Republic of Iraq
Ministry of Higher Education
And Scientific Research
Al Nahrain University
College of Science
Department of Computer Science

Proposed Multi Agents Classification System using Neural Networks

A Thesis

Submitted to the College of Science/Al-Nahrain University as a partial Fulfillment of the requirements for the Degree of Master of Science in Computer Science.

By

Mohammed Abdallazez Mohammed

B.Sc. Computer Science Dept. / College of Science / University of Karbala

Supervised by

Dr. Ban N. Dhannoon
(Professor)

2016 1437

ACKNOWLEDGMENT

I would like to express my sincere appreciation to my advisor **Dr. Ban N. Dhannoon** for guiding me through the project and for detailed suggestions; I did not only learn knowledge from her but also learned a lot from his great personality.

My sincere thanks also goes to **Dr. Haithem A. Alani** the pervious head of computer science department at Al_Nahrain University and to the **teachers in computer science department** who provided me with advice at times of critical need, have support helped me in all the time of research.

Finally, I would like thanks **Miss Hanaa Mohammed Mashjel** to support me in hard time.

Dedication

In every step in my life, I am being reminded that, the learning journey never ends, and always a new learning should be started, such that the Science is in everything we do and the scientists of the past and present are inspirations for all of us.

To whom who taught me that …

To whom who put me on my first step …

To my parents

To my professors

To all who live in my memory

To my friends

To researchers in my country

To all of them …I dedicate my work

Mohammed al majed

<u>Supervisor Certification</u>

I certify that this thesis entitled "**Proposed Multi Agents Classification System using Neural Networks**" was prepared by "**Mohammed Abdallazez Mohammed**" under my supervision at the College of Science/Al-Nahrain University as a partial fulfillment of the requirements for the Degree of Master of Science in Computer Science.

Signature:
Name: **Dr. Ban N. Dhannoon**
Scientific Degree: **Professor**
Date: / / *2016*

In view of the available recommendations, I forward this **thesis** for debate by the examining committee.

Signature:
Name: **Dr. Ban N. Dhannoon**
Scientific Degree: **Professor**
Title: Head of Computer Science Department / Collage of Science / Al Nahrain University
Date: / / *2016*

Abstract

Multiagents technology took a significant role in the field of decision making and machine learning for solving complex problems in the real world. They simulate human ability to decision making where they have ability to reasoning and behave autonomously to solve problems or to support human user.

In this thesis, a classification system using hierarchal multi agent's technology based on neural network and fuzzy logic is introduced, where each agent is implemented as a neural network (trained using back propagation learning algorithm). The system classifies a collection of datasets with some degree of generalization.

The system consists of two layers of agents. The top layer contains one agent working as control agent. Its responsibility is to select the right agent from the agents in the bottom layer to classify the related pattern depending on features of data. If the selected pattern do not recognized, then it is declared as unknown pattern.

The developed system was tested using different standard datasets obtained from the University of California, Irvine (UCI) these are User Knowledge Level, Iris, and Banknote Authentication datasets. Earlier stopping criterion and regularization techniques were used to estimate the generalization of the agents. The final results indicated that the best generalization technique for user knowledge dataset is regularization with cross validation selection mode, while for both iris dataset and banknote authentication is the earlier stopping criterion, also accuracy of testing each classification agent as fallowing ,agent1 is 97.53% , agent2 and agent3 100%, agent4 69.04%.

List of Contents

List of Tables

List of Figures

List of Abbreviations

ANN	Artificial Neural Network
BP	Back Propagation algorithm
FN	False negative
FP	False positive
MAS	Multi Agent System
TP	True positive

Chapter One

General Introduction

1.1 Introduction

An agent is everything capable of observing its surrounding environment by sensors and acting based on that surrounding environment by actuators. A software agent receives file contents, network packets, and keystrokes as sensory inputs and takes action on the environment by writing files, sending network packets and displaying on the screen [Rus10]. The system that consists of a set of agents that can potentially interact is named a multi agent system (MAS). The related branch of artificial intelligence (AI) which deals with design and principles of multi agent systems is named as distributed AI [Ron07].

With the significance growing of complex software systems in the industry of software, the need for using agent technologies to produce large-scale industrial and commercial software systems is developed rapidly [Hai01]. Multi-Agent Systems (or Systems involves Software Agents) are becoming popular in the mainstream of development because an agent tries to process tasks intelligently and autonomously [Chi05]. The software agents and multi-agent systems, inherit many of DAI's goals, it also inherits features which come from AI such as easier maintenance and reusability [Hya96, Ali04]. Essentially the DAI focuses on processes like planning, problem-solving, search, learning and decision making which make the agent's behavior flexibility and rational [Ger99].

One of the most facing decisions-making tasks of human activity is classification. A classification problem happens when an object requires to be allocated into a predefined set or category based on some observed characteristics. Many problems in medicine, industry, science and business can be treated as classification problems [Guo00]. One of the fundamental ways to process data is classify the data into clusters or classes [Rui05]. Classification refers to the ability of a system to learn from examples how to distinguish cases in two or more categories. A collection of cases is used to learn the system, usually called the training set. Each case is represented by a class label and a set of variables. The system after training must be able to predict the label of the class for new cases. When there are two classes, the classification is referred to as binary classification; for more than two classes it is referred to as multiclass classification [Ant13].

Data classification tasks can be satisfied by using artificial neural networks. Artificial Neural Network (ANN) is a collection of interconnected units that learn from their environment (data), to get essential nonlinear and linear trends in the data, so that it supplies dependable predictions for new cases including even noisy information [San06]. To enhance neural network performance, one can use some data preprocessing like normalization and fuzziness [Sol97] [Jia15]. Sometimes fuzzy logic is used with neural network to improve effectiveness in a broad range of problems in the real world [Rob95].

1.2 Aim of the Thesis

The main aim is to develop a pattern classification system based on multi-layer agent technology implemented as neural networks using back propagation as a learning algorithm. This system must be able to classify datasets with a certain degree of generalization and reduce time and efforts when classifying unknown pattern.

1.3 Related Works

Many researchers work on classification problems to enhance accuracy of classification. Several of these research works are listed below:

I. F. Leon, A. D. Leca and G. M. Atanasiu [Flo10]: introduce analyzing to multi-agent system where agents use different training algorithms and different topologies for their neural networks, which they use to solve classification and regression problems provided by a user. These algorithms are: Quickprop, Rprop and Backpropagation, Backpropagation algorithm succeed in perform better than the other in terms of the total utility gained.

II. Preksha Pareek, [Pre12]: Used multi-Layer Perceptron network for classification task. The effect of network architecture on its performance is shown on table 1.1. The algorithm used in the application is gradient descent Backpropagation.

Table 1.1: classification results [Pre12].

Number of hidden layer	MSE	Gradient	Percentage of sample misclassified (%)
1	0.17	0.00518	4.1
2	0.0129	0.00581	5.4
3	0.229	0.00595	33.11

This table reveals that if the number of hidden layers is increased then the percentage of misclassified samples will be increased too.

III. Labhya Sharma, Utsav Sharma [Lab14]: introduced classifier based on neural network that handle the problem of classification. The architecture of neural network is multilayer perception. The classifier classifies iris database existing on the Internet. The data was division into the validation (20%) and training (80%) sets using cross validation the efficiency results were 99.2% for training set and 100% for validation set.

IV. H. M. Mashjel [Han14]: introduced general pattern classification model using Cooperative Neural Network, to solve a sub-tasks problems that composed a difficult classification problems. Back Propagation (BP) algorithm was tested and it founded that BP in pattern mode with adaptive learning rate gave the most excellent result in the proposed method. A data sets generated by simulator has been developed to this purpose. The generated dataset is partitioned in 40% unseen samples for the network test only, and 60% training set, the tests results indicated a success rate nearly 100%.

V. Mohit, R. R. Verma, et.al. [Moh15]: different classification techniques are compared using diverse datasets from University of California, Irvine (UCI) Machine Learning Repository. Time complexity and accuracy for execution is observed by each classifier. Lastly different classifiers are compared and as a result which classifier is best for relevant datasets is observed as shown in table 1.2.

Table 1.2: Iris classification with different algorithms.

Algorithms Name	Accuracy	Time	Ranking
Random forest	94.667	41	6
J48	96	41	3
Naïve Bayes	96	41	2
Multilayer Perceptron	96.667	41	1

1.4 Problem Statement

The design and implementation system processes the classification problem for many datasets, where in the real world there are amounts of data which different in type, features and size. To classify these data effectively and in good time they must be classified with suitable classifier for each collection (dataset) of data. Multi-layer agent's classifier used to solve the classification problem to speed up the classification process.

1.5 Thesis Layout

Beside this chapter, which provides an introduction to the thesis, this thesis contains the following chapters:

Chapter Two: "Theoretical Background"

Several fundamental concepts of agent technology, neural networks, learning and generalization, normalization and theory of fuzzy are introduced.

Chapter Three: "The Developed Classification System"

The details of the developed classification system and methods of training and testing agents with their steps and stages are introduced.

Chapter Four: "Experimental Results"

The results of experimental analysis of some training and testing techniques to choose the best method for each agent classifier, the corresponding performance for each one and results of applied some preprocessing operation like normalization and fuzziness are introduced.

Chapter Five: "Conclusions and Future Works"

This chapter provides a list of conclusions after evaluating the developed system. Also, it gives several suggestions for future work.

Chapter two

Theoretical Background

2.1 Introduction

Artificial intelligence means the study of intelligent behavior and how to make computer programs behaves intelligently [Far00]. The new artificial intelligence (AI) approaches focused on the idea of a rational agent. An agent that always attempts to optimize a proper performance measure is called a ‘rational agent.' Such a description of a ‘rational agent’ is quite general and can include robotic agents (having wheels as actuators, cameras as sensors), human agents (possess hands as actuators, eyes as sensors) or software agents (own a graphical user interface GUI as sensor and actuator). From this viewpoint, AI can be considering as the study of the design and principles of artificial agents. Agents are rarely independent systems. In many cases, they exist and interact with other agents in some different ways. Such a system that involves a set of agents that can potentially interact with each other is called a multi-agent system (MAS) [Ron07].

In general, machine learning includes adaptive mechanisms such artificial neural networks which make computers learn by example; an intelligent system performance can be enhanced by learning capabilities over time [Mic05]. An essential ability of artificial neural networks is learning. Rules of learning are algorithms for computing proper weights [kel14].

The classification is the task of assigning labels to a set of instances in such a way that instances with the same label share some common properties and logically belong to the same class. Based on whether a group of labeled training patterns exists or not, classification can be either supervised or unsupervised. When there are examples to training the classification named supervised, it is the usual way in which a child learns–through the examples to training [San13].

2.2 The Multi Agent Technology

AI is touching practical reasoning, the reasoning to do something. The perception and reasoning, then acting comprise an agent. An agent may be, for illustration, be a robot, a person, a worm, a dog or a computer program that buys and sells. An agent's environment may well involve other agents. An agent together with its environment is called the world. Figure 2.1 shows the inputs and outputs of an agent.

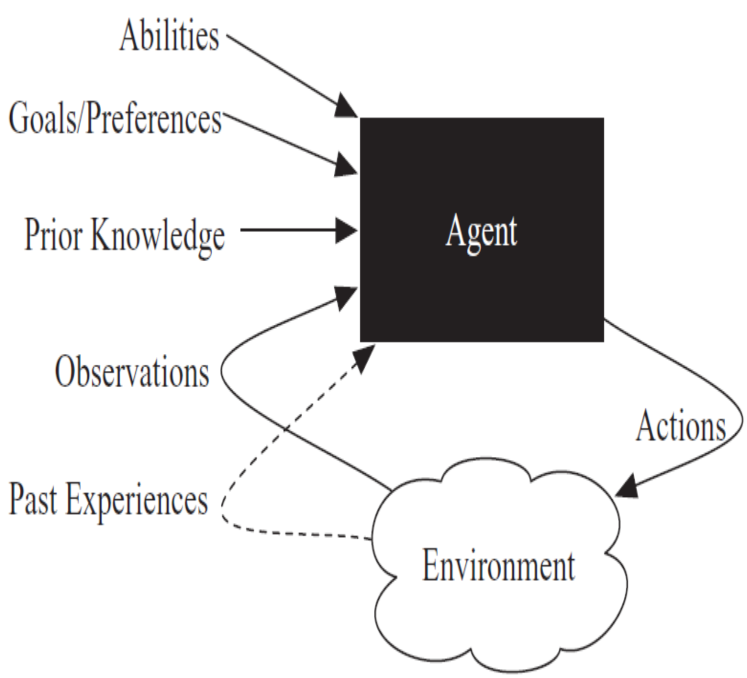


Figure 2.1: An agent interacting with an environment [Dav10].

An agent system is made up of an agent and its environment. The agent receives motives from the environment and carries out actions in the environment [Dav10].

Multi-agent systems have become a chosen paradigm to model and solve real-world problems [Jia16]. Some of the benefits of MAS technology in large systems are speedup and efficiency, due to the asynchronous and parallel computation [Ron07].

