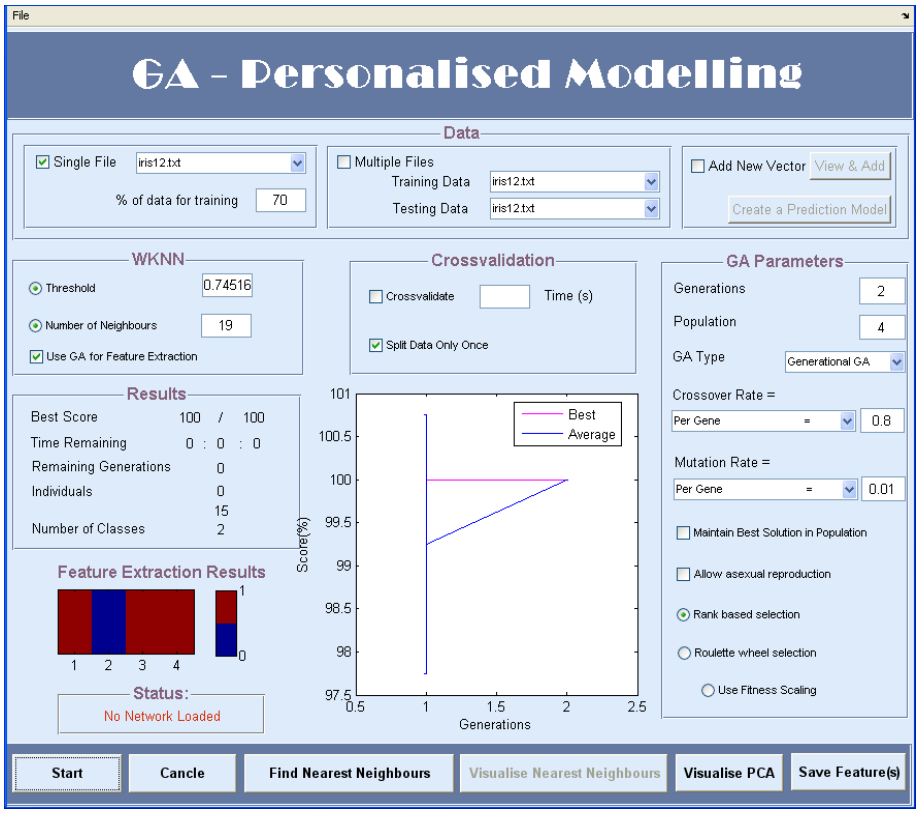


Fig.5.1: Main GUI screenshot for the GA-optimized WKNN algorithm.

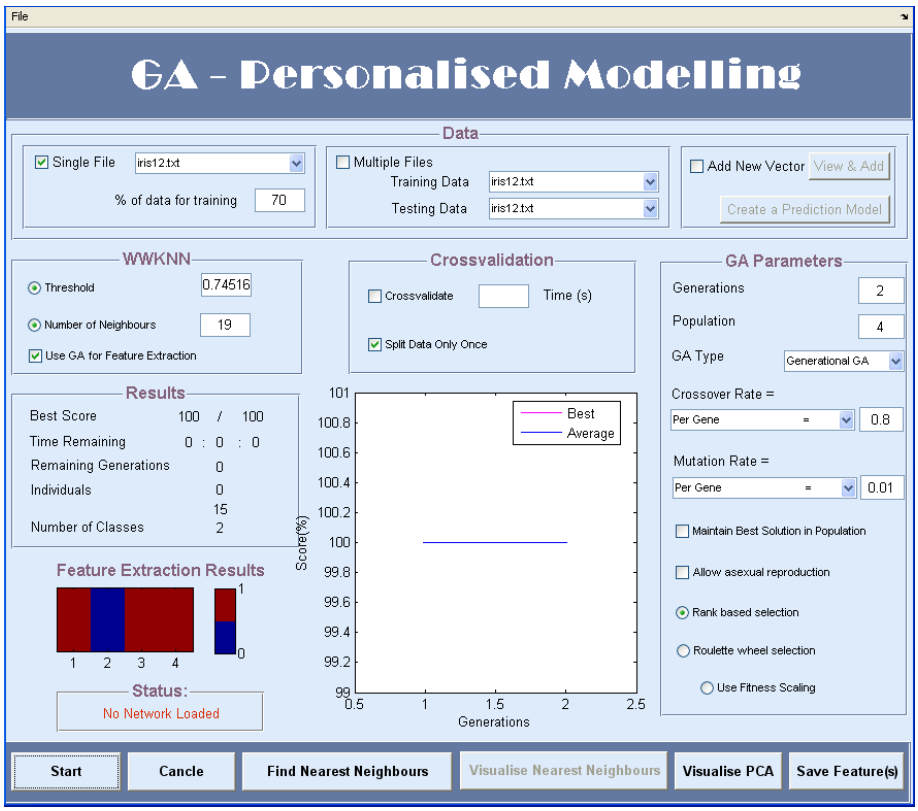


Fig.5.2: Main GUI screenshot for the GA-optimized WWKNN algorithm.

The main stages for creating personalised classification models and personalised prediction models using GAPM are described as follows:

1) Creating Personalised Classification Models

Step 1: Load Data Set

The input data set needs to be pre-processed (e.g. data normalization and data splitting) before being loaded into the GAPM. This ensures that there are no missing values in the data set, and that it is a totally unbiased verification process for all experiments. Once the data set is pre-processed, it is ready to be loaded into the GAPM in ".txt" format. The system allows users to load a single file (e.g. 70% of randomly selected data for training and 30% for testing), as well as allowing users to load multiple files (one for training and another for testing). Once the data set is loaded, it can be visualized by clicking the "Visualise PCA" button to see how the entire data (only the top two features of samples are displayed) are distributed (see Figure 5.3). Principal Components Analysis (PCA) is a powerful statistical technique used for reducing large and high-dimensional data set by removing redundancies and identifying correlation among a number of variables. The applications of this technique have been widely adopted in various scientific areas, such as image processing and compression, face recognition, and molecular dynamics. Most recently, this technique has been applied to

gene expression analysis to compute an alternative representation of the data by using a much smaller set of variables.

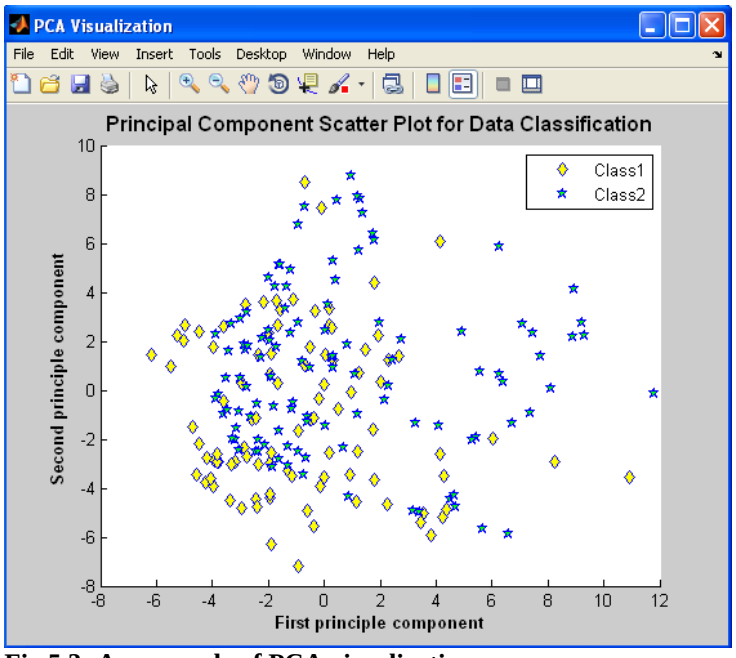
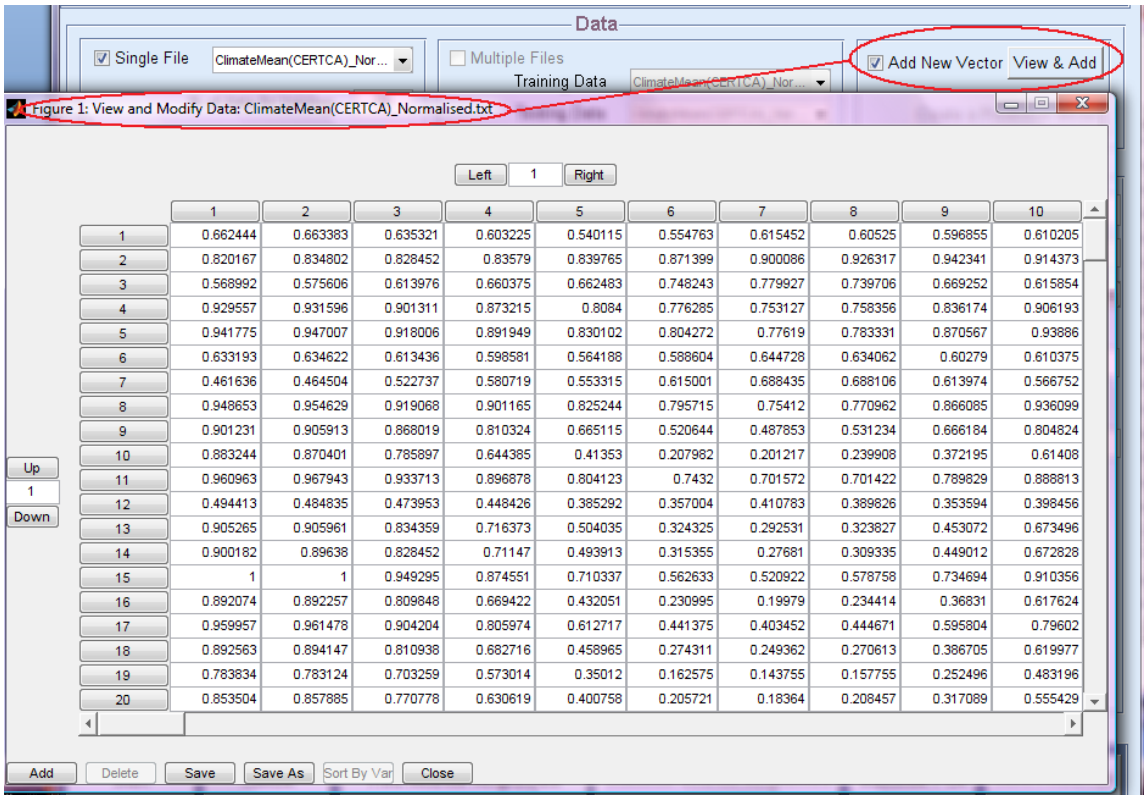


Fig.5.3: An example of PCA visualization.

In addition, the system also allows users to view or modify the loaded data set by ticking the "Add New Vector" check box and clicking the "View & Add" button (see Figure 5.4).



	1	2	3	4	5	6	7	8	9	10
1	0.662444	0.663383	0.635321	0.603225	0.540115	0.554763	0.615452	0.60525	0.596855	0.610205
2	0.820167	0.834802	0.828452	0.83579	0.839765	0.871399	0.900086	0.926317	0.942341	0.914373
3	0.568992	0.575606	0.613976	0.660375	0.662483	0.748243	0.779927	0.739706	0.669252	0.615854
4	0.929557	0.931596	0.901311	0.873215	0.8084	0.776285	0.753127	0.758356	0.836174	0.906193
5	0.941775	0.947007	0.918006	0.891949	0.830102	0.804272	0.77619	0.783331	0.870567	0.93886
6	0.633193	0.634622	0.613436	0.598581	0.564188	0.588604	0.644728	0.634062	0.60279	0.610375
7	0.461636	0.464504	0.522737	0.580719	0.553315	0.615001	0.688435	0.688106	0.613974	0.566752
8	0.948653	0.954629	0.919068	0.901165	0.825244	0.795715	0.75412	0.770962	0.866085	0.936099
9	0.901231	0.905913	0.868019	0.810324	0.665115	0.520644	0.487853	0.531234	0.666184	0.804824
10	0.883244	0.870401	0.785897	0.644385	0.41353	0.207982	0.201217	0.239908	0.372195	0.61408
11	0.960963	0.967943	0.933713	0.896878	0.804123	0.7432	0.701572	0.701422	0.789829	0.888813
12	0.494413	0.484835	0.473953	0.448426	0.385292	0.357004	0.410783	0.389826	0.353594	0.398456
13	0.905265	0.905961	0.834359	0.716373	0.504035	0.324325	0.292531	0.323827	0.453072	0.673496
14	0.900182	0.89638	0.828452	0.71147	0.493913	0.315355	0.27681	0.309335	0.449012	0.672828
15	1	1	0.949295	0.874551	0.710337	0.562633	0.520922	0.578758	0.734694	0.910356
16	0.892074	0.892257	0.809848	0.669422	0.432051	0.230995	0.19979	0.234414	0.36831	0.617624
17	0.959957	0.961478	0.904204	0.805974	0.612717	0.441375	0.403452	0.444671	0.595804	0.79602
18	0.892563	0.894147	0.810938	0.682716	0.458965	0.274311	0.249362	0.270613	0.386705	0.619977
19	0.783834	0.783124	0.703259	0.573014	0.35012	0.162575	0.143755	0.157755	0.252496	0.483196
20	0.853504	0.857885	0.770778	0.630619	0.400758	0.205721	0.18364	0.208457	0.317089	0.555429

Fig.5.4: View and Modify the loaded data set.

Step 2: Parameter Settings

Once the data set is loaded, it is time to setup parameters for the proposed GAPM method, such as initializing the cross-validation times that collaborate with the GA–optimized WKNN and WWKNN classification algorithms to test classification accuracy. At this point, the user will also select the parameters for the genetic algorithms in terms of: the number of generations (which represents the termination criteria for the GA), the number of populations (which is determined based on the number of features available in the loaded data set), the crossover and mutation rates, and the selection procedure (see Figures 5.1 and 5.2).

Step 3: Start the Creation Process

Once the data set is loaded and the parameters are setup, it is time to optimize the parameters {threshold (*T*), *k*-nearest neighbours (*K*), feature masks (*F*)}, and simultaneously train the WKNN and WWKNN classifiers by clicking the "Start" button. The system operates until one or more termination criteria are met.

Step 4: Results Collection

Once the system meets the termination criteria, the overall classification accuracy, and the optimal number of threshold (*T*), *K*-value (*K*) and feature subset (*F*) ("0" for rejected features and "1" for selected features) are returned into the main GUI screen. The overall accuracy is calculated as:

$$OvellAccuracy = \frac{Class1Accuracy + Class2Accuracy + ...ClassNAccuracy}{numClass * 100} \tag{5.1}$$

where the accuracy of each class is calculated as:

$$Accuracy = \frac{\text{accurate classification } + \text{inaccurate classification}}{numSample * 100} \tag{5.2}$$

Once the optimal number of features is selected, the feature subset can be saved by clicking the "Save Feature(s)" button.

As the overall classification accuracy returned to the main GUI screen is only based on the training set (90%), the optimal number of {*T*, *K*, *F*} should be further adopted to calculate the final output by using WKNN and WWKNN algorithms that are applied to the testing set (10%). The main reason for this is to avoid a biased verification process for all of the experiments.

Step 5: Clear Screen

Users can click the "Cancel" button to clear the screen and restart the new model creation process.

2) Creating Personalised Prediction Models

Step 1: Load Data Set

The same steps are involved to load the data set into GAPM as in the previous section.

Step 2: Find Nearest Neighbours

A personalised prediction model for an individual new input vector is created based on its nearest neighbours. To do this, users click the "Find Nearest Neighbours" button to investigate the nearest vectors to a target individual vector. As shown in Figure 5.5, first of all, users need to initialize the number of *k*-nearest neighbours for the target vector and enter the index number of the target vector. After doing that, users can visualise the selected nearest neighbours in the "Comment Window", as well as visualise these selected vectors in a 3-D problem space by clicking the "Visualise Nearest Neighbours" button.

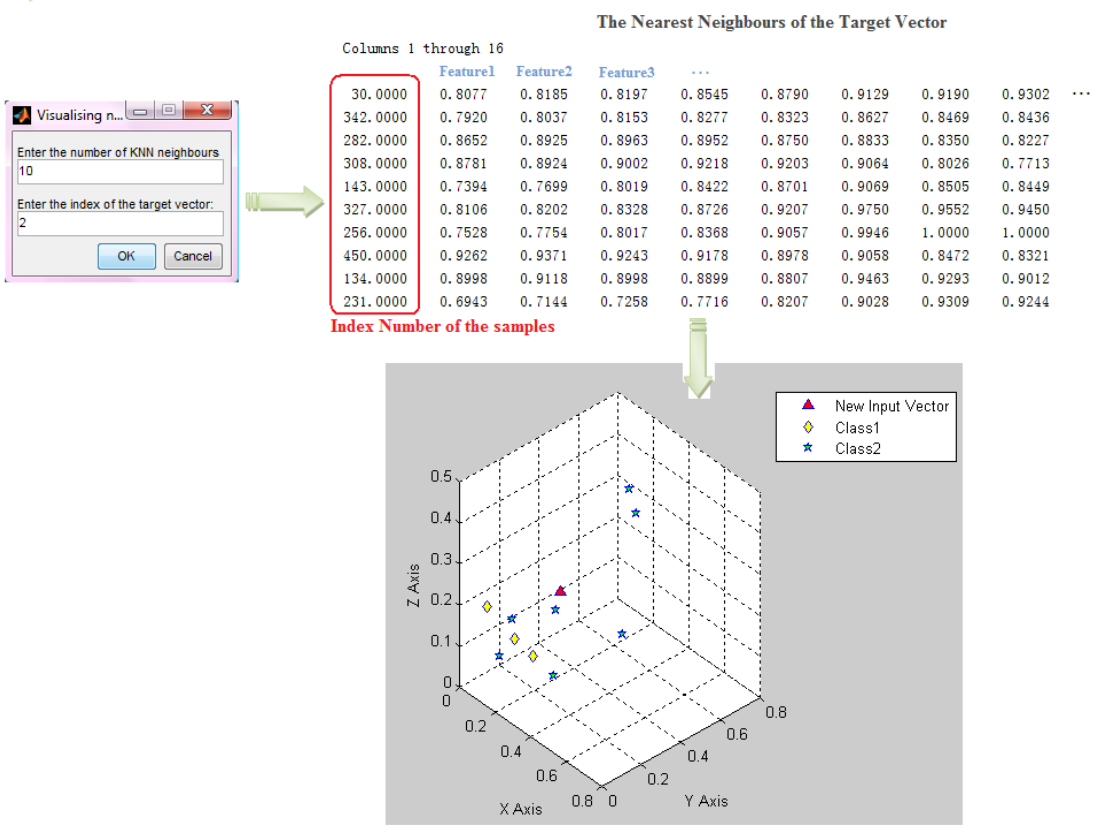


Fig.5.5: The process of finding nearest neighbours for the target vector.

Step 3: Start the Creation Process

Based on the nearest neighbours, a personalised prediction model for an individual target vector can be created by clicking the "Create a Prediction Model" button.

Step 4: Results Collection

Once the personalised prediction model is created, the final output of the target vector is returned in the "Command Window" (see Figure 5.6).

```
======================== WKNN Prediction Model Result ========================

Ihreshold = 0.50   No. of Neighbors = 20

Selected Feature(s) = 40, 47, 48,
Best Accuracy = 90.67 %

     --------- Output for the target vector ---------

Actual Class = 0          Predicted Class = 0          Predicted Output = 1.231280e-002
```

Fig.5.6: An example of a prediction output for an individual target vector.

Step 5: Clear Screen

Users click the "Cancel" button to clear the screen and restart the new prediction models creation process.

5.3. Experiment on Sonar Data Set

In this section, a comparative experiment to compare the classification accuracy between the GA-based personalised modelling with global and local modelling approaches is performed on the Sonar benchmark data set.

5.3.1. Data Set

As presented in Table 5.1, the Sonar data set comprises 208 samples that are described as signals obtained from a variety of different aspect angles. Each sample is a set of 60 attributes in the range 0.0 to 1.0. Each attribute represents the energy within a particular frequency band, integrated over a certain period of time. The class label associated with each signal is either 1 representing the object's signals are recorded as a "rock" or class label 2 if recorded as "mine".

Table 5.1: Summary of Sonar data set used for experimentation.

Data Set Name	Class 1 vs. Class 2	# of Attributes	# of Samples (class 1 / 2)
Sonar	Rock vs. Mine	60	(111 / 97) 208

5.3.1.1. Data Pre-Processing

Firstly, the linear normalisation technique, which is also called the min-max normalisation approach is applied to normalise the data. The linear normalization refers to the fact that all normalized scores are in the range of [0, 1], with the minimal score mapped to 0 and the maximal score to 1. Theoretically speaking, it can be formulated by the following equation (Wu, Crestani, & Bi, 2001):

Normalised Value = $V_i - min(V_i) / max(V_i) - min(V_i)$ (5.3)

where V indicates the feature, $min(V_i)$ is the minimum value of V, while $max(V_i)$ is the maximum value of V. Secondly, the entire data set is split into 90% for training and 10% for testing by using an interleave data splitting method. The training set (90%) is then loaded into the novel GAPM system, the system randomly selects 70% of the data to be used for training and 30% for testing to train the classifiers. In contrast, the testing set (10%) is used to evaluate the final classification accuracy by using WKNN and WWKNN algorithms. Finally, features are selected before investigating the classification accuracy of the global and local modelling approaches by using the SNR feature selection method in NeuCom.

5.3.2. Experimental Setup

5.3.2.1. Software

NeuCom and the novel GA-based personalised modelling (GAPM) system are the software used in this experiment. NeuCom is used to evaluate the classification accuracy of the global and local modelling approaches. In contrast, the novel GAPM system is used to calculate the classification accuracy of the personalised modelling approach.

5.3.2.2. Experimental Method

Unbiased verification method is employed in both feature selection and classification stages. The classification accuracies of global, local and personalised modelling approaches are all calculated by using the Leave-One-Out Cross-Validation (LOOCV) method.

Step 1: Create a global model based on an inductive approach using the SVM algorithm in NeuCom.

Step 2: Create a local model based on an inductive approach using the ECF algorithm in NeuCom.

Step 3: Create personalised classification models based on a transductive approach by running GAPM with GA–optimized WKNN and WWKNN algorithms.

Once all models are created, these four classification models are then compared on the basis of their classification accuracy.

5.3.3. Results and Analysis

As demonstrated in Table 5.2, the SVM, ECF, WKNN and WWKNN classification models are investigated in this study. For both GA-optimized WKNN and WWKNN algorithms, 15 populations are used and run for 20 generations, where the cross-validation method used is 10-fold cross-validation. The accuracy achieved by the different models is presented in Table 5.2, where *K* is the number of nearest neighbours used in both WKNN and WWKNN algorithms. The GA-optimized WWKNN algorithm achieves the best overall classification accuracy at 81.89% (80.54% for Class 1 and 83.23% for Class 2) when compared with other three algorithms. This accuracy is achieved when the value of *K* is 18 and 33 features are selected. The GA-optimized WKNN algorithm provides its best accuracy at 79.88% (79.32% for Class 1 and 80.45% for Class 2) when the value of *K* is 15 and 32 features are selected. In contrast, the classification accuracy achieved by the SVM global model and the ECF local model is 76.19% (66.67% for Class 1 and 83.33% for Class 2) and 67.83% (65.65% for Class 1 and 70.00% for Class 2), respectively. As a result, it can be seen that the GA-optimized WKNN and WWKNN algorithms provide better results when compared with the global modelling and local modelling approaches.

Table 5.2: Experimental results of Sonar data set in terms of model classification accuracy tested using SVM, ECF, WKNN and WWKNN models.

		Global	Local	Personalised	
Model		**NeuCom**	**NeuCom**	**GAPM**	**GAPM**
		Inductive		**Transductive**	
		SVM	**ECF**	**WKNN (k=15)**	**WWKNN (k=18)**
Number of Selected Features		**15**	**20**	**32**	**33**
Accuracy of Each Class (%)	**Class1**	**66.67**	**65.65**	**79.32**	**80.54**
	Class2	**83.33**	**70.00**	**80.45**	**83.23**
Overall Accuracy (%)		**76.19**	**67.83**	**79.88**	**81.89**

5.3.3.1. Knowledge Discovery

The reasons behind the improved classification accuracy using GA-optimized WKNN and WWKNN algorithms are:

Using GAPM to Select an Optimal Number of KNN

The GAPM automatically selects an optimal number of *K* values for each new input vector based on its nearest neighbours instead of manually selecting a *K* value. In GAPM, the *K* value ranges from one to the maximum size of the sample in a problem space.

Using GAPM to Select an Optimal Set of Important Features

The process of feature selection is another important reason for improved classification performance. In the case of both GA-optimized WKNN and WWKNN algorithms, the features are automatically selected using the GAPM to verify the correct range of features. The frequency of feature selection is calculated using the GA-optimized WKNN and WWKNN algorithms as shown in Figures 5.7 and 5.8, respectively. The frequency of feature selection is calculated by executing the system 100 times with fixed GA parameter optimization and cross-validation times (10-fold).

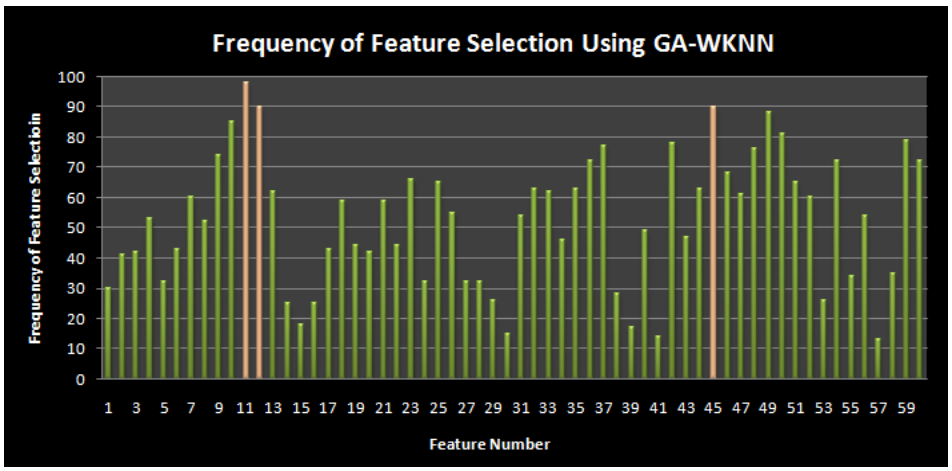


Fig.5.7: The frequency of feature selection as calculated using the GA-optimized WKNN algorithm (Sonar data set).

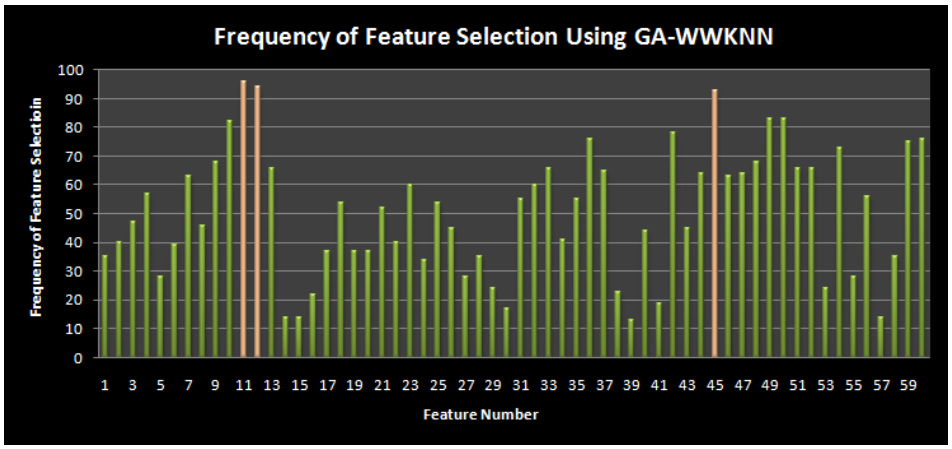


Fig.5.8: The frequency of feature selection as calculated using the GA-optimized WWKNN algorithm (Sonar data set).

As presented in Figures 5.7 and 5.8, the attributes "**11**", "**12**", and "**45**" are the most frequently selected features using both GA-optimized WKNN and WWKNN algorithms after executing the GAPM system 100 times. The justification for executing the GAPM system 100 times is to investigate whether the order of the selected features remains the same, if there is an increase in the selection frequency of each feature. However, since there is no significant increase in selection frequency, the order of the selected features remains the same. Based on this hypothesis, it is assumed that the order of the selected features will not be affected by increasing the number of GAPM executions.