2.2.1 Characteristics of multi agents systems

The fundamental aspects that characterize MAS and distinguish it from a single agent system are [Ron07]:

- *Agent Design*: The design may involve the hardware, software.
- *Environment:* Static or Dynamic.
- *Perception:* observe data that differ spatially, temporally, semantically.
- *Control:* the control in MAS is typically decentralized. Decentralized control is preferred over centralized control (that involves a center) for reasons of robustness and fault-tolerance.
- *Communication:* Communication can be made for coordination among cooperative agents or for negotiation among self-interested agents.

2.3 Artificial Neural Network

Artificial neural networks (ANNs) are systems for information processing that have certain computational properties similar to those which have been assuming for biological neural networks. Artificial neural networks represent the arithmetical models of the neural biology which produce human knowledge. [Far00, Ken11, Muh13].

It consists of many simple units, named neurons, are interrelated by weighted links into bigger structures of important good performance [Mir15, Tos08]. ANNs can learn where they use knowledge to improve their performance. When exposed to an enough number of training patterns, it can generalize to others they have not yet seen. They can recognize terms in human speech, discover explosives at airports and distinguish handwritten characters [Mic05].

The artificial neuron is a simple mathematical model (function), which considered an essential building block of every ANN; three simple sets of rules formed a neural network model: activation, summation, and multiplication. Each input of artificial neuron multiplied by the individual weight. After that sum, function sums bias and all weighted inputs. And they pass through activation function (transfer function) [Ken11]. Figure 2.2 show the general view of neural network.

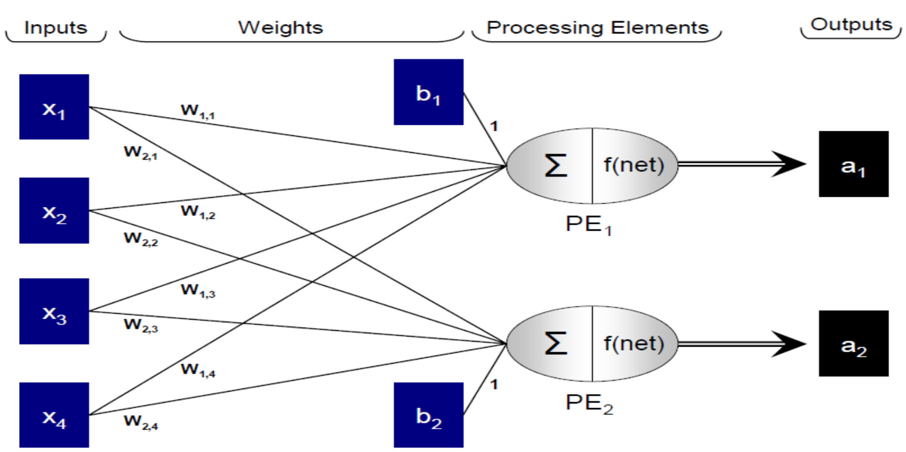


Figure 2.2: Artificial neural network [Sub08].

The neural network designed to accept input and then conclude output. The pattern recognition capabilities represent the real power of a neural network. The neural network should be able to make the desired output even if the input has been a little fuzzy [Jef11]. The architecture of the neural network is significant for performing a computation. Some neurons are set to take inputs from the environment. These neurons are not joined with each other, so the arranging of these neurons is in a layer, named as Input layer. The output of input layer represents the input to next layer. The architecture of ANN can be of multilayer or the single layer. In the single layer neural network, one output layer and one input layer is there, while in multilayer neural network, there can be one or further hidden layer [Ami11, Ken11]. Hidden layers purpose is to give the neural network compatible output with the given input. The difficulty is detecting the number of neuron for the hidden layer. The challenge is to keep away from building a hidden structure that is either too simple or too complex. Too simple will not learn the problem. Too complicated will take too long time to train. An excellent starting point is a single hidden layer [Jef11]. There is presently no theoretical cause to use neural networks with any further than two hidden layers. Also, there's no motivation to use any more than one hidden layer for a lot of practical problems [Ami11].

Earlier studies did not show any enhancement of multiple hidden layers more than single hidden layer systems. The theorem that one hidden layer is sufficient to assign any non-linear functional relationship with a practical level of accuracy, also computational time increasing with too many hidden layers without increasing the accuracy much. So, only a single hidden layer is better [Ang09].

Also performance of classification depends on the structure of the network, with Back propagation learning algorithm. When increase hidden layers then the patterns misclassified will increase [Pre12].

2.3.1 Number of Hidden Units

Different parameters of neural network should be adequately selected when building a neural network. Among these parameters is the number of layers, number of nodes per layer, iterations of training. The most significant parameter regarding network ability and training is the number of hidden nodes [Hug15].

When the number of hidden units is more than necessary, this leads to long training time and while accurate classifications in the training data increases, however, the solution does not necessitate carrying out well classification with test data. That means, the network memorizes the training set patterns and does not generalize this to unseen data [Sub08].

2.3.2 Adding Bias Unit

Except for the output unit, the bias can be added to every layer. This unit has a fixed value of 1 and it is linked to all units in the following layer. The weights on these links can be trained such as other weights. The bias units give a constant term in the sum of the unit's weights in the next layer. The consequence sometimes improves the performance of the network [Oma10, Muh13].

Biases and initial weights, activation function, learning rate value, should be selected carefully. An improper selection of these parameters can guide to network error, failure, or slow network [Mzr11].

2.4 Learning Neural Network Models

Learning algorithms are used for finding suitable weights (and/or) additional network parameters [Kel14]. After preparing inputs of neural network models, the neural network models will be trained with these inputs. The training is performed using appropriate learning algorithm. The three-layer feed forward ANN is the commonly used ANN model and learning with Back Propagation Method [Kha08]. Learning methods are usually divided into unsupervised, reinforcement and supervised learning; unsupervised learning schemes are generally used for data analysis, signal coding, feature extraction, vector quantization, and clustering. Reinforcement learning is usually used in artificial intelligence and control. Supervised learning is broadly applied in optimization, modeling, control, identification and approximation, signal processing, and classification [Kel14].

Supervised learning is change network weights by a direct comparison between the desired and actual output of the network. Supervised learning is feedback system, where the error represents the feedback signal. The measure of error, which is the difference between the output from the training patterns and the network output, is used to direct the learning process. The error measure is often concluded by the mean squared error (MSE). The error is determined after each epoch. In supervised learning, neural networks are often trained by epoch.

An epoch is a full run when all the training patterns are offered to the network and only once the learning algorithm processed training patterns. After learning, a neural network gives a complex relationship. Terminating the learning process is when a failure criterion is met, or error is small enough. To decrease near to zero, a gradient-descent procedure is used. The back propagation algorithm is the most popular gradient-descent based algorithm [Kel14, Shi10].

2.5 Activation Functions

A back propagation is an adaptive network whose neurons (or nodes) execute the same function on input signals; this neuron function or activation function or "transfer function" or sometimes called a step-function is often a composite of the weighted sum and a differentiable nonlinear activation function [Pat05][Dev08].

There are different activation functions, the most usually used activation functions are summarized in Table (2.1).

Table 2.1: Neuron Activation Functions [Pat05].

Activation function	Formula a= f(n)	Derivatives d f(u)/d(u)
Sigmoid	$f(u)=1/1+e^{-u}$	$f(n)[1-f(n)]$
Hyperbolic tangent	$f(u)=\tanh(u)$	$1-[f(u)]^2$
Linear	$f(u)= au + b$	a
Threshold	$f(u)=\begin{cases} 1 & u > 0; \\ -1 & u < 0. \end{cases}$	Derivatives do not exist at u=0

The sigmoid function is the most usually used. Sigmoid function has simply calculated derivate, which can be significant when determining updates of the weight in the artificial neural network [Yuh02].

2.6 Back Propagation Algorithm

Back-propagation is the most popular algorithm for multilayer networks learning. It was created by Bryson and Ho in 1969 [Stu95]. This algorithm cycles via two different passes, a forward pass followed by a backward pass to all layers of the network. The algorithm alternates between these cycles many times to all the training data.

Forward Pass: calculation of outputs of all network's neurons.

- The algorithm begins with the first hidden layer using inputs of pattern from the training data.
- The outputs are determined for all neurons in the first hidden layer by executing the related sum and activation function evaluations.
- These outputs represent inputs for neurons in the second hidden layer if found. Again the related sum and activation function results are executed to determine the outputs of second layer neurons.

Backward pass: error propagation and weights adjustment

- This phase starts with the calculations of error at every neuron in the output layer. Common error function is the squared difference between the output of node and the goal value for the node.

- The goal value is just 1 for the output neuron matching to the class of the pattern and zero for other output neurons.
- The new weight w_{jk} of the links from node j to node k is given by:

  New w_{jk}= old w_{jk} + $\eta o_j \boldsymbol{\delta}_k$ Here η is a significant tuning parameter that is selected by trial and error by frequent runs on the training data set. Usually values for η are not larger than one.
- The backward propagation continues until it arrived at the input layer to adjustments all weights along these lines.
- At this time will get a new collection of weights on which will run a new forward pass when presented with a training data [Ami11].

The BP algorithm designed to reduce the mean square error between the desired output and the actual output of multi-layer feed forward perceptron.

The BP algorithm steps [Ins08] [Dev08] [Flo10]**:**

Step1: *Initialize the weights*

Initial all weights to small random values.

Step2: *Present desired outputs and input*

Present a continuous valued input vector $\mathbf{x_0, x_1 \dots x_{N-1}}$ and select the target output $\mathbf{d_0, d_1 \dots d_{N-1}}$. If the net is used as a classifier then all target outputs are set to zero except for that related to the class the input is from. That target output is 1. The input will be new on each trial or patterns from a training set could be offered cyclically until stabilize.

Step 3: *actual output calculated*

Use the sigmoid non linearity to calculate output $\mathbf{y_0, y_1 \dots y_{N-1}}$.

Step 4: *weights adapted.*

Use a recursive algorithm begins at the output neurons and running back to the first hidden layer.

$$\text{Adjust weights by } w_{ij}(t+1) = w_{ij}(t) + \eta\, \delta_j X_i \qquad (2.1).$$

In this equation $w_{ij}(t)$ is the weight from hidden node i or from an input to node j at time t, X_i is either the output node i or is an input, η is a gain term, and δ_j is an error term for node j, if node j is an output node, then

$$\delta_j = y_j (1 - y_j)(d_j - y_j) \qquad (2.2).$$

Where d_j is the target output of node j and y_j is the actual output. If node j is an internal hidden node, then

$$\delta_j = x_j (1 - x_j) \Sigma\, \delta_{jm} w_{jk} \qquad (2.3).$$

Where, k is over all nodes in the layers above node j.

Step 5: *Repeat by going to step 2.*

2.7 Mode of Error Calculation

The error can be concluded in the batch mode or pattern mode. In batch mode, all the patterns are processed, and the errors are determined for each pattern and then aggregate square error is used to adapt the weights, in pattern mode of error determination, the error is calculated after processed each pattern which is used to modify the weights [Dev08, Jye99].

Pattern mode is usually faster than the batch mode, particularly for large training sets, how much faster pattern mode is based on both the features of the application and the size of the training set [Ran03].

2.8 The Generalization

Generalization refers to estimating the value of correct outputs where there is no example. While Learning is a weights reconstruction depending on existing patterns. Mathematically, the process of learning can be considered a nonlinear curve-fitting process, while generalization can be considered the extrapolation and interpolation of the input data. When a network is over trained with too many epochs, parameters or examples, it may generate excellent results for the training data, but has a bad generalization capacity. That is what is called the over fit phenomenon. In general, the generalization capacity of a network is jointly determined by the complexity of the problem, the size of the training pattern set and the network architecture. The set of training should be suitably large and varied that to represent the problem fine. For good generalization, the training set size, N must be at least many times larger than the network's capacity [Kel14, Kld06].

2.8.1 Generalization by Stopping Criterion

Generalization can be measured during training; overtraining can be limited by stopping the training before reach to the absolute minimum. When training is stopped at an appropriate point, the network will not learn the high-frequency noise. While the training error will always decrease, the generalization error will decrease to a minimum and then starts to rise again as the network is being over trained. Training should stop at the optimum stopping point [Kld06].

The generalization error is defined in the same form as the learning error, but on a separate validation set of data. Early stopping is the default method for improving generalization.

For example, to use early stopping technique, the available data is divided into three subsets. The first subset is the training set; the second is the validation set, the validation set's error is observed during the training process.

The validation error usually decreases during the initial phase of training, as does the error on the training set. However, when the network begins to fit the data over, the error on the validation set starts to grow. When the validation error grows for a particularized number of iterations, training is stopped, and the biases and weights at the minimum of the validation error are returned.

The error on the test set is not used through training, but it is used to compare various models, it is also useful to plot the error on the test set during the training process [Kel14].

2.8.2 Generalization by Regularization

Regularization is a reliable method for improving generalization. The target function is assumed to be smooth, and small changes in the input do not cause substantial changes in the output.

The regularization can be done by network-pruning techniques such as the weight-decay technique also help to improve generalization, where at the end of the training there are some weights significantly varied from zero, while some other weights are close to zero. Those connections with small weights can remove from the network. Biases should be excluded from the penalty term so that the network yields an unbiased estimate of the true target mean.