5.4. Summary

This chapter began with a presentation of the MATLAB implementation of the novel GAPM system. In addition, an experiment was run on the Sonar benchmark data set to compare classification accuracy between the GA-based personalised modelling approach against global and local modelling approaches. The results proved that the GA-optimized WKNN and WWKNN algorithms provide better accuracy than the global and local modelling algorithms. The reason being the GAPM system automatically optimises the number of nearest neighbours and features. It was discovered that the attributes "**11**", "**12**", and "**45**" were the most frequently selected features using both GA-optimized WKNN and WWKNN algorithms after executing the GAPM system 100 times.

Chapter 6

Comparative Analysis of GAPM versus Global and Local Modelling Using Leukaemia Cancer Data Set: A Case Study

6.1. Introduction

In Chapter 4, a novel GA-optimized framework and system for personalised modelling called GAPM was introduced which contains two base personalised algorithms: WKNN and WWKNN. In this chapter, a detailed comparative analysis of global and local modelling approaches against personalised modelling approach is presented using NeuCom and the novel GAPM system on the leukaemia cancer data set. SVM is the global algorithm selected for the comparison with the personalised modelling approaches, whereas ECF is the local algorithm selected for comparative analysis with personalised modelling.

Firstly, the experiments begin with a problem specification section which introduces the basic concept of leukaemia cancer and the reasons for studying a data set related to this area. Secondly, a description of the data set and the data pre-processing stages are presented. This is followed by the experimental setup section which introduces the two pieces of software used in this study, as well as all the steps undertaken in the experiments with the methodology of each step. Finally, the experimental results with detailed analysis are presented.

6.2. Problem Specification

In this study, a novel GA-based system for personalised modelling was developed that allows users to select and optimise the most important features and nearest neighbours for a single sample in relation to a certain problem based on a weighted variable distance measure in order to provide more precise accuracy and personalised knowledge when compared with global modelling and local modelling approaches. As a result, the principle goal of this empirical study is to compare classification accuracy of global, local and personalised modelling approaches as well as to investigate their performance on the leukaemia cancer data set. According to Mitchell (1997), classification accuracy is a common performance metric in machine learning and is widely applied to investigate the performances of classifiers. Appropriate measures of classification accuracy are able to provide us with a measure of classification performance. The most

common tool utilized for defining classification accuracy is a confusion matrix. A confusion matrix presents the number of correct and incorrect predicted classifications made by the model compared with the actual classifications in the testing data. The matrix is *n* x *n*, where *n* is the number of classes. Figure 6.1 illustrates a confusion matrix for a binary classification problem: the rows represent the instances in an actual class, while the columns represent the the instances in a predicted class.

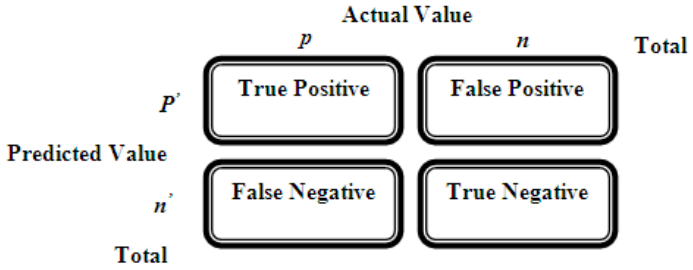


Fig.6.1: An overview of table of confusion.

Cancer is a complex disease of the cells in the body, which arises from a variety of genome-based abnormalities. For instance, leukaemia is one of the harmful cancer of a subset of white blood cells. As mentioned by Dockerty (2008), leukaemia is the commonest cancer in New Zealand, with a significant increase in the incidence rate among children aged from 0-14 (4.89/100,000 person/year in 1953-57 to 7.92/100,000 person/year in 1988-90). Thus, the Leukaemia & Blood Foundation and the National Cancer Registry in New Zealand has commissioned several quality epidemiological studies, specifically on patients affected by leukaemia and related blood conditions. As mentioned above, based on previous studies, the purpose of this experimental study is to investigate and compare classification accuracy by using different software and modelling techniques on the leukaemia cancer data set in order to facilitate new knowledge discovery to help developing more innovative and effective therapeutic treatments and diagnoses for leukaemia cancer.

6.3. Data Set

The biological problem on which all experiments were undertaken was to distinguish two types of Leukaemia: *Acute Lymphoblastic Leukaemia* (ALL) and *Acute Myeloid Leukaemia* (AML). As presented in Table 6.1, the entire data is classified into to two datasets: (1) the training data set contains 38 bone marrow samples (27 ALL patients and 11 AML patients), obtained from acute leukaemia patients at the time of diagnosis, and (2) the testing data contains 34 bone marrow samples (20 ALL patients and 14 AML patients). For each patient, the data consists of 7,129 gene expressions.

Table 6.1: Summary of leukaemia data set used for experimentation.

Data Set	Class 1 vs. Class 2	# of Genes	Training Samples (class 1 / 2)	Testing Samples (class 1 / 2)
Leukaemia	ALL vs. AML	7129	(27 / 11) 38	(20 / 14) 34

6.3.1. Data Pre-Processing

The leukaemia cancer data set does not contain any missing values. However, it is a very high-dimensional data set. In addition, the data set is divided into two data sets. Therefore, it is essential to pre-process the data before running the experiments.

There are four major steps involved in pre-processing the data. The first is to combine the training and testing data sets into one data set. The second is to normalise the data. A linear normalisation technique is applied to normalise the data (there is a detailed explanation in section 5.3.1.1.). In the third step, the entire data set is split into 90% for training and 10% for testing using an interleave data splitting method. The training set (90%) is then loaded into the novel GAPM system, and the system randomly selects 70% of the data to be used for training and 30% for testing to train the classifiers. The testing set (10%) is used to investigate the final output using WKNN and WWKNN algorithms. Finally, features are selected before investigating classification accuracy of the global and local modelling approaches using the SNR feature selection method in NeuCom.

6.4. Experimental Setup

6.4.1. Software

NeuCom and the novel GAPM system are the pieces of software utilized in this study. NeuCom is used to evaluate the classification accuracy of the global and local modelling approaches. In contrast, the novel GAPM system is used to calculate the classification accuracy of the personalised modelling approach.

6.4.2. Experimental Method

Unbiased verification method is employed in both feature selection and classification stages. The classification accuracies of the global, local and personalised modelling approaches are all calculated using the Leave-One-Out

Cross-Validation (LOOCV) method:

Step 1: Create a global model based on an inductive approach using the SVM algorithm in NeuCom.

Step 2: Create a local model based on an inductive approach using the ECF algorithm in NeuCom.

Step 3: Create personalised classification models based on a transductive approach by running GAPM with GA–optimized WKNN and WWKNN algorithms.

Once all models are created, these four classification models are then compared on the basis of their classification accuracy.

6.5. Results and Analysis

As shown in Table 6.2, the SVM, ECF, WKNN and WWKNN classification models are investigated in this study. For both the GA-optimized WKNN and WWKNN algorithms, 20 populations are used and run for 25 generations, and the cross-validation method used is 10-fold cross-validation. The accuracy achieved by the different models is presented in Table 6.2, where *K* represents the number of nearest neighbours used in both the WKNN and WWKNN algorithms. The GA-optimized WKNN algorithm achieves the best overall classification accuracy at 95.10% (95.67% for Class 1 and 94.53% for Class 2) when compared with other three algorithms. This accuracy is achieved when the value of *K* is 10 and 32 features are selected. On the other hand, the GA-optimized WWKNN algorithm achieves its best accuracy at 93.18% (94.52% for Class 1 and 91.85% for Class 2) when the value of *K* is 8 and 33 features are selected. In contrast, the classification accuracy achieved by the global SVM and local ECF algorithms are 90.70% (91.74% for Class 1 and 89.65% for Class 2) and 91.12% (92.53% for Class 1 and 89.71% for Class 2), respectively. It can be seen that the GA-optimized WKNN and WWKNN algorithms provide better results when compared with the global modelling and local modelling approaches.

Table 6.2: Results of leukaemia cancer data set in terms of model classification accuracy tested using SVM, ECF, WKNN and WWKNN models.

		Global	Local	Personalised	
Model		**NeuCom**	**NeuCom**	**GAPM**	**GAPM**
		Inductive		**Transductive**	
		SVM	**ECF**	**WKNN (k = 10)**	**WWKNN (k = 8)**
Number of Selected Features		**15**	**30**	**32**	**33**
Accuracy of Each Class (%)	**Class1**	**91.74**	**92.53**	**95.67**	**94.52**
	Class2	**89.65**	**89.71**	**94.53**	**91.85**
Overall Accuracy (%)		**90.70**	**91.12**	**95.10**	**93.18**

6.5.1. Knowledge Discovery

The results calculated by these four algorithms are all acceptable for real clinical problem of disease diagnosis. The GA-optimized WKNN and WWKNN algorithms provide better results when compared with the SVM and ECF algorithms. The reasons for improved classification accuracy are:

Using GAPM to Select an Optimal Number of KNN

The GAPM automatically selects an optimal number of *K* values for each new input vector based on its nearest neighbours instead of manually selecting a *K* value. In GAPM, the *K* value ranges from one to the maximum size of the sample in a problem space.

Using GAPM to Select an Optimal Set of Important Features

The process of feature selection is another very important reason for improved classification performance. In the case of both GA-optimized WKNN and WWKNN algorithms, the features are automatically selected using GAPM to ensure the correct range of features that have an effect on prediction.

As the leukaemia cancer data consists of 7,129 genes, it is difficult to investigate the frequency of each selected feature as it would mean executing the GAPM system over a hundred times. In Raphael Hu's study (2006), the experimental results showed that the best overall classification result on the testing set was

94.12%, when 35 genes were selected for constructing the final optimized classifier (see Figure 6.2). It can be seen that classification accuracy is slightly improved by using the novel GAPM system with fewer genes selected when compared with Raphael Hu's study.

Microarray dataset	Number of selected genes	TP	TN	FP	FN	Classification accuracy
Leukaemia data	35	12	20	0	2	94.12%

Fig.6.2: The classification result from the leukaemia cancer data set using GAGSc method (Hu, 2006).

6.6. Predicting an Individual Patient's Cancer Type

6.6.1. Experimental Setup

In this empirical study, two patients are studied: one is an "ALL" patient (e.g. Sample 1 – class label 1), while the other is an "AML" patient (e.g. Sample 22 – class label 2). As stated previously, there has been a significant increase in the leukaemia incidence rate among children aged 0-14 (4.89/100,000 person/year in 1953-57 to 7.92/100,000 person/year in 1988-90) (Dockerty, 2008). Thus, it is important to make further research in the area of leukaemia cancer in order to encourage new knowledge discovery, as well as help develop innovative and effective therapeutic treatments and diagnoses focused on leukaemia cancer.

6.6.2. Results and Analysis

Example 1: "ALL" patient (Sample 1 – class label 1)

As mentioned above, the first step in predicting an individual patient is to define its nearest neighbours. Based on its nearest neighbours, the test vector is investigated using the WKNN and WWKNN prediction models, which are described as follows:

1) Using the WKNN Prediction Model

Figure 6.3 shows that features "**2365**", "**4849**", and "**690**" are selected as being the most significant features for predicting the test vector. Furthermore, based on its 15 nearest neighbours, the output of the test vector is predicted as "**1**" which accurately matches the actual output class label.

```
======================== WKNN Prediction Model Result ========================

Threshold = 0.50    No. of Neighbors = 15

Selected Feature(s) = 2365, 4849, 690,
Best Accuracy = 86.67%

     --------- Output for the target vector ---------

Actual Class = 1           Predicted Class = 1           Predicted Output = 1.121387e+000
```

Fig.6.3: Output of "ALL" sample predicted using the WKNN prediction model.

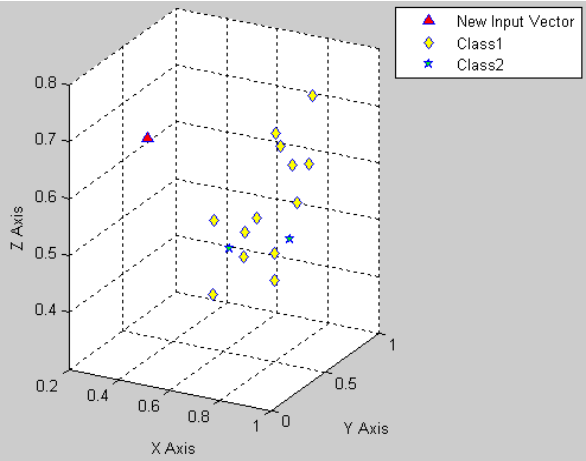


Fig.6.4: Overview of the nearest neighbours of "ALL" sample using the WKNN algorithm.

2) Using the WWKNN Prediction Model

Figure 6.5 shows that features "**1831**", "**3254**", "**2365**", and "**4952**" are selected as being the most significant features for predicting the test vector. Moreover, based on its 25 nearest neighbours, the output of the test vector is predicted as "**1**" which accurately matches the actual output class label.

```
======================== WWKNN Prediction Model Result ========================

Threshold = 0.50    No. of Neighbors = 25

Selected Feature(s) = 1831, 3254, 2365, 4952,
The Weight of Selected Feature(s) = 1, 0.78866, 0.78345, 0.71341,
Best Accuracy = 92.00%

     --------- Output for the target vector ---------

Actual Class = 1           Predicted Class = 1           Predicted Output = 1.074155e+000
```

Fig.6.5: Output of "ALL" sample predicted using the WWKNN prediction model.

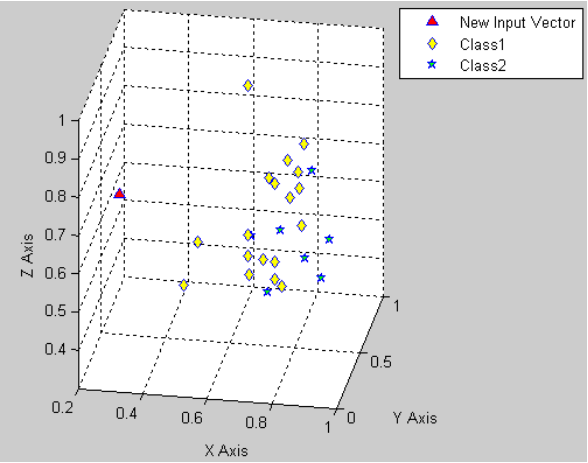


Fig.6.6: Overview of the nearest neighbours of "ALL" sample using the WWKNN algorithm.