Also, regularization can be done by training with a small amount of jitter in the input while keeping the same output can improve generalization. With jitter, the learning problem is equivalent to a smoothing regularization with the jitter variance playing the role of the regularization parameter.

Training with jitter thus allows regularization within the conventional layered feed-forward network architecture. Although large networks are generally trained rapidly, they tend to generalize poorly due to insufficient constraints, training with jitter helps to prevent over-fitting.

Jitter is added to the current training set to create an unlimited source of training samples. This is described as a kernel estimate of the probability density that describes the training vector distribution.

It helps to improve the generalization performance, speed up the BP, and reduce the possibility of local minima entrapment [Kel14, Shi10, and Kld06].

2.9 Model Selection

There is principle says: “No more things should be presumed to exist than are necessary.” That is, if two models of different complexity fit the data approximately equally well, the simpler one usually is a better predictive model. From models approximating the noisy data, the ones that have the smallest complexity should be taken. The objective of model selection is to find a model that is as simple as possible that fits a given dataset with sufficient accuracy and has a good generalization capability to unseen data.

The generalization performance of a network provides a measure of the quality of the chosen model. One of the most model selection approaches is cross-validation; in cross validation method, many networks of different complexity are trained and then tested on an independent validation set. Cross-validation is a standard model-selection method.

The total pattern set is randomly partitioned into a training set and validation (test) set. The major part of the total pattern set is included in the training set, which is used to train the network. The remaining, typically, 10–20%, is included in the validation set and is used for validation.

The popular K-fold cross validation employs a non overlapping test set selection scheme. The data is divided into K non overlapping data subsets of the same size. Each data subset is then used as a test set, with the remaining K – 1 folds acting as a training set, and an error value is calculated by testing the classifier in the remaining fold. Finally, the K-fold cross validation estimation of the error is the average value of the errors committed in each fold. Thus, the K-fold cross validation error estimator depends on two factors: the training set and the partitioning into folds. Cross-validation methods split the data such that a sample does not appear in more than one validation set [Kel14, Kld06].

2.10 Fault Tolerance and Generalization

Fault tolerance is robustly associated with generalization. Input noise during training enhances generalization capability, and synaptic noise during training improves fault tolerance. When fault tolerance is improving, the generalization capacity is usually improved, and vice versa [Kld06, Kel14].

2.11 Data Normalization

Normalization is the important part in the testing and training of neural networks, normalization is the very significant concern in an artificial neural network [Dev08].

Artificial neural network training could be given additional efficient by doing confirmed preprocessing steps on the inputs and targets of the network. The input of network processed by transforms functions into improved form for the network use. The process of normalization for the raw inputs has a big effect on data preparing to be appropriate for the training. With the normalization, training the neural network would have been faster.

There are several kinds of data normalization. It used to scale the data in the same domain of values for each input trait to minimize bias within the neural network between features. The training time can be improved by convert data to normalization form [Taj11].

The Min-Max Normalization method is applied as fallowing:

$$\text{New F} = \frac{dF - MinF}{MaxF - MinF} \qquad (2.4).$$

Where:

dF: is the feature value.

$MinF$: is the minimum value that the feature F can get.

$MaxF$: is the maximum value that the feature F can get.

New F: is the normalize value of feature F [Han14].

2.12 Fuzzy logic

Fuzzy logic is a length of Boolean logic by L. Zadeh in 1965 based on the mathematical theory of fuzzy sets, which is an extent of the classical set theory. By introducing the notion of the degree in the verification of a condition, thus enabling a condition to be in a state other than false or true, fuzzy logic gives a very important flexibility for reasoning, which makes it possible to take into account inaccuracies and uncertainties [Fra13].

2.12.1 Fuzzy Sets

A fuzzy set is a set that all members have grades of membership between 1 and 0, it is different from classical sets where each member must have 1 or 0 as the membership grade —if 1, the member is fully in the set; if 0, the member is fully outside the set. As fuzzy logic is associated with the theory of fuzzy set, classical logic is associated with the theory of classical set. By using fuzzy set theory can get precise representations of relations and concepts that are ambiguous, that mean, with no sharp no-yes limit lines between situations covered, and situations not covered, by the relation or concept [Lui15, Tim10, Jye99].

2.12.2 Membership Functions

Functions of membership allow the notion of a class to be extended to categories that don't have clear-cut borders; the functions of membership are one of the representations of fuzzy sets. The idea of the membership function is vital in any fuzzy set-based scheme

[Lui15] [Tim10] [Jye99]. The membership functions most generally used in practice are:

I. Triangles membership function

It used three parameters {a, b, c} as shown in figure 2.3.

$$\text{Where triangle } (x: a, b, c) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \le x \le b \\ \frac{c-x}{c-b} & b \le x \le c \\ 0 & x > c \end{cases} \quad (2.5).$$

x: original feature's value.

a: minimum feature's value.

c: maximum feature's value.

b: average a and c.

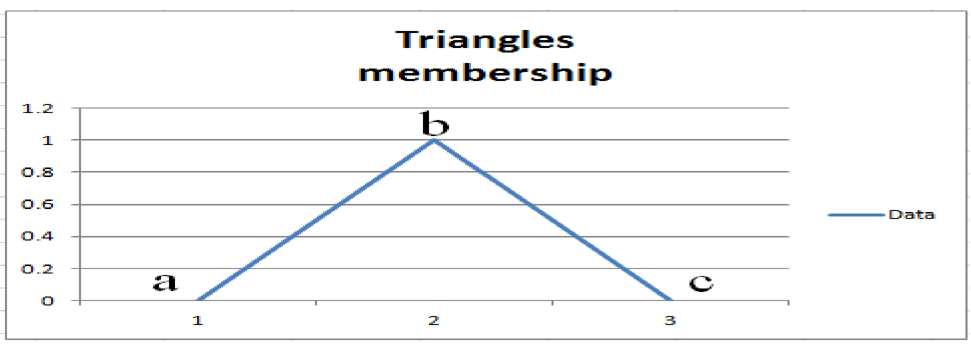


Figure 2.3: The triangles membership [Jye99].

II. Bell curves

It used three parameters {a,b,c} as shown in figure 2.4 .

$$\text{Where bell}(x: a, b, c) = \frac{1}{1+|(x-b)/a|^{2c}} \quad (2.6).$$

x: original feature value.

a: minimum feature value.

c: maximum feature value.

b: average a and c.

The parameter c is often positive where (b and a) can be adjusted to vary the center and width of the function and then use c to control the slopes [Jye99].

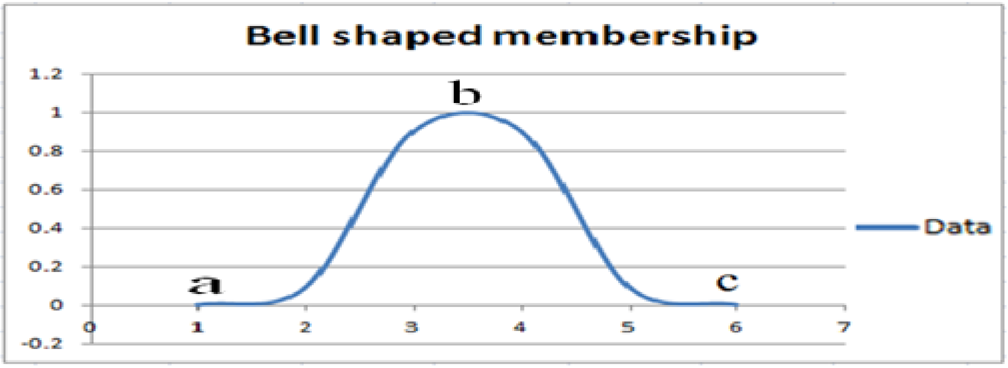


Figure 2.4 Bell shaped membership [Jye99].

2.13 Weka software

Weka (Waikato Environment for Knowledge Analysis) is popular machine learning software programmed in Java, produced at the University of Waikato, New Zealand. It is free software licensed following the General Public License. It contains a collection of visualization tools and algorithms for data and predictive modeling, together with graphical user interfaces for simple access to these functions. It used in many different application areas, in particular for educational purposes and research. Weka maintains several standard data mining tasks, more specifically, data preprocessing, clustering, classification. A set of data items (dataset) is an essential idea of machine learning. A dataset is approximately equivalent to a two-dimensional spreadsheet or database table. In WEKA, A dataset is a collection of Instances. Each Example consists of some traits, any of which can be numeric (= a real or integer number), nominal (= one of a predefined list of values) or a string (= an arbitrarily long list of characters, enclosed in "double quotes"). There are four test forms [Rem16]:

1. ***Use training set***. The classifier is assessed on how well it predicts the class of the instances it was trained on.

2. ***Supplied test set***. The classifier is assessed on how well it predicts the class of a set of examples loaded from a file. Clicking the Set... button brings up a dialog enabling you to choose the file to test on.

3. ***Cross-validation***. The classifier is assessed by cross-validation, using the number of folds that are entered in the Folds text field.

4. ***Percentage split***. The classifier is assessed on how well it predicts a certain percentage of the data which is held out for testing. The number of data held out depends on the value entered in the % field.

2.13.1 Steps to apply feature selection and classification techniques on data set to get results in Weka [Anu15]:

Step1: Take the input dataset and open it from preprocess tab.

Step2: Go to the Select Attribute tab and choose cfsSubsetEval as Attribute Evaluator and Genetic Search as the search method. This will perform feature selection or dimension reduction.

Step3: Checkmark only those traits which are selected by cfsSubsetEval and Genetic Search. Remove rest of the features.

Step4: Apply the classifier algorithm on the whole data set.

Step5: Note the accuracy given by it and time required for execution.

Step6: Repeat steps 2, 3, 4 and 5 for different classification algorithms on various datasets.

Step 7: Compare the different accuracy provided by the dataset with different classification algorithms and identify the significant classification algorithm for particular dataset.

2.13.2 Performance Measures

Weka's Performance Measures of accuracy for classification are the following [Anu15]:

1. ***Accuracy Classification***: all classification result could have an error rate and it may fail to classify correctly. So accuracy can be calculated as follows:

   Accuracy= Instances Correctly Classified/Total Number of Instances)*100 %.

2. ***Mean Absolute Error (MAE)***: MAE is the average of the difference between the predicted and actual value in all test situations. The formula for determining MAE is given in equation shown below:

   $$MAE = (|a_1 - c_1| + |a_2 - c_2| + \ldots + |a_n - c_n|) / n \qquad (2.7).$$

   Here 'a' is the actual output and 'c' is the expected output.

3. ***Confusion Matrix***: A confusion matrix includes information about actual and predicted classifications done by a classification system.

4. ***Time Complexity***: Time taken to execute the code of each classifier is calculated using:

   long b = System. Current Time Millis(); at the starting of code and

   long a =System. Current Time Millis(); at the ending of code.

   Finally printing the time taken by calculating the difference between a and b using System. out. println(a-b).

The classification accuracy, time complexity, means, absolute error and confusion matrices are determined for each machine learning algorithm.

5. ***The accuracy by a class array [Rem16]:***

- The True Positive (TP) rate is the ratio of examples which were classified as class c, among the entire examples which truthfully have class c, it is equal to Recall.

$$(TP)\ rate = TP/Actual\ c \qquad (2.8).$$

- The False Positive (FP) rate is the ratio of examples which were classified as class c, but in the fact it belongs to another class, among all examples which are not of class c.

$$(FP)\ rate = FP/Actual\ not\ c \qquad (2.9).$$

- The Precision is the ratio of the examples which truthfully have class c among all those which were classified as class c.

$$Precision = TP/predicted\ as\ c \qquad (2.10).$$

2.14 Data Sets Information

Four datasets (User Knowledge Level, Iris, Banknote Authentication and seeds) used for training and testing the system each one of these datasets are allocated to one classification agent. The number of features in each data set represents inputs for the corresponding classification agent and number of classes in the dataset equals to the outputs of the corresponding classification agent.

2.14.1 User Knowledge Level Dataset

It is in the area of education and associated with classification and clustering tasks, it is include 403 patterns, the number of attributes is five and number of classes is four, it is related to the students' knowledge which represents in five features for each user (pattern) these features are illustrated in table 2.2. An example of this dataset is shown in table 2.3, the behavior of each feature in all classes will be shown in the figures 3.1, 3.2, 3.3, 3.4 and 3.5.

Table 2.2: Features of user knowledge level dataset.

features	Meaning	range	
STG	degree of study time for goal object materials	[0, 0.99]	
SCG	degree of repetition number of user for goal object materials	[0, 0.90]	
STR	degree of study time of user for related objects with goal object	[0, 0.95]	
LPR	exam performance of user for related objects with goal object	[0, 0.99]	
PEG	exam performance of user for goal objects	[0, 0.99]	
UNS	numbers of classes in this dataset belong to four classes which represent knowledge level of the user	Very Low: Low: Middle: High:	50 (patterns). 129 (patterns). 122 (patterns). 102 (patterns).

Table 2.3: Examples of the user knowledge level dataset.

STG	SCG	STR	LPR	PEG	UNS
0.08	0.08	0.1	0.24	0.9	*High*
0.1	0.1	0.15	0.65	0.3	*Middle*

0.06	0.06	0.05	0.25	0.33	*Low*
0.3	0.5	0.1	0.3	0.15	*very low*

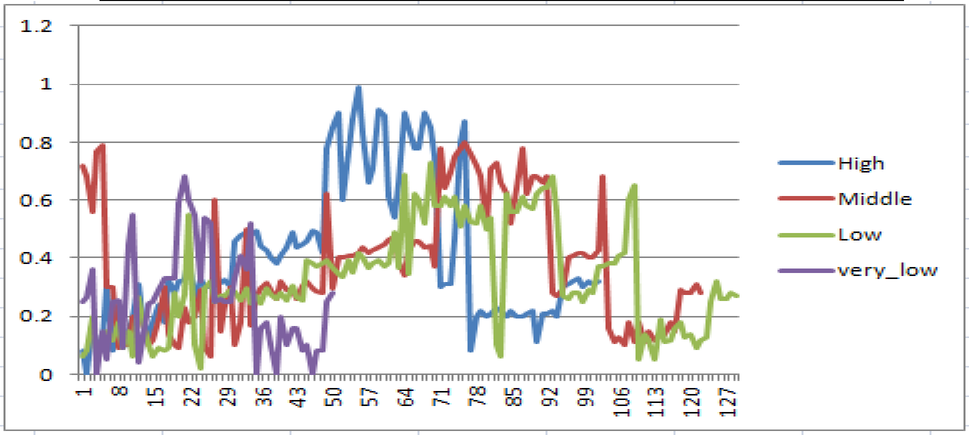


Figure 2.5: behavior of STG feature in all classes.

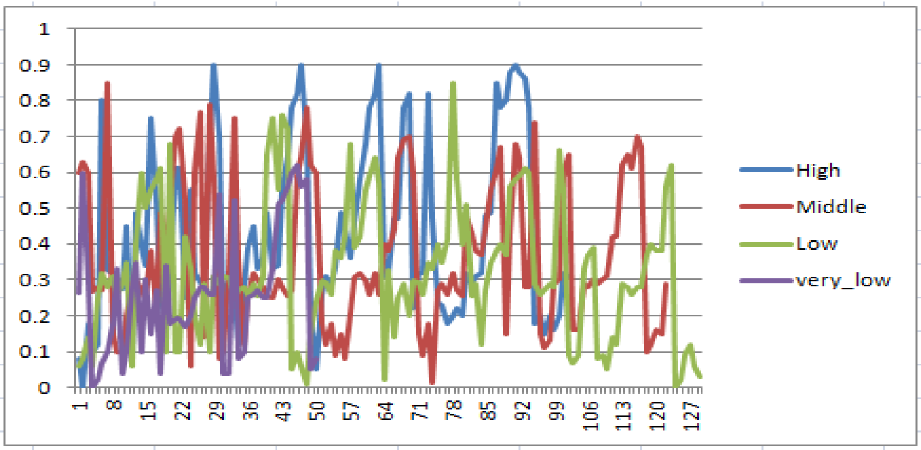


Figure 2.6: behavior of SCG feature in all classes.

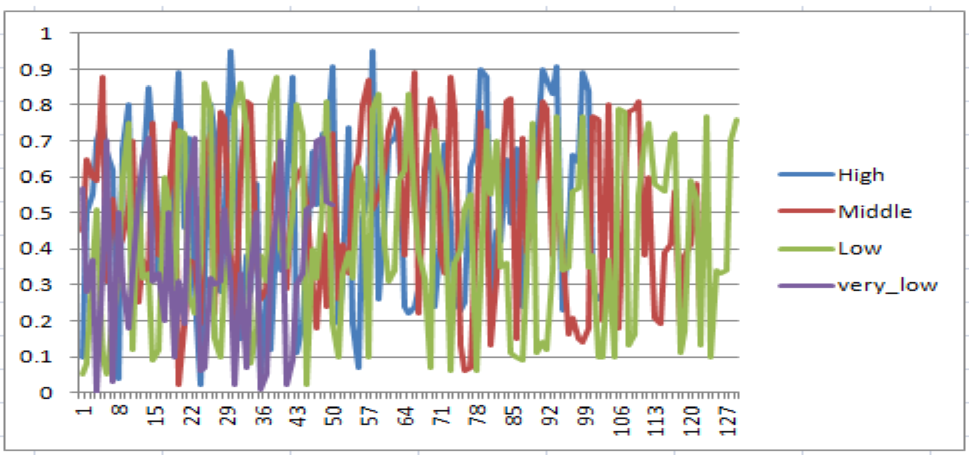


Figure 2.7: behavior of STR feature in all classes.

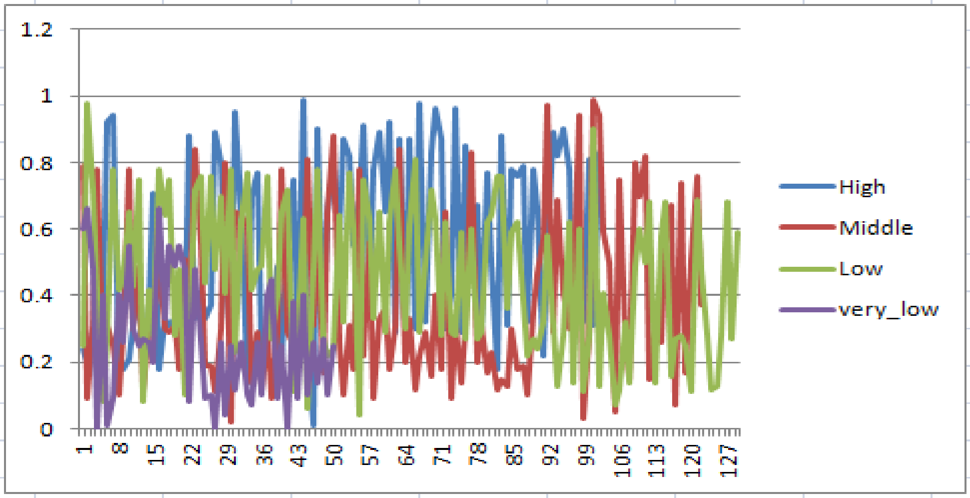


Figure 2.8: behavior of LPR feature in all classes.

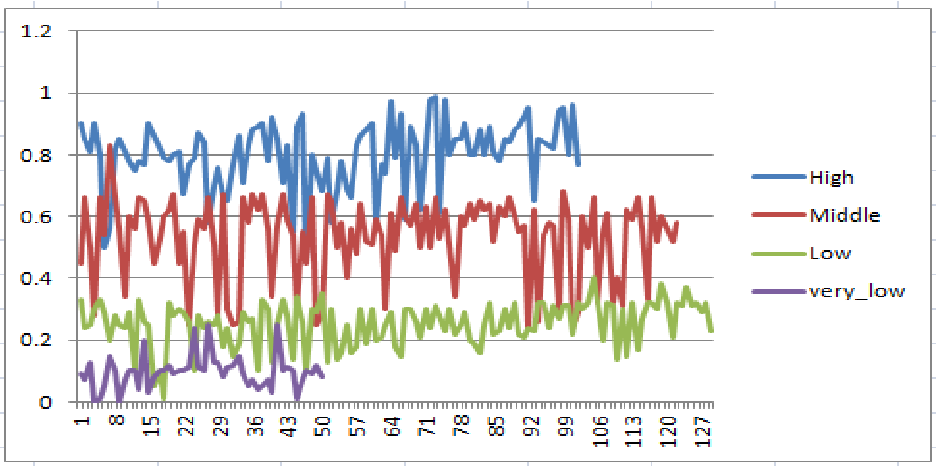


Figure 2.9: behavior of PEG feature in all classes.

2.14.2 Iris Dataset

It is in the area of life and associated with classification task, it includes 150 patterns, the number of attributes is four and number of classes is three, this dataset related to the iris flowers which represents in four features for each class (pattern). These features (input values) are illustrated in table 2.4. The number of classes in this dataset (target value) is three classes represent iris types:

- Iris Setosa (50 patterns).
- Iris Versicolour (50 patterns).

- Iris Virginica (50 patterns).

Table 2.4: Features of iris dataset.

Features	Meaning	range
S length	Sepal length in cm	[4.3, 7.9]
S width	Sepal width in cm	[2, 4.4]
P length	Petal length in cm	[1, 6.9]
P width	Petal width in cm	[0.1,2.5]

An example of this dataset is shown in table 2.5, while the behavior of each feature in all classes is shown in figures 2.10, 2.11, 2.12 and 2.13.

Table 2.5: Examples of the iris dataset.

slength	swidth	plength	pwidth	class
5.1	3.5	1.4	0.2	*Setosa*
6.2	2.2	4.5	1.5	*Versicolour*
6.3	3.3	6	2.5	*Virginica*

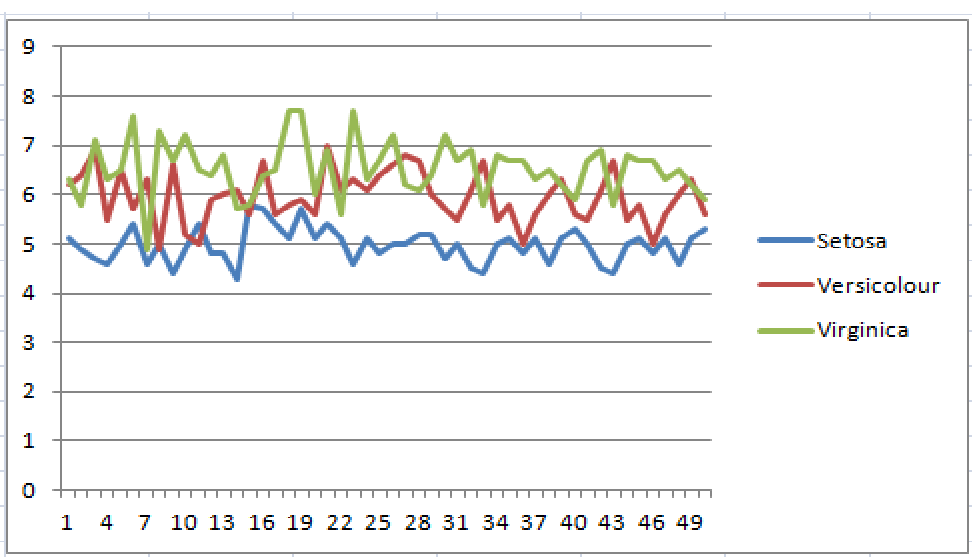


Figure 2.10: Behavior of slength feature in all classes.

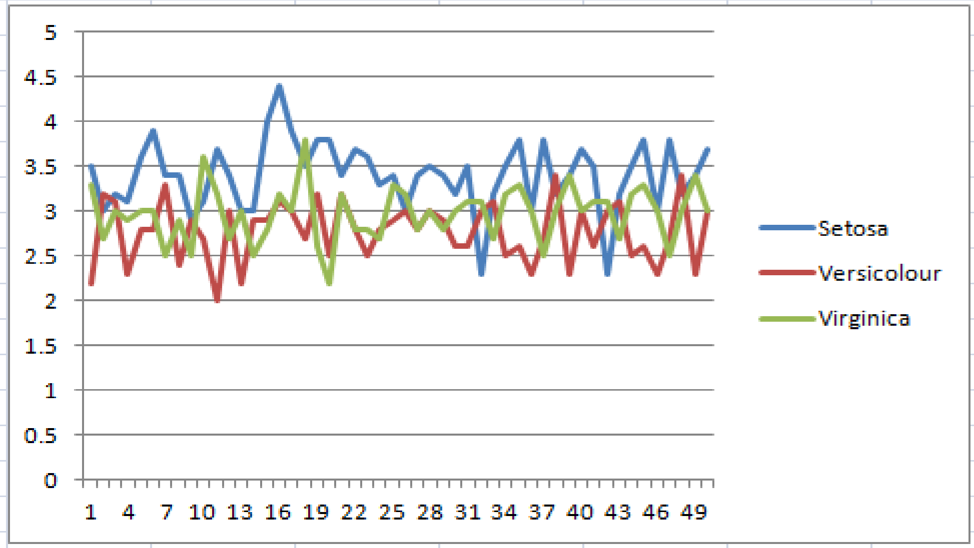


Figure 2.11: Behavior of swidth feature in all classes.

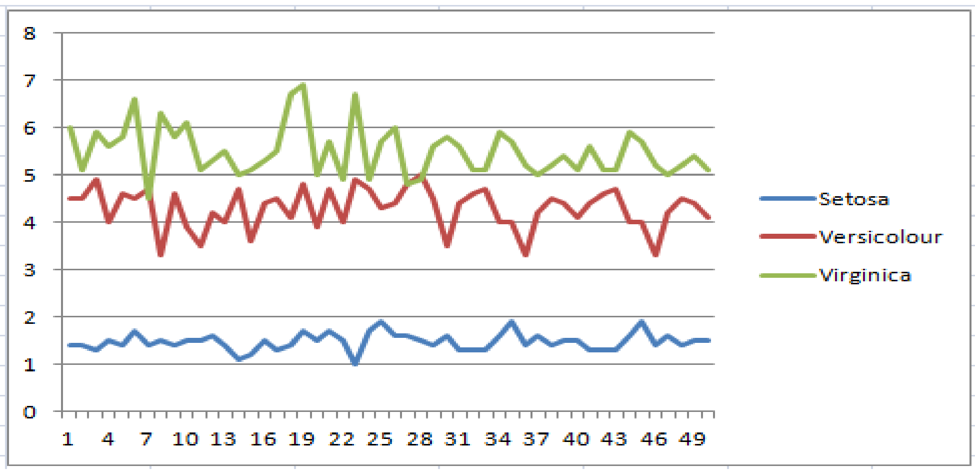


Figure 2.12: Behavior of plength feature in all classes.