Knowledge Discovery

As observed in Figures 6.3 and 6.5, both WKNN and WWKNN prediction models give an accurate prediction for an individual "ALL" patient. As mentioned in Chapter 2, the basic idea behind the WWKNN algorithm is the output of each new input vector is not only dependent upon the distance between the existing vectors and the new input vector, but it is also upon the power of each vector as weighted according to their importance within the sub-space to which the new input vector belongs. Figure 6.5 shows the weight of each selected feature.

A further investigation on the effects of different threshold settings on the accuracy is made. The number of nearest neighbours is maintained but the accuracy based performance evaluation is carried out using different threshold values ranging from a minimum value of 0.1 to the maximum value of 1.

Figure 6.7 shows the influence of different threshold settings on the classification accuracy obtained using the WKNN algorithm. Initially, the accuracy obtained is 83.33% for threshold values ranging from 0.1 to 0.3, but the accuracy drops down to 80.43% when the threshold value is 0.4. The accuracy significantly increases to 86.67% for threshold values ranging from 0.5 to 0.7. Finally, the accuracy drops to 83.33% again for threshold values ranging from 0.8 to 1.

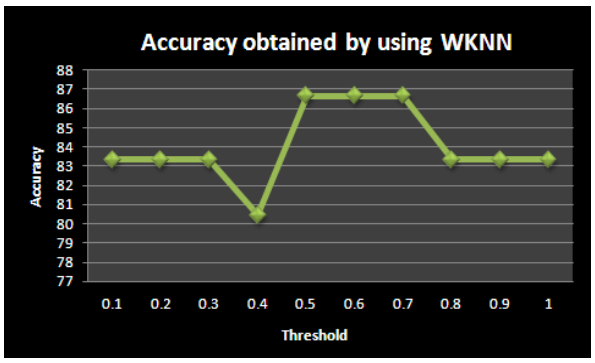


Fig.6.7: The threshold settings effect on the accuracy of "ALL" sample obtained using the WKNN prediction model.

Figure 6.8 show the influence of different threshold settings on the classification accuracy obtained using the WWKNN algorithm. Initially, the accuracy obtained is 89.00% for threshold values ranging from 0.1 to 0.3, but the accuracy decreases to 86.33% when the threshold value is 0.4. Surprisingly, the accuracy increases to 92.00% for threshold values ranging from 0.5 to 0.6. The accuracy drops down to 86.33% again for threshold values ranging from 0.7 to 0.8. The accuracy jumps to 89.00% again for threshold values ranging from 0.9 to 1.

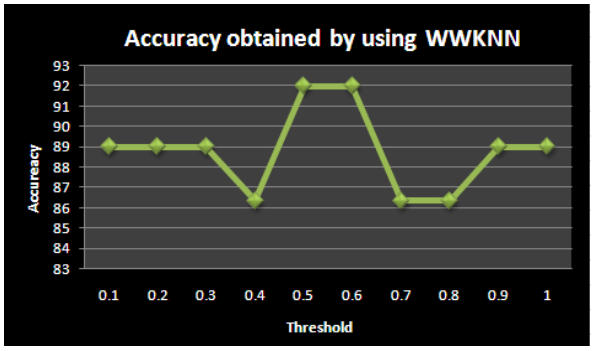


Fig.6.8: The threshold settings effect on the accuracy of "ALL" sample obtained using the WWKNN prediction model.

As shown in Figures 6.7 and 6.8, threshold values ranging from 0.5 to 0.6 provide the highest accuracy when using either algorithm.

Example 2: "AML" patient (Sample 22 – class label 2)

The experiment begins with the definition of the nearest neighbours for the test vector. Based on its nearest neighbours, the test vector is investigated using the WKNN and WWKNN prediction models, which are described as follows:

1) Using the WKNN Prediction Model

As shown in Figure 6.9, the test vector is predicted as "**2**" which precisely matches the actual output class label, based on its 28 nearest neighours. The features"**3254**", "**2290**", and "**2365**" are selected as the most important features

for predicting the test vector.

```
======================== WKNN Prediction Model Result ========================

Threshold = 0.50    No. of Neighbors = 28

Selected Feature(s) = 3254, 2290, 2365,
Best Accuracy = 81.43%

      --------- Output for the target vector ---------

Actual Class = 2            Predicted Class = 2            Predicted Output = 1.546793e+000
```

Fig.6.9: Output of "AML" sample predicted using the WKNN prediction model.

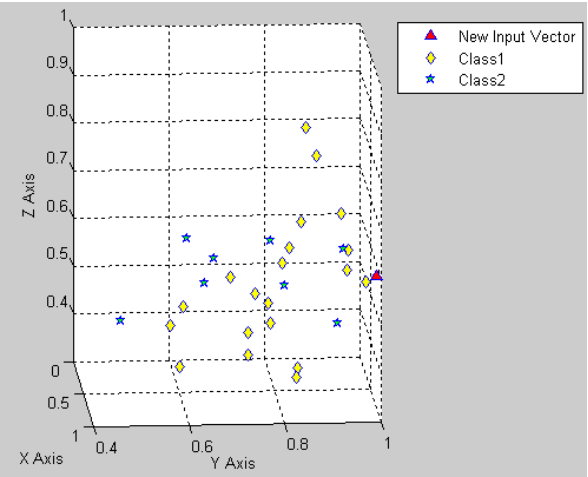


Fig.6.10: Overview of the nearest neighbours of "AML" sample using the WKNN algorithm.

2) Using the WWKNN Prediction Model

As demonstrated in Figure 6.11, the test vector is predicted as "**2**" which precisely matches its actual output class label, based on its 32 nearest neighours. The features"**3254**", "**2290**", "**6283**" and "**1830**" are selected as the most important features for predicting the test vector.

```
======================== WWKNN Prediction Model Result ========================

Threshold = 0.50    No. of Neighbors = 32

Selected Feature(s) = 3254, 2290, 6283, 1830,
The Weight of Selected Feature(s) = 1, 0.56069, 0.40097, 0.376195,
Best Accuracy = 90.88%

      --------- Output for the target vector ---------

Actual Class = 2            Predicted Class = 2            Predicted Output = 1.741460e+000
```

Fig.6.11: Output of "AML" sample predicted using the WWKNN prediction model.

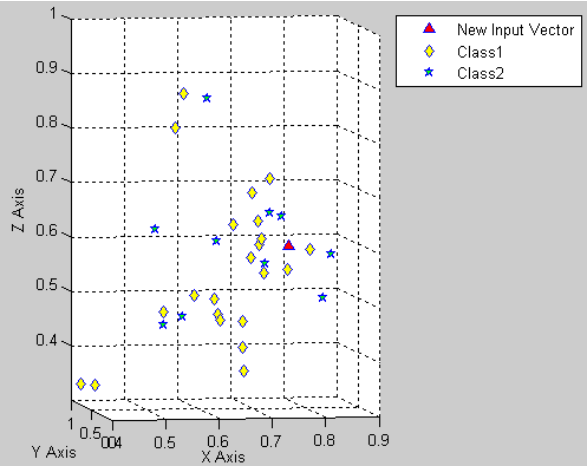


Fig.6.12: Overview of the nearest neighbours of "AML" sample using the WWKNN algorithm.

Knowledge Discovery

As observed in Figures 6.9 and 6.11, both WKNN and WWKNN prediction models give an accurate prediction for an individual "AML" patient. Figure 6.11 also shows the weight of each selected feature. As because in the WWKNN algorithm, the output of each new input vector is dependent upon the distance between the existing vectors and the new input vector, as well as the power of each vector weighted according to their importance within the local space to which the new input vector belongs.

The effects of different threshold settings on the overall accuracy are further investigated, the number of nearest neighbours is maintained but the classification accuracy based performance evaluation is carried out using different threshold values ranging from a minimum value of 0.1 to the maximum value of 1.

Figure 6.13 shows the influence of different threshold settings on the classification accuracy obtained using the WKNN algorithm. Initially, the accuracy achieved is 68.57% for threshold values ranging from 0.1 to 0.2, and this increases to 71.43% for threshold values ranging from 0.3 to 0.4. The accuracy significantly increased to 81.43% for threshold values ranging from 0.5 to 0.8. However, the accuracy drops down to 71.43% again for threshold values ranging from 0.9 to 1.

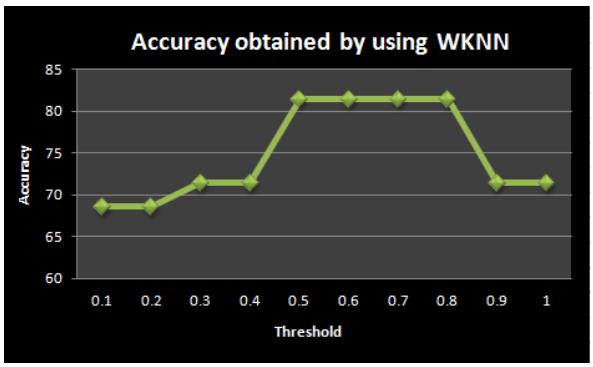


Fig.6.13: The threshold settings effect on the accuracy of "AML" sample obtained using the WKNN prediction model.

Figure 6.14 shows the influence of different threshold settings on the classification accuracy obtained using the WWKNN algorithm. Initially, the accuracy achieved is 85.26% for threshold values ranging from 0.1 to 0.2, and this jumps to 89.00% for threshold values ranging from 0.3 to 0.4. The accuracy slightly increases to 90.88% for threshold values ranging from 0.5 to 0.8. However, the accuracy decreases to 89.00% again for threshold values ranging from 0.9 to 1.

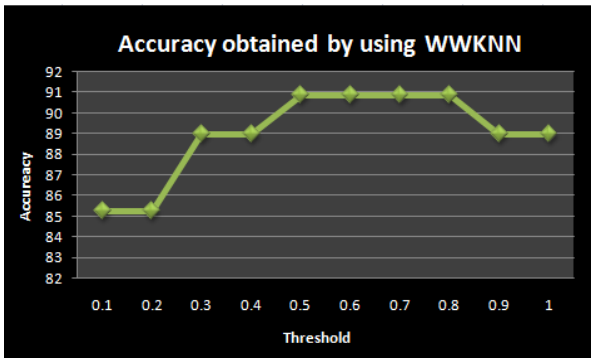


Fig.6.14: The threshold settings effect on the accuracy of "AML" sample obtained using the WWKNN prediction model.

As shown in Figures 6.13 and 6.14, the threshold values ranging from 0.5 to 0.8 provide the highest accuracy when using either algorithm.

The output for both "ALL" and "AML" patients are accurate using the WKNN and WWKNN prediction models. Two major reasons for this are:

- ✧ Using the *k*-nearest neighbour (KNN) algorithm with Euclidean distance measure to estimate the similarities between the test vector and its nearest neighbours. The KNN algorithm estimates values of a potential model for only a single point (new input vector) of the problem space by using additional information related to that point (the nearest neighbours of the

test point).

- ✧ Based on the nearest neighbours, the system automatically selects an optimal set of features to verify the correct range of features that effect diagnosis for an individual patient.

6.7. Summary

In this chapter, a detailed comparative analysis of GA-optimized WKNN and WWKNN personalised models and global SVM and local ECF models was performed on a leukaemia cancer data set. The experimental results proved that the personalised modelling approach provided better classification accuracy when compared with the global and local modelling approaches. In addition, this chapter also presented a detailed experimental study on predication for an individual patient using the personalised prediction models. The results proved that both WKNN and WWKNN prediction models gave an accurate prediction for an individual "ALL" and "AML" patient. Furthermore, the effects of different threshold settings on overall accuracy were investigated using the WKNN and WWKNN algorithms. The results proved that accuracy varies when using different threshold settings, and the best accuracy was obtained with threshold values ranging from 0.5 to 0.6.

Chapter 7

Comparative Analysis of GAPM versus Global and Local Modelling Using Pest-Related Climate Data Set: A Case Study

7.1. Introduction

In Chapter 6, a detailed comparative analysis of GA-based personalised modelling against global and local modelling was presented on the leukaemia cancer data set. In this chapter, a performance evaluation of the global and local modelling approaches against the GA-based personalised modelling approach is conducted on a real world pest-related climate data set which contains information on pest establishment in numerous regions of the world. As stated in the previous experimental study, SVM is the global algorithm selected for comparison with the personalised modelling approaches, and ECF is the local algorithm.

The problem specification section begins the experiment outlining the need for different modelling approaches to study the potential of pest establishment in several regions of the world. Secondly, a description of the data set along with the motivation behind selection of this data set, as well as the data pre-processing stages is presented. Thirdly, the experimental setup section introduces the two pieces of software used in this study, along with all of the steps included in the experiments with a methodology of each step. Finally, detailed experimental results are presented and discussed.

7.2. Problem Specification

Jones and Kitching (1981) define "pest" as an organism, which has the potential to destroy products, damage crops, cause or transmit diseases, or have serious impact on flora and fauna. Most pest species arrive into an area either on purpose or accidentally, and reproduce quickly until they occupy large areas, thus badly impacting local, native ecosystems (Sailer, 1983; Pimentel, 1986; Worner, 1994). Therefore, pest invasions have been seen as a major cause of environmental and economic harm. Horticultural and agricultural products provide important earnings for New Zealand. A New Zealand government annual report states that the direct economic expense caused by pests is approximately 880 million NZD per year, including the cost of eradication control programmes and losses of agricultural products (Barlow & Goldson, 2002).

One of the best methods of minimizing the impact of pest invasions is to control their establishment. In 2002, Stynes suggested that an appropriate way to solve this problem is to predict the probability of a pests' establishment when they arrive into a new area. However, this is not a comprehensive solution for the problem. It is better to find the factors that have an impact on the pests' establishment in a certain area. Generally, a pests' establishment in a new area primarily dependent upon both *biotic* and *abiotic* factors, such as climate and specific environmental conditions (Mooney & Drake 1989). Recently, a variety of computer-based techniques have been applied to monitor and control these factors in order to prevent pest establishment. These techniques include multiple linear regression (MLR), correlation analysis, artificial neural networks (ANN) (Brosse et al., 1999). Kasabov, Pang, Soltic, Worner and Peackok (2004) extracted rules from data related to pest establishment in different regions that were integrated using local models based on data with similar characteristics instead of using global models from the data of several regions. Their experimental results prove that local models provide better accuracy than global models. However, there is a growing interest in using personalised models to focus on an individual species of pest. Because a personalised model is created for every single new input vector of the problem space, based on its nearest neighbours, more precise results than both global and local models are provided.

As mentioned above, in this experimental study, I compare the global, local and personalised modelling approaches in order to prove that personalised modelling is more suitable for monitoring, controlling and investigating pests' establishment prognosis.