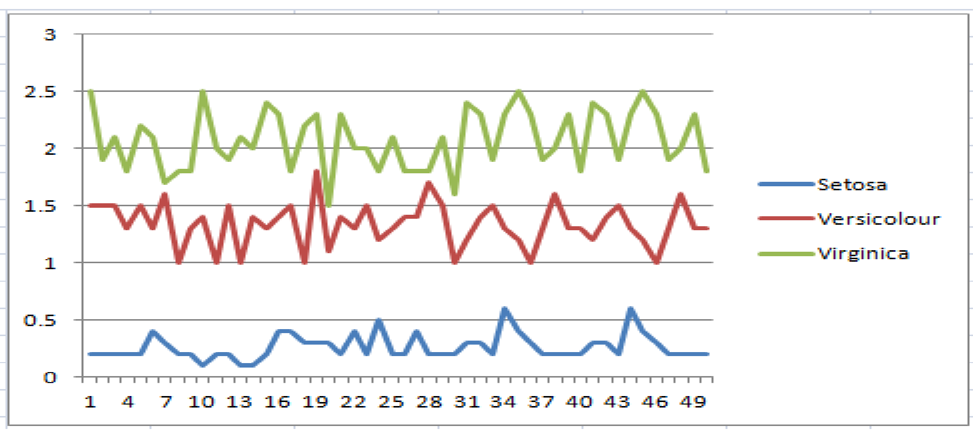


Figure 2.13: Behavior of pwidth feature in all classes.

2.14.3 Banknote Authentication Dataset

It is in the area of computer and associated with classification task, it include 1372 patterns, the number of attributes is four and number of classes is two, this dataset related to the authentication of banknote which represents in 4 features for each banknote (pattern), these features (input values) are illustrated in table 2.6. An example of this dataset is shown in table 2.7. The behavior of each feature in all classes is shown in figures 2.14, 2.15, 2.16 and 2.17.

Table 2.6: Features of banknote dataset.

features	Meaning	range
F1	Variance of wavelet transformed image (continuous)	[-7.0421, 6.8248]
F2	Skewness of wavelet transformed image (continuous)	[-13.7731, 12.9516]
F3	Curtosis of wavelet transformed image (continuous)	[-5.2861, 17.9274]
F4	Entropy of image (continuous)	[-8.5482, 2.4495]

The numbers of classes in this dataset (target value) are two classes represent the authentication of banknote (integer value):

- 1 (unauthenticated).
- 0 (authenticated).

Table 2.7: Examples of the banknote authentication dataset.

F1	F2	F3	F4	F5
3.6216	8.6661	-2.8073	-0.44699	*0*
-1.3971	3.3191	-1.3927	-1.9948	*1*

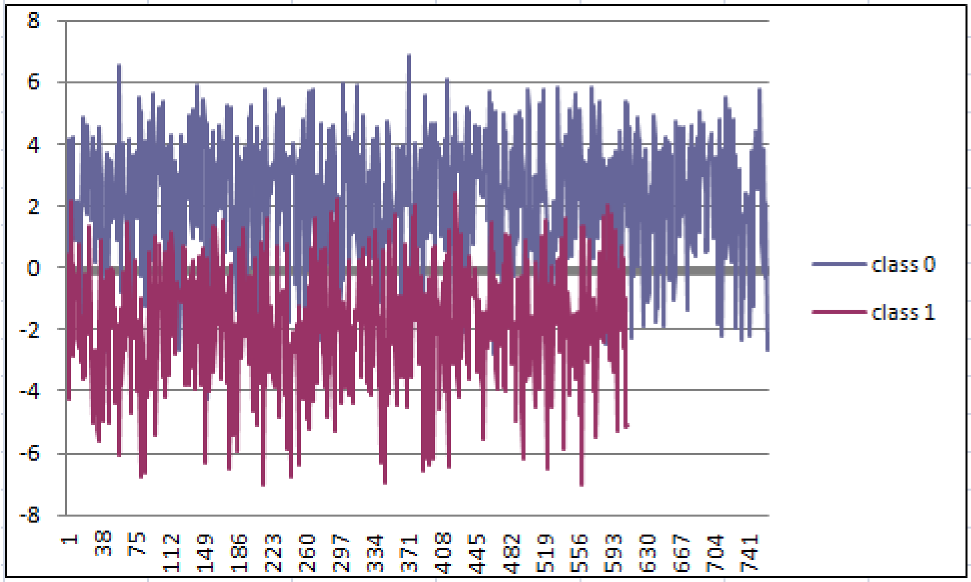


Figure 2.14: Behavior of f1 feature in all classes.

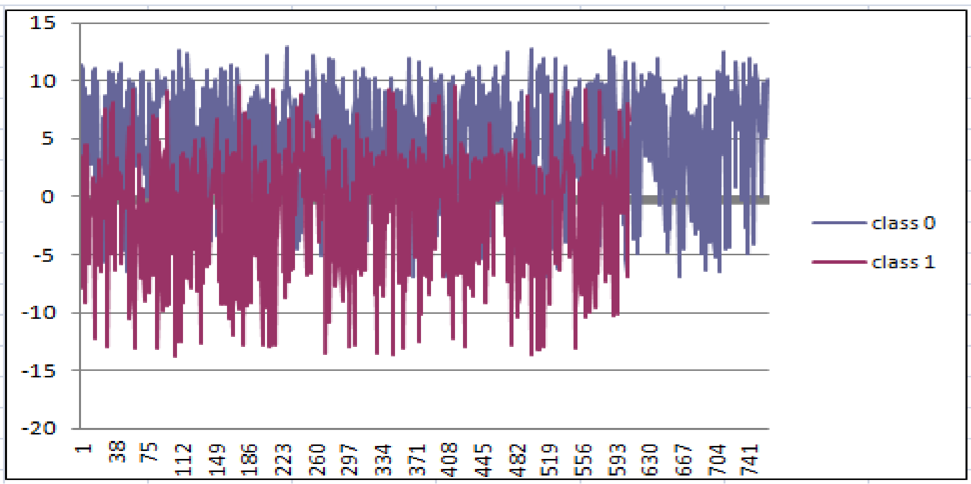


Figure 2.15: Behavior of f2 feature in all classes.

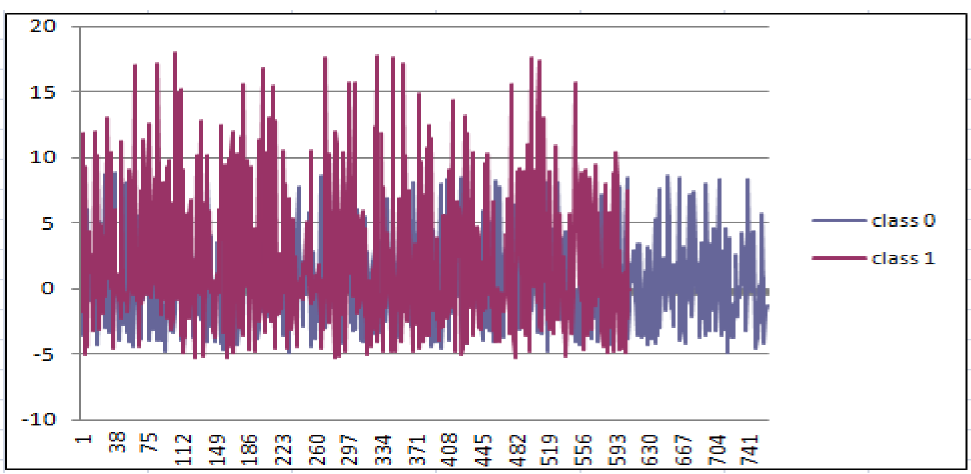


Figure 2.16: Behavior of f3 feature in all classes.

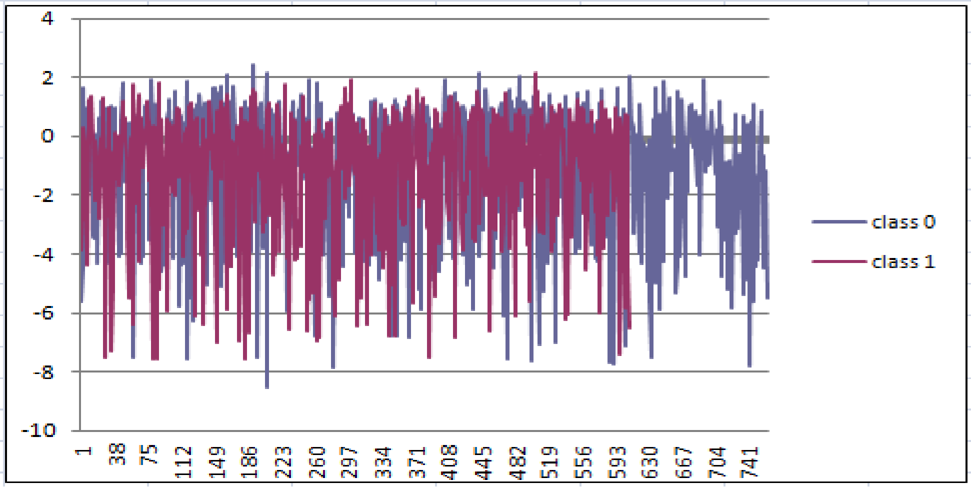


Figure 2.17: Behavior of f4 feature in all classes.

2.14.4 Seeds Dataset

It is in the area of life and associated with classification and clustering tasks, it include 210 patterns, the number of attributes is seven and number of classes is three, this dataset related to Measurements of geometrical properties of kernels belonging to three different varieties of wheat which represents in 7 features for each kernel (pattern), these features (input values) are illustrated in table 2.8. An example of this dataset is shown in table 2.9. The behavior of each feature in all classes is shown in figures 2.18, 2.19, 2.20, 2.21, 2.22, 2.23 and 2.24.

Table 2.8: Features of seeds dataset.

features	Meaning	range
F1	area A	[10.59, 21.18]
F2	perimeter P	[12.41 , 17.25]
F3	compactness C = 4*pi*A/P^2	[0.8081, 0.9183]
F4	length of kernel	[4.899 , 6.675]
F5	width of kernel	[2.63,4.033]
F6	asymmetry coefficient	[0.7651,8.456]

F7	length of kernel groove	[4.519,6.55]

The numbers of classes in this dataset (target value) are three classes represent the kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian (real value):

- 1 (Kama).
- 2 (Rosa).
- 3(Canadian).

Table 2.9: Examples of seeds dataset.

F1	F2	F3	F4	F5	F6	F7	CLASS
14.46	*14.35*	*0.8818*	*5.388*	*3.377*	*2.802*	*5.044*	*1*
18.65	*16.41*	*0.8698*	*6.285*	*3.594*	*4.391*	*6.102*	*2*
12.79	*13.53*	*0.8786*	*5.224*	*3.054*	*5.483*	*4.958*	*3*

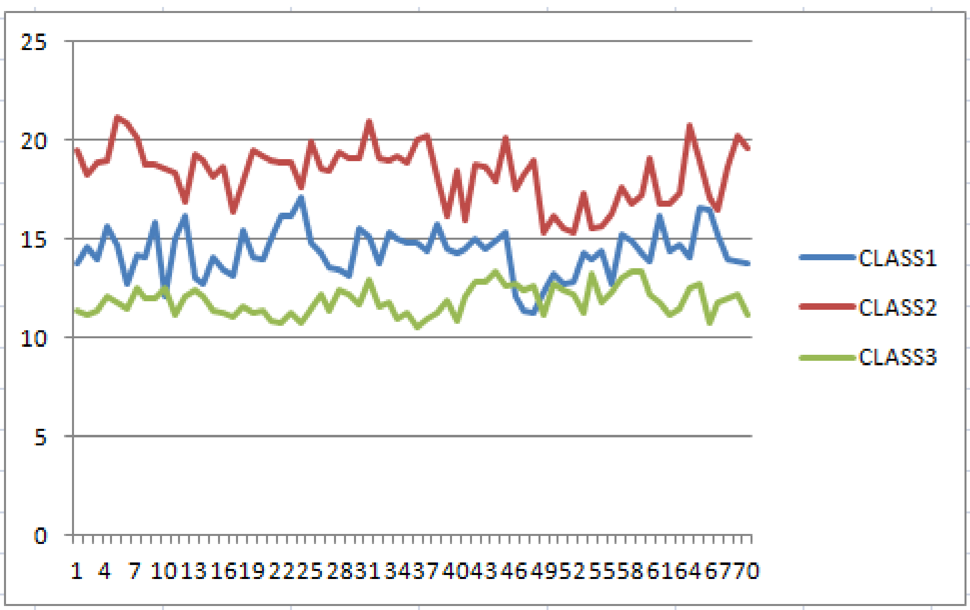


Figure 2.18: Behavior of f1 feature in all classes.

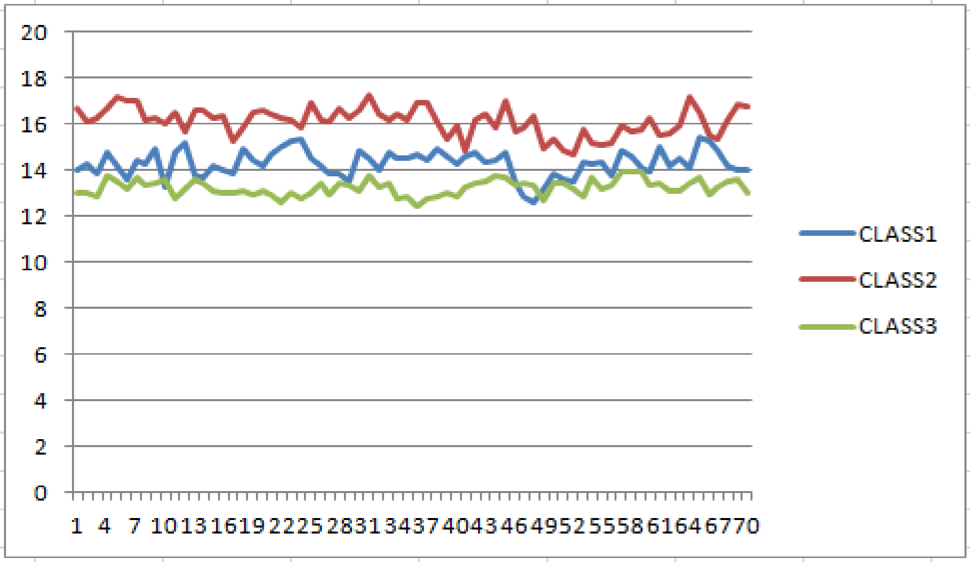


Figure 2.19: Behavior of f2 feature in all classes.

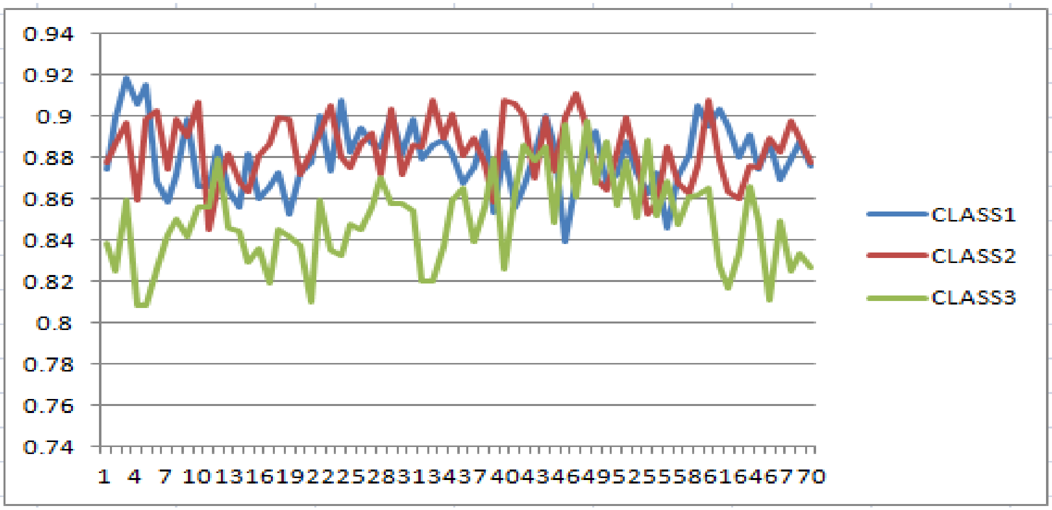


Figure 2.20: Behavior of f3 feature in all classes.

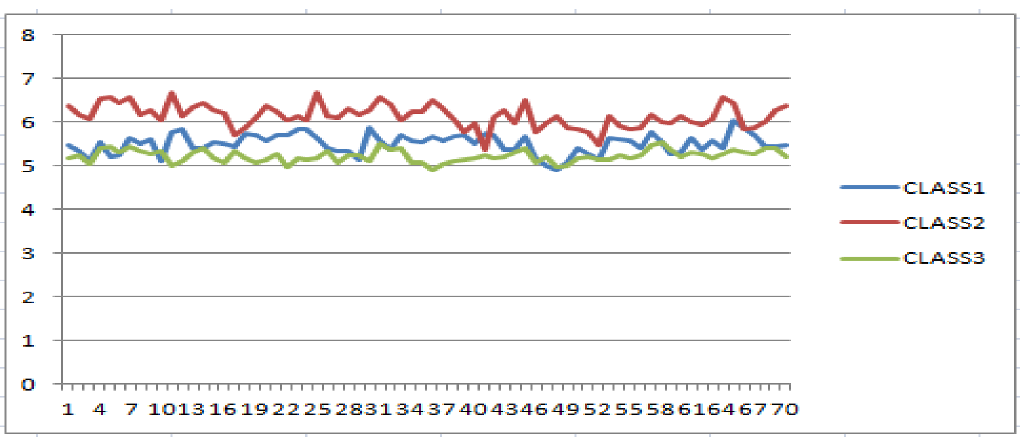


Figure 2.21: behavior of f4 feature in all classes.

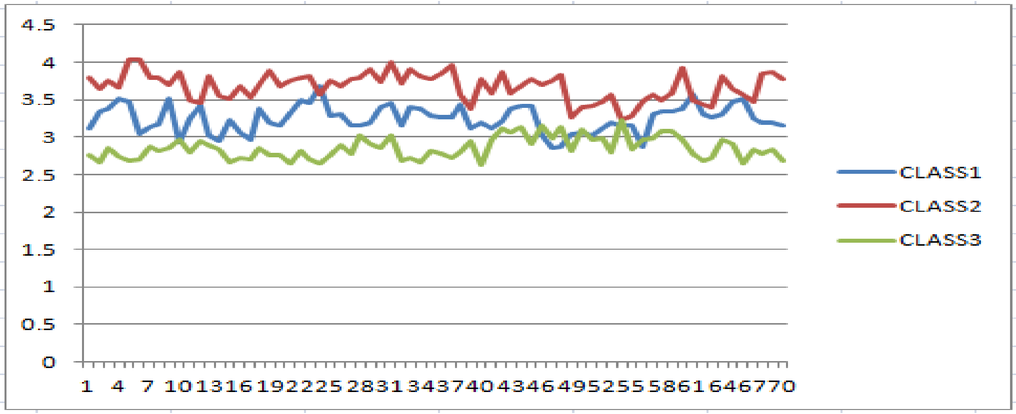


Figure 2.22: Behavior of f5 feature in all classes.

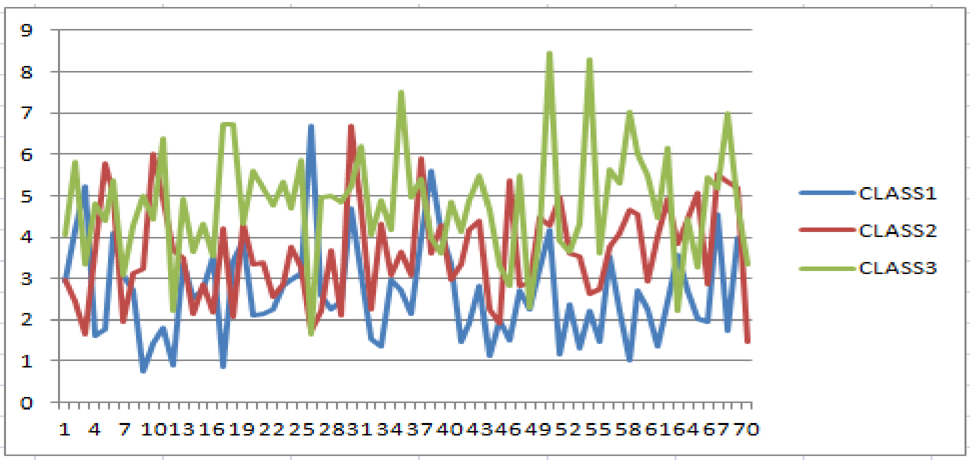


Figure 2.23: Behavior of f6 feature in all classes.

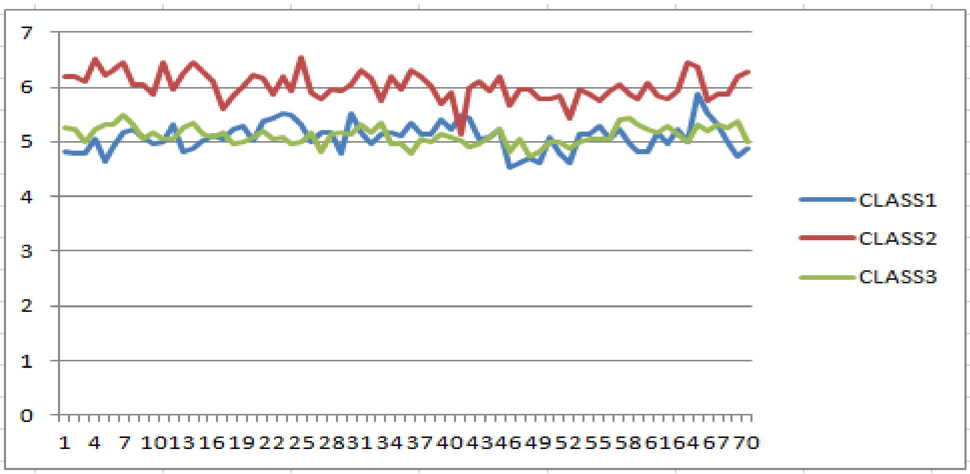


Figure 2.24: Behavior of f7 feature in all classes.

Chapter Three

The Proposed Classification System

3.1 Overview of the Proposed System

This thesis aims to develop a hieratical multi_agent system for patterns classification task. The system classifies a collection of real dataset with degree of generalization. The Proposed system uses five agents, each one of them implement as a multilayer feed forward neural networks with back propagation learning algorithm. The system uses two layers of agents, the top layer contains one agent works as control agent, and its responsibility is to select one of the agents in the lower layer to classify the entered pattern depending on features of data. Each neural network has one hidden layer. The number of nodes in input and output layers depends on the number of features and classes in datasets while the number of nodes in hidden layer is the average number of nodes of input and output layers. The training and testing of the system are implemented on real datasets where each classification agent is trained on a single dataset while the control is agent trained on all datasets to specify the suitable classified agent. In the absence of the control agent, an unknown pattern's domain will be checked by the agents one by one. The first agent will either classify it or will declare it as an unknown pattern. If the first agent couldn't classify the pattern, the second agent will try to classify it, and so on until it will be classified or it will be declared as an unknown pattern. In the absence of the control agent, the probability that any agent is the right agent is 1/n, where n is the number of classification agents.

The aim of the control agent is to either direct the tested pattern to the dedicated agent or to declare it as an unknown pattern. Thus using the control agent will reduce the time and effort of selecting the suitable agent to classify the tested pattern and as a result, the system reduces the time and effort to classify a collection of datasets.

Two techniques were applied to estimate and improve system’s generalization. To compare between generalization techniques and to assessment the efficiency of the proposed system, random noise was added to the tested data.

3.2 The System Structure

The structure of the proposed system consists of two hierarchal layers of agents shown in figure 3.1. The neural network that represents each agent is composed of three layers; input layer containing z nodes(where z is the number of features in a class) output layer containing c nodes (where c is the number of classes in dataset) and hidden layer containing p nodes, where:

$$p = (z + c) / 2 \qquad (3.1)$$

The number of nodes in input layer of the control agent equals to the largest number of the dataset's features, while its output layer equal to the number of classification agents.

3.2.1 Classification Agents

The classification agents are used to classify the datasets and they are implemented as neural networks with back propagation learning algorithm. The activation function that is used in back propagation learning algorithm is the sigmoid function.