7.3. Data Set

This pest-related climate data set is extracted with permission of the Crop Protection Compendium which contains a wide range of information about all aspects of crop protection (e.g. pests, crops, and diseases, etc.) associated with most of the geographical areas of the world. As presented in Figure 7.1, the entire data set comprises 844 pest species (followed by an output class label: 0 and 1 indicating the absence or presence of each species in each geographical area) for 459 geographical areas.

Code	Species name
ACACS1	Acanthocoris scabrator
ACAIHE	Acanthiophilus helianthi
ACAMTO	Clavigralla tomentosicollis
ACANOB	Acanthoscelides obtectus
ACAYVT	Acalymma vittatum
ACHELA	Acherontia lachesis
ACHEST	Acherontia styx
ACLRGL	Acleris gloverana
ACRAAC	Acraea acerata
ACRICI	Acrida cinerea
ACRNRU	Acronicta rumicis
...	...

	Order	Species number
1	Lepidoptera	257
2	Hemiptera	228
3	Coleoptera	203
4	Diptera	83
5	Thysanoptera	39
6	Orthoptera	16
7	Hymenoptera	11
8	Isoptera	3
9	Psocoptera	3
10	Collembola	1
	Total	844

Code	Region name
AD	Andorra
AE	United Arab Emirates
AF	Afghanistan
AG	Antigua and Barbuda
AI	Anguilla
AL	Albania
AM	Armenia
AN	Netherlands Antilles
AO	Angola
AR	Argentina
AS	American Samoa
AT	Austria
...	...

844 x 459

Species	Regions										
	AD	AE	AF	AG	AI	AL	AM	AN	AO	AR	...
ACACS1	0	0	0	0	0	0	0	0	0	0	
ACAIHE	0	0	1	0	0	1	0	0	0	0	
ACAMTO	0	0	0	0	0	0	0	0	1	0	
ACANOB	0	0	0	0	0	1	1	0	1	1	
ACAYVT	0	0	0	0	0	0	0	0	0	0	
ACHELA	0	0	0	0	0	0	0	0	0	0	
ACHEST	0	1	0	0	0	0	0	0	0	0	
ACLRGL	0	0	0	0	0	0	0	0	0	0	
ACRAAC	0	0	0	0	0	0	0	0	0	0	
ACRICI	0	0	0	0	0	0	0	0	0	0	...
ACRNRU	1	0	1	0	0	1	1	0	0	0	
ACROAS	0	0	0	0	0	0	0	0	0	0	
ACYRKO	0	0	1	0	0	0	0	0	0	1	
ACYRON	0	0	1	0	0	1	0	0	0	1	
ADORSI	0	0	0	0	0	0	0	0	0	0	
ADORVE	0	0	0	0	0	0	0	0	0	0	
AEOLSA	0	0	1	0	0	0	0	0	0	0	
AGMYFR	0	0	1	0	0	0	0	0	0	0	
AGMYOR	0	0	0	0	0	0	0	0	0	0	
AGRILI	0	0	0	0	0	0	0	0	0	0	
AGRIOB	0	0	0	0	0	0	0	0	0	0	
...					...						...

Fig.7.1: Overview of all pest species represented in the data set.

Due to the time and size limitations of this study, it is too difficult to investigate all pest species, and so only one species *CERTCA- ceratitis capitata* is studied. As shown in Table 7.1, the data set used for the experiments contained 459 samples (356 absence of species (Class1) and 103 presence of species (Class 2)) with 69 features (as described in Figure 7.2). In this study, all of the 69 features in the problem space are used to perform a comparative analysis of the global, local and personalised modelling approaches.

Table 7.1: Summary of pest-related climate data set used for experimentation.

Data Set Name	Class 1 vs. Class 2	# Genes	# Samples (class 1 / 2)
Pest-Related Climate	Absence vs. Presence	69	(356 / 103) 459

459 x 69

	Climate variables									
	MEAN TJan	MEAN TFeb	...	MEAN TSum1	MEAN TSum2	...	MEAN AD300mm	MEAN AS300 mm	...	Class Lable
Regions										
AD	7.5	8.6		20.3	23.05		295.255108	0		0
AE	17.75	18.9		30.55	32.8		1761.443999	0		0
AF	1.42675	3.32575		26.56325	28.684		760.5642869	0		0
AG	24.859	24.716		27.471	27.766		519.9490918	0		0
AI	25.653	25.642	...	28.377	28.556	...	818.7190007	0	...	0
AL	5.599	6.87183333		21.3955	24.05283333		135.2397871	594.0326473		1
AM	-5.55	-3.35		22.25	25.55		513.9401186	0		0
AN	26.1	26.1		28.1	27.8		1236.643732	0		1
AO	23.0181429	23.1728095		22.91914286	23.01814286		327.4494228	251.0560834		1
AR	21.8492566	21.0390354		20.80811504	21.84925664		229.4710273	84.97330057		1
AS	26.9	26.9		26.7	26.9		1.130692042	1287.324998		0
AT	-3.41995652	-2.1283913		13.89826087	16.03917391		6.270587676	550.3568004		0
AUlh	23.2803044	23.17572		22.35	22.95		370.7830287	246.2294116		1
AUnt	22.95	22.6		29.75238095	29.43690476		0	807.2245032		0
AUnw	29.4369048	28.8261905		21.1	22.42305195		912.4592216	84.68239059		0
AUql	22.423052	22.3522727	...	26.55769231	26.83461538	...	178.5665219	199.401934	...	0
AUsa	26.8346154	26.5115385		20.61036585	22.45487805		403.277293	199.3578905		0
AUta	22.4548781	22.4658537		13.81810345	15.3887931		445.6994759	12.93469092		1
AUvi	15.3887931	15.7948276		17.9415	19.9165		37.466127	465.5533021		0
AUwa	19.9165	20.287		25.54576923	26.83769231		162.5478177	143.7343953		0
AW	26.8376923	26.5680769		26.708	26.212		826.2477725	56.94618661		1
...	...	26.667			...		...	...		...
ZW	22.81736	22.51438		22.8602	22.81736		374.2985803	35.01214284		1

Climate variables	INTERPRETATION
TJan	Temperature (celsius) for month of january
TFeb	Temperature (celsius) for month of february
TMar	Temperature (celsius) for month of march
TApr	Temperature (celsius) for month of april
TMay	Temperature (celsius) for month of May
TJun	Temperature (celsius) for month of june
TJul	Temperature (celsius) for month of july
TAug	Temperature (celsius) for month of August
TSep	Temperature (celsius) for month of sep
TOct	Temperature (celsius) for month of October
TNov	Temperature (celsius) for month of November
TDec	Temperature (celsius) for month of December
TSum1	Temperature (celsius) for the first summer month (Dec for Southern hemp, Jun for Northern hemp)
TSum2	Temperature (celsius) for the second summer month (Jan for Southern hemp, Jul for Northern hemp)
TSum3	Temperature (celsius) for the third summer month (Feb for Southern hemp, Aug for Northern hemp)
TAut1	Temperature (celsius) for the first autumn month (Mar for Southern hemp, Sep for Northern hemp)
...	...

Fig.7.2: Overview of the pest-related climate data set used for experimentation.

The reasons for selecting this data set are described as follows:

1) The data set does not contain any missing or noise values.

2) This data set is selected by reason of instant availability and accessibility.

3) Because the data set contains 459 samples, it is not too small to give a good overview of the nature of the problem in hand and it is not too large to cause problems with respect to time and computational complexity.

7.3.1. Data Pre-Processing

The pest-related climate data set does not contain any missing or noise values. However, it is still necessary to pre-process the data before running the experiments in order to achieve better classification accuracy throughout experiments. Firstly, the linear normalisation technique is applied to normalise the data as was adopted for the previous experiments (introduced in section 5.3.1.1.). Moreover, the entire data set is randomly split into 90% of the data for training and 10% for testing using an interleave data splitting method. The training set (90%) is then loaded into the novel GAPM system, and the system randomly selects 70% of the data for training and 30% for testing to train the

classifiers. In contrast, the testing set (10%) is utilized to investigate the final output using WKNN and WWKNN algorithms. Finally, features are selected before investigating classification accuracy of the global and local modelling approaches by using the SNR feature selection method in NeuCom.

7.4. Experimental Setup

7.4.1. Software

In this experimental study, NeuCom is selected to calculate classification accuracy of the global and local modelling approaches. The novel GAPM system is applied to evaluate classification accuracy of the personalised modelling approach.

7.4.2. Experimental Method

Unbiased verification method is employed in both the feature selection and classification stages. The classification accuracy of global, local and personalised modelling approaches is all calculated using the Leave-One-Out Cross-Validation (LOOCV) method:

Step 1: Create a global model based on an inductive approach using the SVM algorithm in NeuCom.

Step 2: Create a local model based on an inductive approach using the ECF algorithm in NeuCom.

Step 3: Create personalised classification models based on a transductive approach by running GAPM with GA–optimized WKNN and WWKNN algorithms.

Once all four classification models are created, they are then compared on the basis of their classification accuracy.

7.5. Results and Analysis

As demonstrated in Table 7.2, the SVM, ECF, WKNN and WWKNN classification models are investigated in this study. For both the GA-optimized WKNN and WWKNN algorithms, 20 populations are used and run for 30 generations, and the cross-validation method used is 10-fold cross-validation. The accuracy achieved by the various models is presented in Table 7.2, where K is the number of nearest neighbours used in both the WKNN and WWKNN algorithms. The GA-optimized WWKNN algorithm achieves the best overall classification accuracy at 83.64% (88.73% for Class

1 and 78.55% for Class 2) when compared with other three algorithms. This accuracy is achieved when the value of *K* is 17 and 36 features are selected. The GA-optimized WKNN algorithm achieves its best accuracy at 82.15% (86.61% for Class 1 and 77.76% for Class 2) when the value of *K* is 11 and 35 features are selected. In contrast, the classification accuracy achieved by the SVM global model and the ECF local model is 77.18% (79.65% for Class 1 and 74.72% for Class 2) and 79.96% (83.57% for Class 1 and 76.36% for Class 2), respectively. As a result, it can be seen that the GA-optimized WKNN and WWKNN algorithms provide better results when compared with the global and local modelling approaches.

Table 7.2: Results of pest-related climate data set in terms of model classification accuracy tested using SVM, ECF, WKNN and WWKNN models.

		Global	Local	Personalised	
Model		**NeuCom**	**NeuCom**	**GAPM**	**GAPM**
		Inductive		**Transductive**	
		SVM	**ECF**	**WKNN (k = 11)**	**WWKNN (k = 17)**
Number of Selected Features		**12**	**18**	**35**	**36**
Accuracy of Each Class (%)	**Class1**	**79.65**	**83.57**	**86.61**	**88.73**
	Class2	**74.72**	**76.36**	**77.76**	**78.55**
Overall Accuracy (%)		**77.18**	**79.96**	**82.15**	**83.64**

7.5.1. Knowledge Discovery

The reasons for improved classification performance when using both of the GA-optimized WKNN and WWKNN algorithms are:

Using GAPM to Select an Optimal Number of KNN

The GAPM automatically selects an optimal number of *K* values for each new input vector based on its nearest neighbours instead of manually selecting a *K* value. In GAPM, the *K* value ranges from one to the maximum size of the sample in a problem space.

Using GAPM to Select an Optimal Set of Important Features

The process of feature selection is another important reason for improved

classification performance. In the case of both GA-optimized WKNN and WWKNN algorithms, the features are automatically selected using the GA-based system to ensure the correct range of climatic factors that have an effect on determining the potential establishment of pest in several regions of the world. The frequency of feature selection is investigated using GA-optimized WKNN and WWKNN algorithms are shown in Figures 7.3 and 7.4, respectively. The frequency of feature selection is ranked by executing the system 100 times with fixed GA parameter optimization and cross-validation times (10-fold).

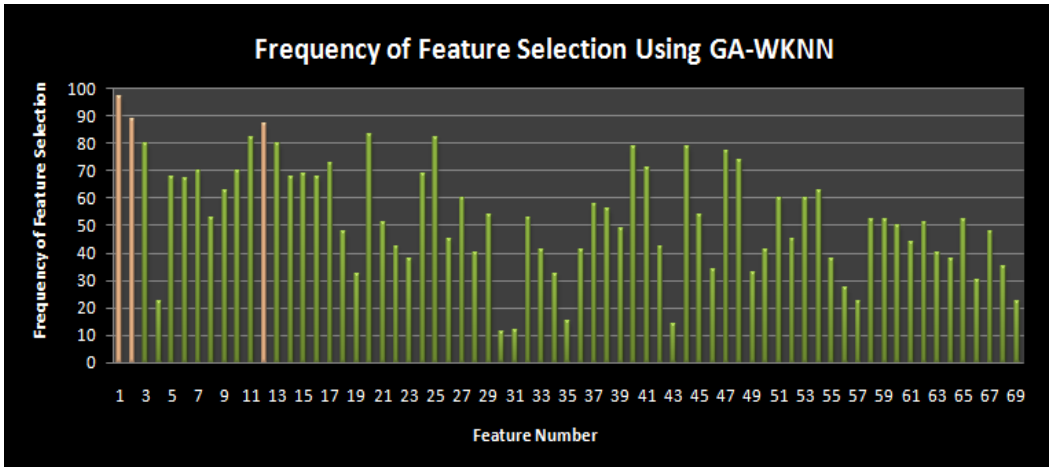


Fig.7.3: The frequency of feature selection using the GA-optimized WKNN algorithm (Pest-related climate data set).

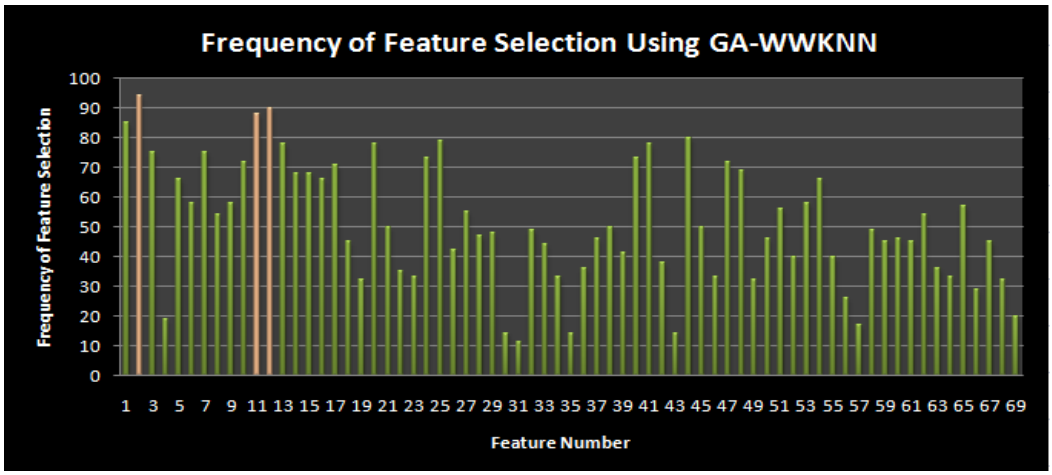


Fig.7.4: The frequency of feature selection using the GA-optimized WWKNN algorithm (Pest-related climate data set).