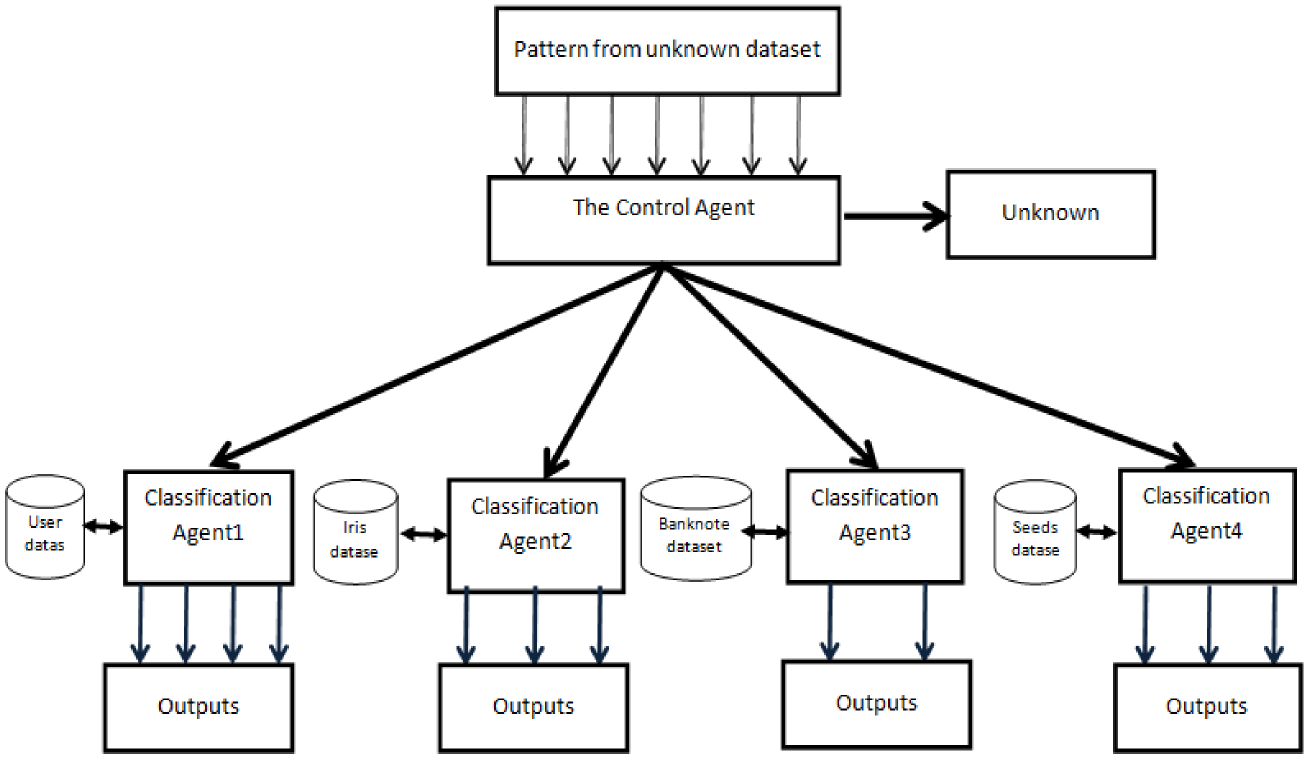


Figure 3.1: The general structure of the proposed system.

I. Classification agent1

This agent is used to classify the user knowledge level dataset implemented as a neural network with three layers with 5:5:4 nodes as input, hidden and output number of nodes, each layer is represented by an array processed repeatedly until the desired output is produced. Initial values are generated randomly allocated to weights arrays and then training the neural network with back propagation algorithm to reach the weights that give the desired output.

The bias node equals to one and has weights like other nodes, number of hidden nodes (p) determined by equation (3.1) where p =(5+4)/2≈5. The agent read training data from excel data base to be ready for actual test with unseen data.

II. Classification agent2

This agent used to classify the iris dataset implemented as neural network with three layers with 4:4:3 nodes as input, hidden and output number of nodes. Equation (3.1) is used to compute number of hidden nodes p where p= (4+3)/2≈4.

III. Classification agent3

This agent is used to classify the banknote authentication dataset implemented as neural network with 4:3:2 nodes as input, hidden and output number of nodes, number of hidden nodes is determined by equation (3.1) where p=(4+2)/2=3 .

IV. Classification agent4

This agent is used to classify the seeds dataset implemented as neural network with 7:5:3 nodes as input, hidden and output number of nodes, number of hidden nodes is determined by equation (3.1) where, p=(7+3)/2= 5 .

3.2.2 Control agent

The aim of the control agent is to either direct the tested pattern to the dedicated agent or it declared as an unknown pattern, it was implemented as neural network with 7:6:4 nodes as input, hidden and output layer, number of hidden nodes (p) determined by equation (3.1) where $p=(7+4)/2\approx 6$.

The agent read training data from excel database for all datasets to become ready for actual test.

3.3 Training Phase

Back propagation learning algorithm with pattern mode will be used to train each agent in the system; two techniques were applied to estimate agent’s generalization, The first one is generalization by earlier stopping criterion and the second one is generalization by regularization.

To compare between generalization techniques and to assessment the efficiency of the proposed system, random noise was added to test data. Then the best one of them will be adopted for each agent. The training phase of the proposed classification system is illustrated in figure 3.2, while the testing phase will be illustrated in figure 3.3.

3.3.1 Earlier Stopping Criterion

Earlier Stopping Criterion divided the dataset into three sets. The first set is the training set; the second set is the validation set and the third set is the (unseen) test set. So the dataset will be divided according the following rates: 60%, 20%, and 20% for the training, validation and testing set consequently.

Each classification agent will be trained on 60% of dataset; Training should stop at the suitable stopping point according to the validation set error (The generalization error) as illustrated in figure 3.4.

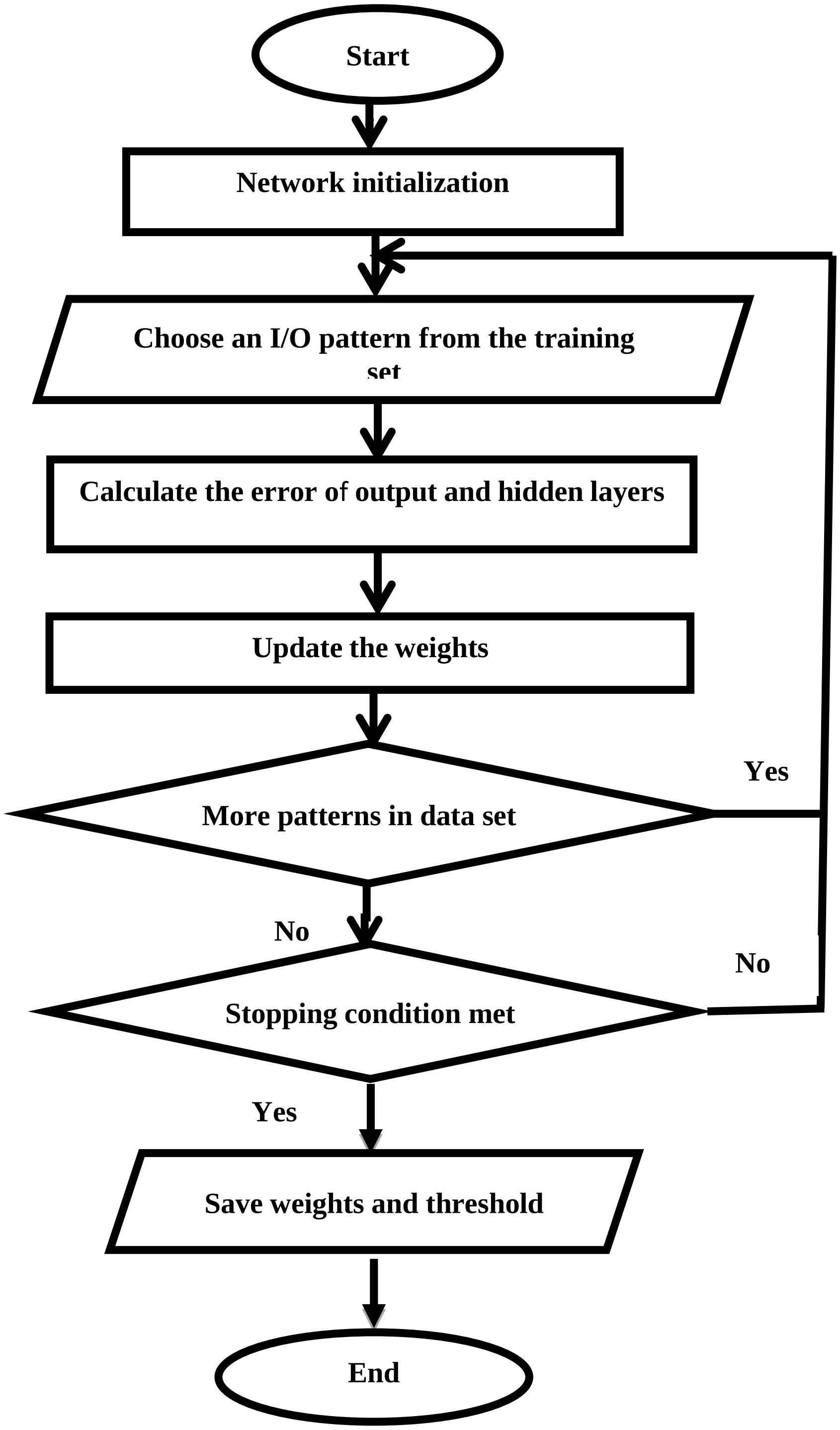


Figure 3.2: Pattern mode training flow chart for each agent.

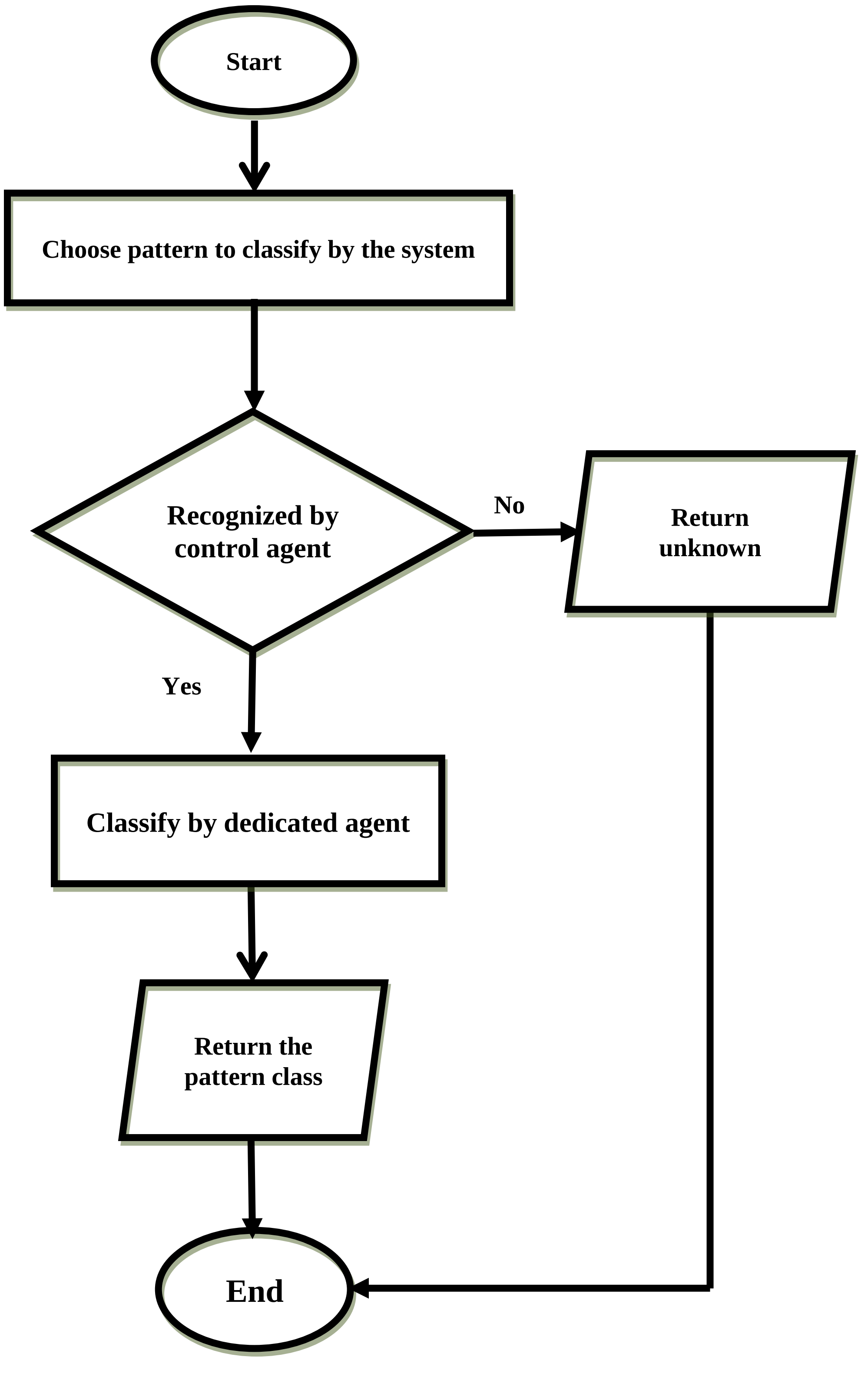


Figure 3.3: Testing phase flowchart for each agent.