As presented in Figures 7.3 and 7.4, features "**1**" (Temperature (celsius) for month of January), "**2**" (Temperature (celsius) for month of February), and "**12**" (Temperature (celsius) for month of December) are the top three features selected using GA-optimized WKNN, whereas features "**2**" (Temperature (celsius) for month of February), "**12**" (Temperature (celsius) for month of

December), and "**11**" (Temperature (celsius) for month of November) are the top three features selected using the GA-optimized WWKNN algorithm after executing the GAPM system 100 times. The reason behind executing the GAPM system 100 times is to investigate whether the order of the selected features remains the same, if there is an increase in the selection frequency of each feature. However, since there is no significant increase in the selection frequency, the order of the selected features remains the same. Based on this hypothesis, it is assumed that the order of the selected features will not be affected by increasing the number of GAPM executions.

7.6. Predicting the Establishment of an Individual Pest Species

7.6.1. Experimental Setup

Due to time and size limitations of this study, only one sample / region is studied in this experiment. The sample "**306**" is selected as the test vector, which represents the region of New Zealand. Two pest species are used for prediction in this experiment: one is an absence of species in New Zealand (e.g. acanthocoris scabrator (ACACS1) - class label 0); while the other is a presence of species in New Zealand (e.g. ceratitis capitata (CERTCA) - class label 1).

7.6.2. Results and Analysis

Example 1: Absence of species (class label 0)

As mentioned previously, the first step to predicting the establishment of an individual pest species is to define its nearest neighbours. Based on its nearest neighbours, the establishment of an individual pest species is further investigated using the WKNN and WWKNN prediction models, described as follows:

1) Using the WKNN Prediction Model

As shown in Figure 7.5, the test vector is predicted as "**0**" which precisely matches the actual output class label based on its 20 nearest neighours. The features"**40**" (Rainfall (mm) for the first summer month), "**47**" (Rainfall (mm) for the second winter month), and "**48**" (Rainfall (mm) for the third winter month) are selected as being the most important climatic factors for predicting the test vector.

```
======================== WKNN Prediction Model Result ========================

Threshold = 0.50    No. of Neighbors = 20

Selected Feature(s) = 40, 47, 48,

Best Accuracy = 90.67 %

      --------- Output for the target vector ---------

Actual Class = 0           Predicted Class = 0           Predicted Output = 1.231280e-002
```

Fig.7.5: Output of absence of species predicted using the WKNN prediction model.

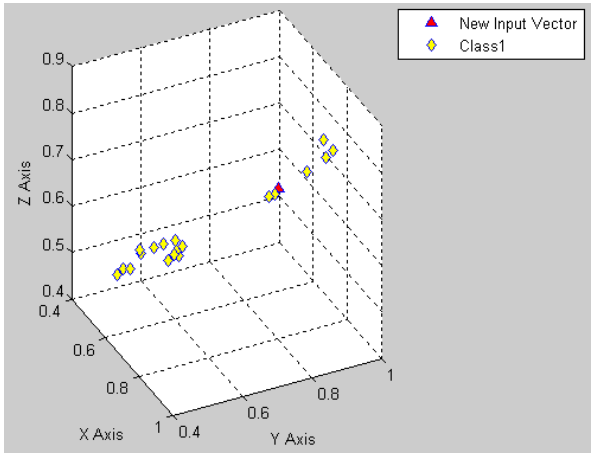


Fig.7.6: Overview of the nearest neighbours of absence of species using the WKNN algorithm.

2) Using the WWKNN Prediction Model

As demonstrated in Figure 7.7, the test vector is predicted as "**0**" which precisely matches the actual output class label based on its 30 nearest neighours. The features"**40**" (Rainfall (mm) for the first summer month), "**47**" (Rainfall (mm) for the second winter month), and "**48**" (Rainfall (mm) for the third winter month) are selected as being the most important climatic factors for predicting the test vector.

```
======================== WWKNN Prediction Model Result ========================

Threshold = 0.50    No. of Neighbors = 30

Selected Feature(s) = 40, 47, 48,

The Weight of Selected Feature(s) = 1, 0.74815, 0.72654,

Best Accuracy = 89.54 %

      --------- Output for the target vector ---------

Actual Class = 0           Predicted Class = 0           Predicted Output = 1.974647e-002
```

Fig.7.7: Output of absence of species predicted using the WWKNN prediction model.

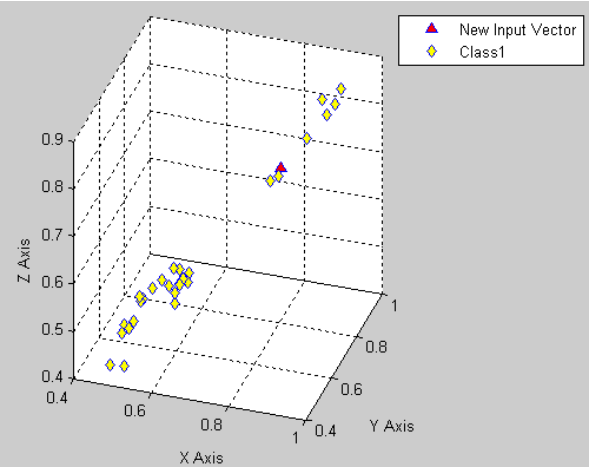


Fig.7.8: Overview of the nearest neighbours of absence of species using the WWKNN algorithm.

Knowledge Discovery

As observed in Figures 7.5 and 7.7, both WKNN and WWKNN prediction models give an accurate prediction for an individual absence of species. The features "**40**" (Rainfall (mm) for the first summer month), "**47**" (Rainfall (mm) for the second winter month), and "**48**" (Rainfall (mm) for the third winter month) are selected as the most important climatic factors for predicting the test vector using both the WKNN and WWKNN prediction models. Moreover, Figure 7.7 also shows the weight of each selected feature. This is because, in the WWKNN algorithm, the output of each new input vector is dependent upon the distance between the existing vectors and the new input vector, and the power of each vector is weighted according to its importance within the local space to which the new input vector belongs.

To further investigate the effects of different threshold settings on overall accuracy, the number of nearest neighbours is kept the same but the accuracy based performance evaluation is carried out using different threshold values ranging from a minimum value of 0.1 to the maximum value of 1.

Figure 7.9 shows the influence of different threshold settings on the overall accuracy achieved using the WKNN algorithm. Initially, the accuracy achieved is 88.42% for threshold values ranging from 0.1 to 0.4. The accuracy increases to 90.67% for threshold values ranging from 0.5 to 0.8. However, the accuracy decreases to 88.42% again for threshold values ranging from 0.9 to 1.

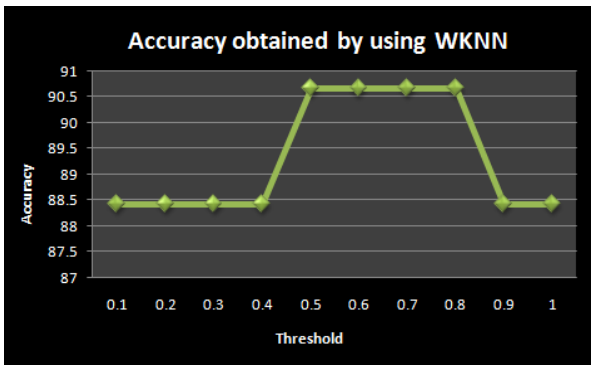


Fig.7.9: The threshold settings effect on the accuracy of absence of species obtained using the WKNN prediction model.

Figure 7.10 shows the influence of different threshold settings on the overall accuracy achieved using the WWKNN algorithm. Initially, the accuracy achieved is 78.32% for threshold values ranging from 0.1 to 0.2, and increases to 81.33% for threshold values ranging from 0.3 to 0.4. The accuracy significantly increases to 89.54% for threshold values ranging from 0.5 to 0.7. However, the accuracy drops down to 81.33% for threshold values ranging from 0.8 to 0.9, and further decreases to 78.32 when the threshold value is 1.

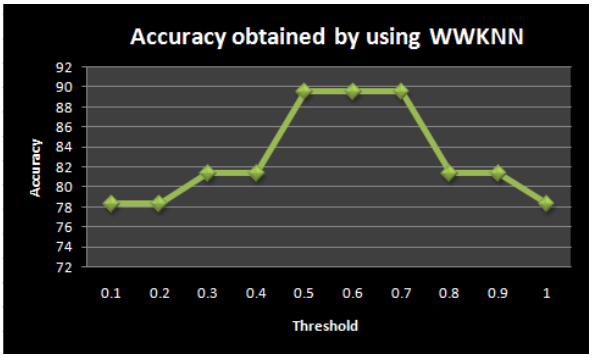


Fig.7.10: The threshold settings effect on the accuracy of absence of species obtained using the WWKNN prediction model.

As shown in Figures 7.9 and 7.10, the threshold values ranging from 0.5 to 0.7 provide the highest accuracy when using either algorithm.

Example 2: Presence of species (class label 1)

Firstly, the experiments begin with definition of the nearest neighbours for the test vector. Based on its nearest neighbours, the establishment of an individual presence of species is investigated using the WKNN and WWKNN prediction models, described as follows:

1) Using the WKNN Prediction Model

Figure 7.11 shows that features "**3**" (Mean temperature (celsius) for month of March), "**14**" (Mean temperature (celsius) for the second summer month), and

"**4**" (Mean temperature (celsius) for month of April) are selected as being the most significant climatic factors for predicting the test vector. Furthermore, based on its 30 nearest neighbours, the output of the test vector is predicted as "**1**" which accurately matches the actual output class label.

```
======================== WKNN Prediction Model Result ========================

Ihreshold = 0.50   No. of Neighbors = 30

Selected Feature(s) = 3, 14, 4,
Best Accuracy = 83.00%

     --------- Output for the target vector ---------

Actual Class = 1          Predicted Class = 1          Predicted Output = 1.265146e+000
```

Fig.7.11: Output of presence of species predicted using the WKNN prediction model.

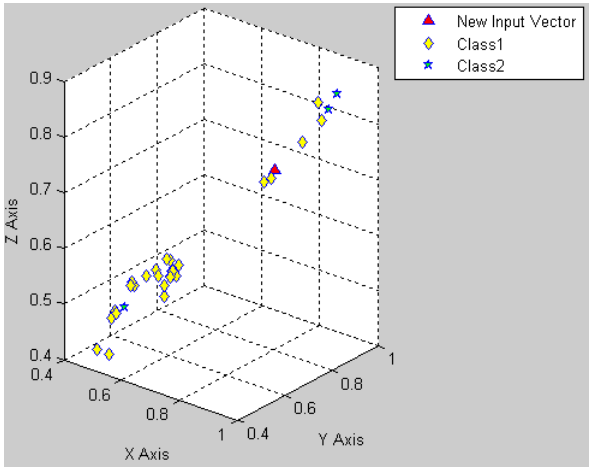


Fig.7.12: Overview of the nearest neighbours of presence of species using the WKNN algorithm.

2) Using the WWKNN Prediction Model

Figure 7.13 shows that features "**3**" (Mean temperature (celsius) for month of March), "**14**" (Mean temperature (celsius) for the second summer month), and "**4**" (Mean temperature (celsius) for month of April) are selected as being the most significant climatic factors for predicting the test vector. Moreover, based on its 30 nearest neighbours, the output of the test vector is predicted as "**1**" which accurately matches the actual output class label.

```
======================== WWKNN Prediction Model Result ========================

Threshold = 0.50    No. of Neighbors = 30

Selected Feature(s) = 3, 14, 4,

The Weight of Selected Feature(s) = 1, 0.91939, 0.88457,

Best Accuracy = 85.00%

      --------- Output for the target vector ---------

Actual Class = 1            Predicted Class = 1            Predicted Output = 1.389332e+000
```

Fig.7.13: Output of presence of species predicted using the WWKNN prediction model.

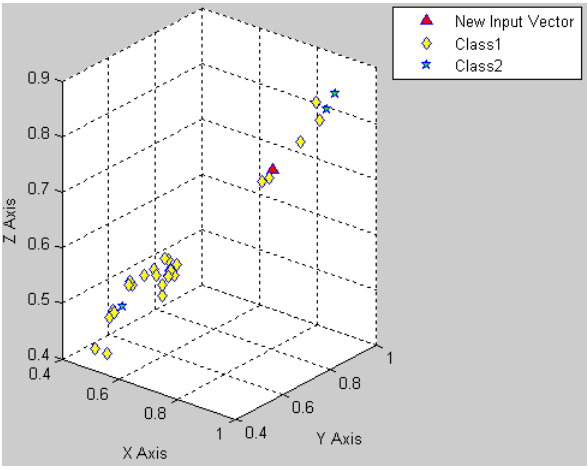


Fig.7.14: Overview of the nearest neighbours of presence of species using the WWKNN algorithm.

Knowledge Discovery

As observed in Figures 7.11 and 7.13, both WKNN and WWKNN prediction models give an accurate prediction for an individual presence of species. The features "**3**" (Mean temperature (celsius) for month of March), "**14**" (Mean temperature (celsius) for the second summer month), and "**4**" (Mean temperature (celsius) for month of April) are selected as being the most important climatic factors for predicting the test vector using both the WKNN and WWKNN prediction models. Figure 7.13 also shows the weight of each selected feature. As because in the WWKNN algorithm, the output of each new input vector is dependent upon the distance between the existing vectors and the new input vector, as well as the power of each vector is weighted according to its importance within the local space to which the new input vector belongs.

The effects of different threshold settings on the overall accuracy is also investigated, the number of nearest neighbours is kept the same but the accuracy

based performance evaluation is carried out using different threshold values ranging from a minimum value of 0.1 to the maximum value of 1.

Figure 7.15 shows the influence of different threshold settings on the overall accuracy achieved using the WKNN algorithm. Initially, the accuracy achieved is 67.34% for threshold values ranging from 0.1 to 0.2, and jumps to 70.33% for threshold values ranging from 0.3 to 0.4. The accuracy significantly increases to 83.00% for threshold values ranging from 0.5 to 0.7. However, accuracy decreases to 73.00% again for threshold values ranging from 0.8 to 1.

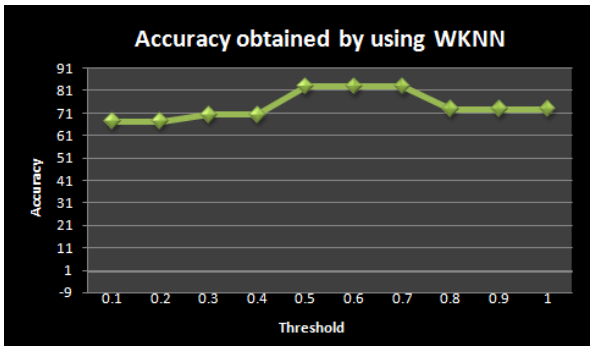


Fig.7.15: The threshold settings effect on the accuracy of presence of species obtained using the WKNN prediction model.

Figure 7.16 shows the influence of different threshold settings on the overall accuracy achieved using the WWKNN algorithm. Initially, the accuracy achieved is 78.25% for threshold values ranging from 0.1 to 0.2, but decreases to 70.33% for threshold values ranging from 0.3 to 0.4. The accuracy significantly increases to 85.00% for threshold values ranging from 0.5 to 0.8. However, the accuracy drops down to 70.33% again for threshold values ranging from 0.9 to 1.

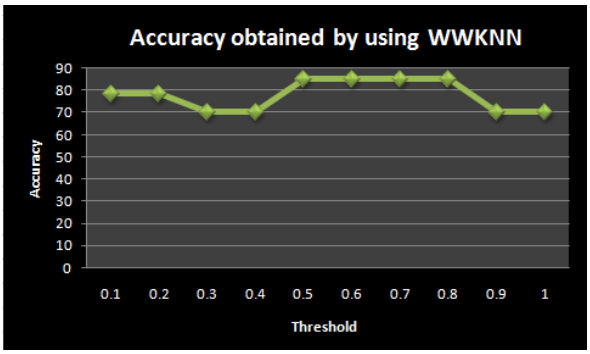


Fig.7.16: The threshold settings effect on the accuracy of presence of species obtained using the WWKNN prediction model.

As shown in Figures 7.15 and 7.16, the threshold values ranging from 0.5 to 0.7 provide the highest accuracy when using either algorithm.

As demonstrated above, the results prove that both absence and presence of species are accurately predicted using the WKNN and WWKNN prediction models. The two major reasons for this are:

- ✧ Using *k*-nearest neighbour (KNN) algorithm with Euclidean distance measure, similarities between the test vector and its nearest neighbours are investigated. The KNN algorithm estimates values of a potential model for an individual point (new input vector) of the problem space using additional information related to that point (the nearest neighbours of the test point).

- ✧ Additionally, the system automatically selects an optimal set of features to verify the correct range of climatic factors that effect on determining the potential establishment of pest species.

7.7. Summary

In this chapter, a detailed comparative analysis of GA-optimized WKNN and WWKNN personalised models and global SVM and local ECF models was performed on a real world pest-related climate data set. The experimental results proved that the personalised modelling approach gave better classification accuracy when compared with the global and local modelling approaches. In addition, this chapter also presented a detailed experimental study on predicating the establishment of an individual pest species using personalised prediction models. The results proved that the output for an individual absence and presence of species were accurately predicted using the WKNN and WWKNN prediction models. The effects of different threshold settings on the overall accuracy of WKNN and WWKNN algorithms were also investigated. The results of this presented that accuracy subject to different threshold settings. The best accuracy was obtained using threshold values ranging from 0.5 to 0.7.

Chapter 8
Conclusions and Future Directions

8.1. Conclusions

The concept of personalised modelling has its roots in machine learning technologies that have been utilized for numerous decades to understand, evaluate and solve a variety of modelling problems in the fields of personalised medicine, personalised drug design as well as problems in business, finance, crime prevention, and so on. However, personalised modelling is not without issues, such as defining the correct number of neighbours, and defining an appropriate number of features. For this reason, the goal of this research is to study and address these issues by creating a novel framework and system for personalised modelling that allows users to select and optimise the most important features and nearest neighbours in relation to a certain problem based on a weighted variable distance measure in order to provide more precise accuracy and personalised knowledge when compared with the global modelling and local modelling approaches. In this study, the genetic algorithm (GA) is adopted and integrated with the WKNN and WWKNN classifiers to solve the parameter optimization and feature selection problems, based on the idea originally from Siedlecki and Slansky (1989).

The proposed GA-based personalised modelling (GAPM) system has two major contributions, which are briefly presented as follows:

(1) It allows users to use the GA–optimized WKNN and WWKNN algorithms to create classification models that test classification accuracy in order to provide more accurate and predictive knowledge and information for investigators.

(2) It also allows users to create personalised prediction models for each new individual input vector using the WKNN and WWKNN predication algorithms. One drawback is that the genetic algorithm does not collaborate with the two base algorithms, thus the output of a single target vector might not be optimal.

This novel GAPM is first adopted to perform a comparative analysis of global and local modelling approaches against personalised modelling approach to compare the classification accuracy. The experiments are run on a benchmark data set (Sonar, which was cited from the UCI-Repository), a leukaemia cancer data set and a real world pest-related climate data set, respectively. All the experimental results prove that the GA-

optimized WKNN and WWKNN algorithms provide better results when compared with the global (SVM) and local modelling (ECF) approaches. Secondly, this novel GAPM is used for prediction of an individual sample from the leukaemia cancer data set and the pest-related climate data set. The results prove that the output for an individual target vector is accurate using both the WKNN and WWKNN prediction models. As mentioned above, the concept of personalised modelling is worth further investigation and it has vast scope for future development.

8.2. Strengths of this Study

- ✧ In this study, GA is adopted to optimize various parameters (optimal threshold, optimal number of nearest neighbours for every new input vector, and an optimal set of features contribute most towards the classification task) in order to deal with the opening questions: "*How many nearest neighbours should be selected for every new input vector?*" and "*What features are significant for every new input vector?*" GA is an optimal method to solve complex optimization problems after a number of iterative computations, as well as being able to deal with a large problem space efficiently.

- ✧ This study performs two case studies to compare the classification accuracy between the personalised modelling approach and the global and local modelling approaches using the proposed GAPM. The case studies involve two data sets and in different fields, the leukaemia cancer data set is from the health and clinical area, while the real world pest-related climate data set is from the ecological area. The experimental results facilitate new knowledge discovery that may help develop more innovative and effective therapeutic treatments for leukaemia cancer patients, as well as to effectively and precisely monitor, control and predict the establishment of pests.

- ✧ One of the main strengths of the GAPM system is its performance in comparison to the global and local modelling approaches. In addition, the novel system provides more precise personalised knowledge and a better understanding of meaningful information.

- ✧ In this study, a graphical user interface (GUI) for both GA-optimized WKNN and WWKNN algorithms is designed using MATLAB, which are easy to use (see

Figures 5.1 and 5.2).

- ✧ The proposed GAPM also comprises other functions, such as allowing users to visualize how the entire data is distributed (see Figure 5.3), to view or modify the loaded data set (see Figure 5.4), or to visualise the nearest neighbours of an individual vector in a 3-D problem space (see Figure 5.5).

8.3. Limitations of this Study

- ✧ When creating personalised prediction models for an individual input vector, the genetic algorithm is not integrated with the WKNN and WWKNN algorithms, thus the final output for an individual input vector may not be optimal.

- ✧ In this study, only the default crossover rate (0.8) and mutation rate (0.01) are chosen to investigate the performance of a typical GA.

- ✧ In addition, only one data splitting / sampling technique (cross-validation) is used to measure how well the results of a statistical analysis can generalize to an independent data set. The cross-validation method is very computationally expensive due to the large number of times the training process is repeated.

8.4. Future Directions

As mentioned above, there are a number of areas where future work is required:

- ✧ In order to significantly improve the performance of an individual input vector, the genetic algorithm needs to be integrated with the WKNN and WWKNN classifiers to create personalised prediction models.

- ✧ In the future, different data splitting / sampling techniques can be used to estimate the generalization error for feature selection. For instance, “Bootstrap” is another simple, but powerful data sampling method to evaluate the statistical accuracy.

- ✧ As the right choice of parameter values is an important issue in the GA, in order to improve accuracy and provide more precise personalised knowledge, the relationship between crossover and mutation rates, and how well a typical GA performs using different range of crossover and mutation rates needs to be looked at.

- ✧ In this study, the global model and local model are created based on the inductive approach using the SVM algorithm and the ECF algorithm in NeuCom. In future study, both the SVM and ECF parameters will be optimized using GA, and a comparative analysis of GA-based personalised modelling as opposed to GA-based global and GA-based local modelling will be presented.

- ✧ In addition, new methods can be used for personalised modelling in the future, such as the transductive neuro-fuzzy inference model with weighted data normalization (TWNFI). TWNFI is a dynamic neuro-fuzzy system with local generalization, in which, either the *Zadeh-Mamdani* type fuzzy inference engine or the *Takagi-Sugeno* fuzzy inference engine is applied. This approach not only results in a "personalised" model with better accuracy of prediction for an individual new input vector, but also presents the most significant features for the model that may be used for personalised medicine.

- ✧ Nowadays, computer and information technology is playing an increasingly critical role in medicine, health and life sciences research. In the future, a further study in the area of personalised medicine, especially investigating various forms of cancer (e.g. breast cancer, brain cancer, and liver cancer) will be conducted. It can provide more precise personalised prediction, diagnosis, prognosis tracking, and targeted therapy.

In my PhD study, two new methods will be adopted and further developed for personalised modelling which are "Probabilistic Evolving Spiking Neuron Networks (peSNN)" and "Quantum-inspired GA (QiGA)". As spiking processes in biological neurons are stochastic by nature and much has become known about these, it would be possible to add new information processing functionality to a neuronal model through introducing probabilistic parameters. However, one challenge is what method to use to deal with these probabilistic parameters for efficient learning and generalization to take place. In my PhD study, the peSNN will be combined with a QiGA to optimize features and parameters of a peSNN for classification, exploring quantum parallelism based on probabilistic superposition of states. In this way, the input features as well as information spikes will be represented by quantum bits that result in exponentially faster feature selection and model learning. The methods will be first applied on synthetic data and real Brain injury data for a personalised outcome prediction.

References

Aizerman, E. M., & Braverman, L. R. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automat Remote Control*, *25*, 821–837.

Allison, D., Cui, X., et al. (2006). Microarray data analysis: from disarray to consolidation and consensus. *Nature Reviews Genetics*, *7*(1), 55-65.

Angluin, D., & Smith, C. H. (1983). Inductive Inference: theory and methods. *Computing Surveys*, *15*(3), 237-269.

Angrist, M. (2005). Breast Cancer: integrating the patient with her genome. *Trends in biotechnology, 23*(1), 3-5.

Arbib, M. (2003). The handbook of brain theory and neural networks. Cambridge, MIT Press, MA.

Axelrod, R. (1984). *The evolution of cooperation*. Basic Books.

Axelrod, R., & Dion, D. (1988). The further evolution of cooperation. *Science 242, 4884*, 1385–1390.

Baek, O., Gaffney, T., Joshi, K., Robson, B., Rosen, D., Stahlbaum, C., Taylor, R., & Vortman, P. (n.d.). *Personalised healthcare 2010: Are you ready for information-based medicine?* Retrieved July 10, 2008, from http://www-935.ibm.com/services/in/igs/pdf/g510-3565-personalized-healthcare-2010.pdf

Barlow, N., & Goldson, S. (2002). *Biological Invasions Economic and Environmental Costs of Alien Plant, Animal and Microbe Species*. Florida, CRC Press.

Biermann, A. W., & Krishnaswamy, R. (1976). Constructing programs from example computations. *IEEE Transactions on Software Engineering*, *2,* 141-153.

Bosnic, Z., Kononenko, I., et al., (2003). Evaluation of prediction reliability in regression using the transduction principle. *Computer as a Tool*. *8*(2), 99-103.

Braga-Neto, U., Hashimoto, R., et al. (2004). Is cross-validation better than resubstitution for ranking genes? *Bioinformatics*, *20*(2), 253-258.

Brosse, S., Guegan, J. F., Tourenq, J. N., & Lek, S. (1999). Use of artificial neural networks to assess fish abundance and spatial occupancy in the littoral zone of a mesotrophic lake. *Ecological Modelling, 120, 299–311.*

Chen, M., Han, J., & Yu, P. (1996). Data mining: An overview from database perspective. *IEEE Transactions on Knowledge and Data Engineering*, *8*(6), 866-883.

Chen, Y., Wang, G., & Dong, S. (2003). Learning with progressive transductive support vector machine. *Pattern Recognition Letters, 24*(12), 1845 - 1855.

Craven, P., & Wahba, G. (1979). Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, *31*(4), 377–403.

Dockerty, J. D. (2008). Epidemiology of childhood leukemia in New Zealand: Studies of infectious hypotheses. *Blood Cells, Molecules, and Diseases*, *42*(2009), 113-116.

Doucet, P. G. (1974). The syntactic inference problem for D0L sequences. *L-Systems,* [Lecture Notes]. New York, Springer-Verlag.

Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. New York, John Wiley and Sons.

Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on System Man and Cybernetics*, 325-327.

Feliciangeli, H., & Herman, G. (1977). Algorithms for producing grammars from sample derivations: A common problem of formal language theory and developmental biology. *Journal of Computer and Systems Sciences*, 7, 97-118.

Ferri, F. J., Pudil, P., Hatef, M., & Kittler, J. (1994). *Comparative study of techniques for large scale feature selection*. Amsterdam, Springer Verlag.

Fix, E., & Hodges, J. L. (1951). *Discriminatory analysis: Nonparametric discrimination: Consistency properties*. Randolph Field, Texas: UASF School of Aviation Medicine.

Fung, G. M., & Mangasarian, O. L. (2004). A feature selection newton method for support vector machine classification. *Computational Optimization and Applications*, *28*(2), 185-202.

Ginsburg, G. S., & McCarthy, J. J. (2001). Personalized medicine: revolutionizing drug discovery and patient care. *Trends in Biotechnology*, *19*(12), 491-496.

Gold, E. M. (1967). Language identification in the limit. *Infection Control*, *10,* 447-474.

Goldberg, D. E. (1989). *Genetic Algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley.

Goh, L., Song, Q., et al. (2004). *A novel feature selection method to improve classification of gene expression data*. Paper presented at the ACM International Conference Proceeding Series: Proceedings of the second conference on Asia-Pacific bioinformatics, Dunedin, New Zealand.

Gunn, S. (1998). *Support vector machines for classification and regression*: Image speech and intelligent systems research group, University of Southampton.

Hand, D. J., & Vinciotti, V. (2003). Choosing k for two-class nearest neighbour classifiers with unbalanced classes. *Pattern Recognition Letters, 24*, 9-10.

Hillis, W. D. (1992). Co−evolving parasites improve simulated evolution as an optimization procedure. *Artificial Life II*, 228-234.

Hu, Y. J. (2006). *Gene selection based on consistency modelling, algorithm and applications*. Master Thesis. Auckland University of Technology. Retrieved December 11, 2008, from Auckland University of Technology Digital Theses.

Huang, L., Song, Q., & Kasabov, N. (2005). *Evolving connectionist systems based role allocation of robots for soccer playing*. Paper presented at the 2005 proceedings of IEEE International Symposium on Intelligent Control, Vancouver, Canada.

Iakovidis, I. (2007). *Challenge 5 - Towards sustainable and personalised healthcare*. Retrieved July 10, 2008, from http://ec.europa.eu/information_society/events/phs_2007/docs/slides/phs2007-iakovidis-ch5-1a.pdf

Jakobsson, E, Wang, M. D., & Molnar, L. (2007, October 14) *Bio-Nano-Info integration for personalized medicine*. Paper presented at the 7th IEEE International Conference on Bioinformatics and Bioengineering Cutting-Edge Research Workshop Keynote Lecture, Harvard Medical School Conference Center, Boston, Massachusetts, USA.

Joachims, T. (1999). *Transductive Inference for Text Classification using Support Vector Machines.* Paper presented at the Proceedings of the Sixteenth International Conference on Machine Learning, Dortmund, Germany.

Joachims, T. (2003). *Transductive Learning via Spectral Graph Partitioning.* Paper presented at the Proceedings of the Twentieth International Conference on Machine Learning, ICML-2003, Washington DC.

Jones, R. E., & Kitching, R. L. (1981). Why an ecology of pests? *The Ecology of Pests, Some Australian Case Histories*. Melbourne, CSIRO Australia.

Kasabov, N. (2004). Knowledge based neural networks for gene expression data analysis, modelling and profile discovery. *Drug Discovery Today*, BIOSILICO, vol.2, No.6, 253-261.

Kasabov, N. (2007a). *Evolving connectionist systems: methods and applications in bioinformatics, brain study and intelligent machines*. London, Springer Verlag.

Kasabov, N. (2007b). Global, local and personalised modelling and pattern discovery in bioinformatics: An integrated approach. *Pattern Recognition Letters*, *28*, 673–685.

Kasabov, N., & Pang, S. (2004). Transductive support vector machines and applications in bioinformatics for promoter recognition. *Neural Information Processing - Letters and Reviews, 3*(2), 31-38.

Kasabov, N., Pang, S., Soltic, S., Worner, S., & Peacock, L. (2004, November 22-25). *Dynamic Neuro-fuzzy Inference and Statistical Models for Risk Analysis of Pest Insect Establishment*. Paper presented at the 11th International Conference on Neural Information Processing, Science City, Calcutta.

Kasabov, N. (2009). To spike or not to spike: A probabilistic spiking neuron model. *Neural Networks*.

Kriete, A. (2004). Gene expression analysis enriched. *Drug Discovery Today, 9*(21), 913-914.

Kukar, M. (2003). Transductive reliability estimation for medical diagnosis. *Artifical intelligence in medicine, 29*, 81 - 106.

Lai, C., Reinders, M., et al. (2004). *On univariate selection methods in gene expression datasets*. Paper presented the Tenth Annual Conference of the Advanced School for Computing and Imaging, Port Zelande, Netherlands.

Lankhorst, M. M., van Kranenburg, H., Salden, A., & Peddemors, A. J. H. (2002, January 7-10). *Enabling technology for personalizing mobile services*. Paper presented at the proceedings of the 35th Annual Hawaii International Conference on System Sciences, Hawaii, United States.

Larose, D. T. (2005). *Discovering knowledge in data: An Introduction to Data Mining*. Hoboken, New Jersey, United States: John Wiley & Sons, Inc.

Lee, Y. J., & Mangasarian, O. L. (2001). SSVM: A smooth support vector machine for classification. *Computational Optimization and Applications*, *22*(1), 5-21.

Lesko, L. J. (2007) "Personalized medicine: Elusive dream or imminent reality?" *Clinical Pharmacology & Therapeutics*, *81*, 807 -816.

Liang, G., Song, Q., & Kasabov, N. (2004). *A novel feature selection method to improve classification of gene expression data*. Paper presented at the 2nd Asia-Pacific Bioinformatics Conference (APBC2004), Dunedin, New Zealand.

Magoulas, G. D., & Dimakopoulos, D. N. (2005). *Designing personalised information access to structured information spaces*. Retrieved July 11, 2008, from http://www.dcs.bbk.ac.uk/~gmagoulas/Designing%20Personalised%20Information%20Access.pdf

McGowan, J. P., Kushmerick, N., & Smyth, B. (2002). *Computer Science* [Lecture notes]. Dublin, Ireland: University College Dublin.

Mitchell, T. M. (1997). *Machine learning*. Boston, McGraw−Hill, Massachusetts.

Mooney, H. A., & Drake, J. A. (1989). *Biological invasions: a SCOPE program overview*. Chichester, NY: John Wiley & Sons.

Nevins, J. R., Huang, E. S., Dressman, H., Pittman, J., Huang, A. T., & West, M. (2003). Towards integrated clinico-genomic models for personalized medicine: combining gene expression signatures and clinical factors in breast cancer outcomes prediction. *Human Molecular Genetics, 12*(2), 153–157.

Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, 24(12), 1565-1567.

Personalized Medicine Coalition. (2008). *The case for personalized medicine*. Retrieved July 10, 2008, from http://www.personalizedmedicinecoalition.org/communications/pmc_pub_11_06.php

Pimentel, D. (1986). Biological invasions of plants and animals in agriculture and forestry. *Ecology of Biological Invasions of North America and Hawaii*. New York, Springer Verlag.

Pittman, J., Huang, E., Dressman, H., Hong, C.-F., Cheng, S., Tsou, M.-H., et al. (2004). Integrated modelling of clinical and gene expression information for personalized prediction of disease outcomes. *National Academy of Science, 101*(22), 8431-8436.

Proedrou, K., Nouretdinov, I., Vovk, V., & Grammerman, A. (2002). Transductive confidence machines for pattern recognition. *Lecture notes in artificial intelligence, 2430*, 381-390.

Provost, F., & Kolluri, V. (1999). A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery*, *2*, 131-169.

Putnam, H. (1975). *Probability and confirmation*. Cambridge University. Cambridge, MIT Press.

Ransohoff, D. F. (2004). Rules of evidence for cancer molecular marker discovery and validation. *Nature Reviews Cancer*, *4*, 309-314.

Sailer, R. J. (1983). History of insect introductions. *Exotic Plant Pests and North American Agriculture*. New York, Academic Press.

Schölkopy, A. J., & Smola, P. L. (2000). New support vector algorithms. *Neural Computation*, *12*(5), 1207-1245.

Shaw, D., Swarvout, W., & Green, C. (1975). *Inferring LISP programs from example problems*. Paper presented at the proceedings of the 4th International Joint Conference on Artificial Intelligence, Tbilisi, Georgia.

Siedlecki, & Slansky. (1989). A note on genetic algorithms for large scale feature selection. *Pattern Recognition Letters, 10*, 335-347.

Song, Q., & Kasabov, N. K. (2005). NFI: A neuro-fuzzy inference method for transductive reasoning. *Transactions On Fuzzy Systems*, *13*(6), 799-807.

Spang, R. (2003). Diagnostic signatures from microarrays: a bioinformatics concept for personalised medicine. *Biosilico, 1*(2), 64-68.

Stearns, S. D. (1976). *On selecting features for pattern classifiers*. Paper presented at the 3rd international Conference on Pattern Recognition, Coronado, CA.

Stynes, B. (2002) *Pest risk analysis: methods and approaches*. Retrieved July 15, 2008, http://www.nappo.org/PRA-Symposium/PDF-Final/Stynes.pdf

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3), 293-300.

TEMU. (2008). *Personalised Medicine: Current Trends and Scientific Challenges*. Retrieved July 10, 2008, from http://www.temu.gr/2008/sessions.html

Tyrer, J., Duffy, S., & Cuzick, J. (2004). A breast cancer prediction model incorporating familial and personal risk factors. *Statistics in medicine, 23*, 1111-1130.

Vapnik, V. (1998). *Statistical learning theory*. NY, Wiley-Interscience.

Vapnik, V. (2005). *Crucial questions of theory*. Retrieved July 14, 2008, from http://www.lancs.ac.uk/users/esqn/windsor04/handouts/vapnik.pdf

Vapnik, V., & Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, *24*, 774–780.

Wagacha, P. W. (2003). *Instance based learning: K-nearest neighbour*. Retrieved August 20, 2008, from www.uonbi.ac.ke/acad_depts/ics/course_material/machine_learning/kNN.pdf

Walsh, F. (2009). *Era of personalised medicine awaits*. Retrieved April 8, 2009, from http://news.bbc.co.uk/2/hi/health/7954968.stm

Weston, J., Pérez-Cruz, F., Bousquet, O., Chapelle, O., Elisseeff, A., & Schölkopf, B. (2003). Feature selection and transduction for prediction of molecular bioactivity for drug design. *Bioinformatics, 19*(6), 764-771.

Wexler, K., & Culicover, P. (1980). *Formal principles of language acquisition*. Cambridge, MIT Press.

Williams. (2003). Personalised health planning. *Science, 300*, 549.

Wolf, L., & Mukherjee, S. (2004). *Transductive learning via model selection*. Cambridge, MA: The center for Biological and Computational Learning, Massachusetts Institute of Technology.

Worner, S. P. (1994). Predicting the establishment of exotic pests in relation to climate. *Quarantine Treatments for Pests of Food Plants*. Westview Press, Boulder, CO.

Wu, D., Cristianini, N., Shawe-Taylor, J., & Bennett, K. P. (1999, June 27 - 30). *Large Margin Trees for Induction and Transduction.* Paper presented at the Proceedings for 16th International conference of machine learning, Bled, Slovenia.

Wu, S. L., Crestani, F., & Bi, Y. X. (2001). *Evaluating score normalization methods in data fusion*. Retrieved April 23, 2008, from http://personal.cis.strath.ac.uk/~fabioc/papers/06-airs.pdf

Yang, Y., & Pedersen, J. O. (1997). *A comparative study on feature selection in text categorization*. Paper presented at the 14th International Conference on Machine Learning, Nashville, TN.

Appendices

Appendix A: Overview of NeuCom

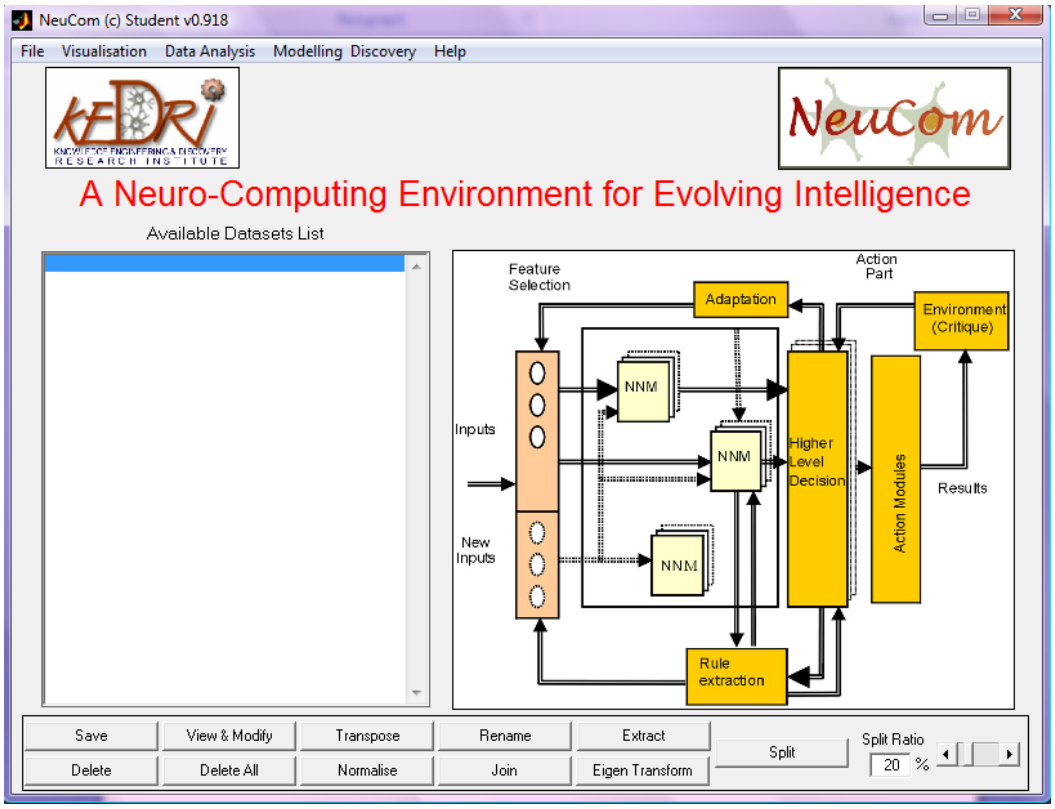


Fig. A1: Overview of NeuCom.

NeuCom is a New Generation Computer Environment

NeuCom is a self-learning, reasoning and programmable computer environment which is based on the theory of *Evolving Connectionist Systems* (ECOS) as proposed by Professor Kasabov (2004). It has the ability to learn from data, hence it is always developing new connectionist modules. The modules have the capability of adopting new data in a life-long learning, on-line incremental mode, and might extract valuable and meaningful rules in order to help users discover new knowledge in their individual fields.

NeuCom can be used to Solve Complex Problems

NeuCom can solve complex problems including classification, clustering, prediction, pattern recognition, and adaptive control from databases in a multi-dimensional and probably changing the data environment. In the recent, NeuCom has been widely adopted in different areas such as education, science, business, medicine, and bioinformatics.