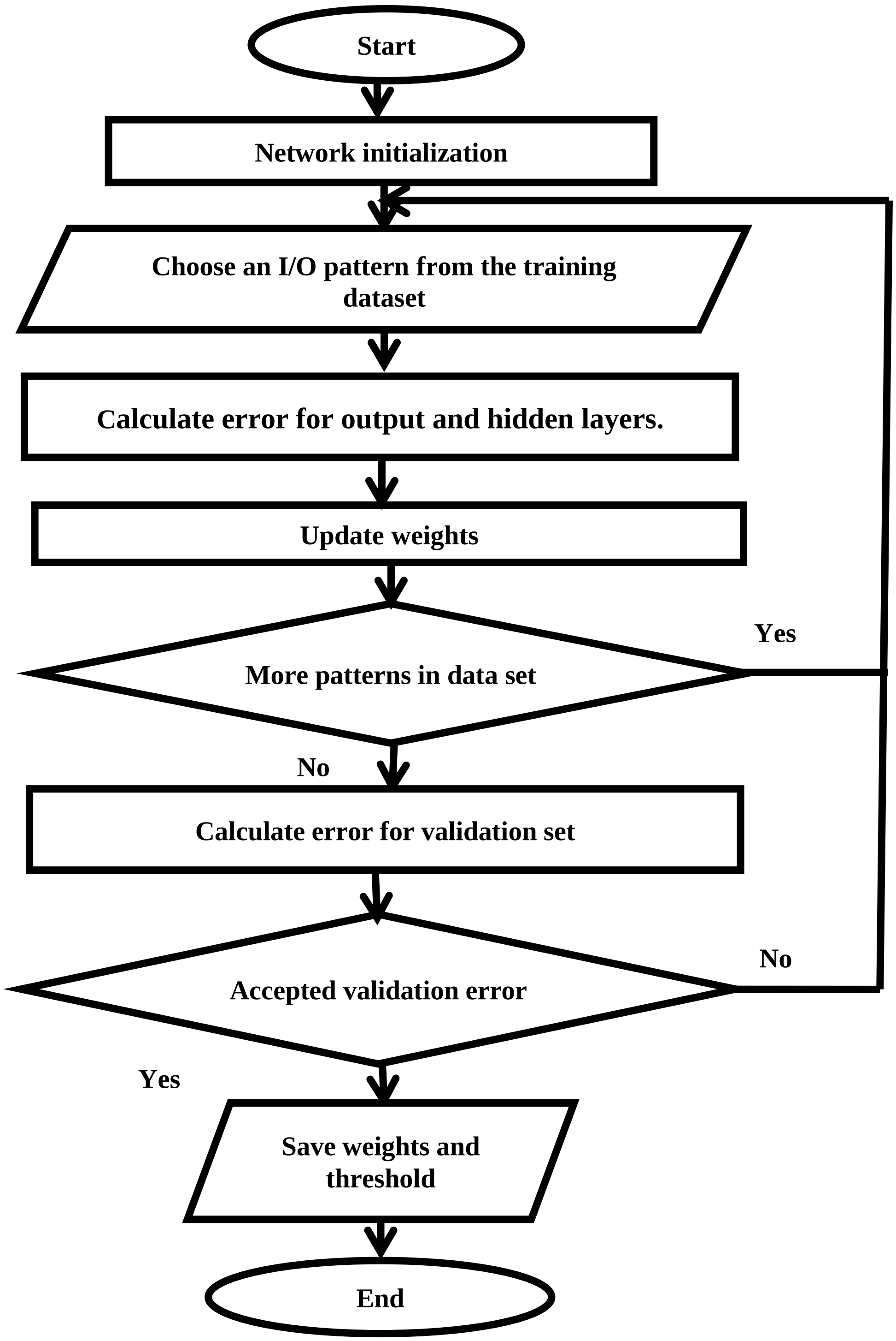


Figure 3.4: Generalizations by earlier stopping criteria flowchart

3.3.2 Regularization by Jitter

According to this technique the dataset is divided into two sets, the training set and testing set. A cross validation selection model with K=5 will be used to select the best training set for each agent.

To estimate the generalization of the system, a small amount of jitter(noise) will be added to the original data, the noise take range suitable to each dataset depending on nature of the data and generated randomly, each noised and original data is trained by the agent, then test the original and noised tested data.

Finally test the noised data (test data) by an agent trained with the original data to estimate the generalization, the training set will be chosen by cross validation, as illustrated in figure 3.5.

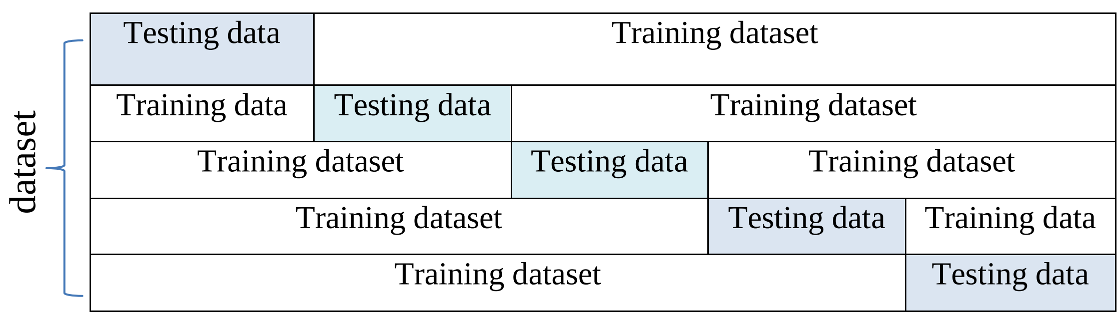


Figure 3.5: Cross validation selection model.

3.3.3 Back Propagation (BP) Training in Pattern Mode

The training with back propagation in pattern mode described in algorithm 3.1 where the weights are updated after each pattern offered to the network as an input.

Algorithm 3.1: BP algorithm with pattern mode.

Input: Training data from dataset.

Output: Trained data.

Begin:

Step 1: Give a small weights generated randomly to all weights on links between the layers in the network. The activation function used is sigmoid.

Step 2: For each training pattern (features and target) from training set:

(a) Compute the difference between the determined and the target values of the output layer by equation (2.2).

(b) Determine the error for each node in the hidden layer by equation (2.3).

(c) Adjust each link between the hidden layer and output layer by equation (2.4).

(d) Execute a similar adjustment to the weights on the input layer to hidden layer links.

Step 3: Repeat Step 2 to all errors becomes either zero or satisfactorily low.

Step 4: Save weights.

End.

3.4 Generate the noise

The noise was generated randomly in excel data base by using the fallowing function:

((ROUND ((RAND ()*(maxv - minv) + minv), 2))*(-1^RANDBETWEEN (1, 2))) (3.2).

Where

Maxv: represent the maximum feature's value.

Minv: represent the minimum feature's value.

This equation generates value between *maxv* and *minv*.

Table 3.9. Shows the noise ranges that will be added to each dataset.

Table 3.1: The noise ranges.

User Knowledge Level	Iris	Banknote authentication
0_0.03	0.05_0.1	0_0.1
0_0.05	0.05_0.2	0_0.3
0_0.07	0.05_0.3	0_0.7
0_0.09	0.05_0.4	0_1
0_0.1	0.05_0.5	0_1.5

3.5 Data Normalization

The technique that will be used to normalize the data is described in algorithm 3.2.

Algorithm 3.2: Normalization Procedure.

Input: sample from dataset

Output: normalized sample

Begin

For each sample in dataset

Step 1: Find $min\ c$ // minimum number

Step 2: Find $max\ c$ //maximum number

Step 3: For each feature in sample (f)

newf = f_i − minc / (maxc – minc)

Step 4: newf the normalized value of feature f_i

End

3.6 Data Fuzzify

The method that will be used to fuzzify the data using triangle shaped method is described in algorithm 3.3.

Algorithm 3.3: Fuzziness Procedure.

Input: sample from dataset

Output: Fuzzified sample

Begin

For each sample (column) in dataset

Step 1: Find $min\ c$ // minimum feature's value.

Step 2: Find $max\ c$ //maximum feature's value.

Step 3: Compute A= $(minc + maxc)/2$

For each feature (f_i) in column

If $f_i < minc$ then

$newf_i = 0$

elseif $f_i >= minc$ and $f_i <= A$ then

$newf_i = (f_i - minc)/(A - minc)$

elseif $f_i >= A$ and $f_i <= maxc$ then

$newf_i = (maxc - f_i)/(maxc - A)$

else $newf_i = 0$

endif

Endfor

Step 4: $newf_i$ is the fuzziness value of feature f_i.

End

Chapter Four
Experimental Results

4.1 Introduction

This chapter demonstrated the experimental results of the created hierarchal multi-agent system used to classify a collection of datasets. The performance and execution time are both evaluated, where different dataset (vary in features and number of patterns) are used. The training and testing of the system implemented on four real datasets (user knowledge level, iris, seeds and banknote authentication).

Each classification agent trained on a single dataset using back propagation with pattern mode while the control agent trained on all datasets to determine the reasonable classified agent.

Two possibilities first one when the generalization is not conditioned, the conventional divisions are used (70%, 30% or 80%, 20%) will be selected the best division of them. The second if the generalization is required, two techniques will be applied to estimate the generalization (earlier stopping criteria, regularization) for each agent and picked the best one after think about the outcomes.

The developed system has been established using Visual Basic.net programming language, and the tests have been conducted under the environment: Windows 7 (64 bit) operating system, laptop computer (Processor: AMD E_450 APU CPU, 1.65 GHz, and (3GB) RAM.

4.2 Experimental Results

The experiments will describe the training and testing each one of the system's agent. First if the generalization is not required where conventional divisions are used (70%, 30% or 80%, 20%) for training and testing respectively.

Second if the generalization is required initially the results of training with BP (in pattern mode) with earlier stopping criteria technique will be clarified then with regularization technique (cross validation as selection model). After that a clarification is done on the consequences of testing for each agent by confused matrix which is a matrix that will be used to describe the details of classes, after that estimate the generalization for each one of classification agents. This estimation is used to pick the right technique to construct the agent.

Note that agent4 (Seed) data set is not classified correctly neither using earlier stopping criteria technique nor by regularization techniques, so it's testing is ignored from these techniques.

4.2.1 Experiments of classification agents (conventional divisions)

i. The division (70% for training , 30% for testing)

The results of training the system on the ratio of (70% training) are shown in table 4.1. While the results of testing the system on 30% of data are shown in table 4.2.

ii. The division (80% for training , 20% for testing)

The results of training the system on the ratio of (80% training) are shown in table 4.3. While the results of testing the system on 20% of data are shown in table 4.4.

Table 4.1: Training the system on 70% of data.

The Classification agent	Training dataset	Number of patterns	Training time (sec)	No. of Iteration	Net error	True rate	False rate
Agent1	User Knowledge Level	282	17.03	2975	0.0094	97.52%	2.48%
Agent2	iris	105	5.60	3598	0.0009	95.23%	4.77%
Agent3	banknote authentication	960	0.15	14	0.0002	98.33%	1.67%
Control agent	All above	1347	60.66	2880	0.0009	100%	0%

Table 4.2: Testing the system on 30% of data.

Agents	True rate	False rate
Agent1	96.69% (117 patterns)	3.30% (4 patterns)
Agent2	100% (45 patterns)	0% (0 patterns)
Agent3	100% (411 patterns)	0% (0 patterns)
Control agent	99.82% (576 patterns)	0.18% (1 pattern)

Table 4.3: Training the system on 80% of data.

The Classification agent	Number of patterns	Training time (sec)	No. of Iteration	Net error	True rate	False rate
Agent1	322	5.07	782	0.0098	98.13%	1.86%
Agent2	120	3.26	1855	0.0008	90.00%	10.00%
Agent3	1097	0.14	12	0.0001	100%	0%
Control agent	1539	69.66	2925	0.0009	100%	0%

Table 4.4: Testing the system on 20% of data.

Classification Agents	True rate	False rate
Agent1	97.53% (79 patterns)	2.46% (2 patterns)
Agent2	90.00% (27 patterns)	10.00% (3 patterns)
Agent3	100% (274 patterns)	0% (0 patterns)
Control agent	99.74% (384 patterns)	0.25% (1 pattern)

iii. The results above shown the best division for each data set, these best divisions are shown in Table 4.5. While The **Classification agent4** was not able to distinguish only one class (33.33% of training and testing sets) in both divisions.

Table4.5: The best conventional divisions.

Classification agent	Best division
Agent1	80% training, 20% testing
Agent2	70% training, 30% testing
Agent3	80% training, 20% testing

The results of training and testing the control agent on the best divisions are shown in table 4.6.

Table 4.6: Training and testing of control agent on best divisions.

Number Of Patterns	1524
Training Accuracy	100% (1524 patterns)
False Rate (Training)	0% (0 pattern)
Iterations	287
Time	7.2 sec
Net Error	0.0029
Testing Accuracy	100% (400 patterns)
False Rate(Training)	0% (0 patterns)

4.2.2 Experiments of classification agents (generalization techniques)

The results of training and testing each agent with two techniques (*Earlier Stopping Criteria technique* and *Regularization Technique with Jitter*) will be displayed for each agent and select the best technique for each one independently.

A. Classification Agent1

This agent trained on user knowledge level dataset with back propagation learning algorithm in pattern mode.

i. Earlier Stopping Criteria

The results of training agent1 by earlier stopping criteria technique are appeared in table 4.7, where the dataset partitioned into 60% for training, 20% for validation and 20% for testing. Likewise the overall accuracy of testing is appeared in table 4.8.

Table 4.7: Training the agent1 by earlier stopping criteria.

Number of patterns	241
Training time	0.00001 s
Good Iteration	3763
Total Iteration	23764
MSE(training set)	6.253E-04
MSE(validation set)at good iteration	9.4567E-09
MSE(validation set)at last iteration	3.544E-08
Accuracy (training set)	97.5% (235 patterns)
Error Rate(training set)	2.5% (6 patterns)
Accuracy (validation set)	85.2 % (69 patterns)
Error Rate(validation set)	14.8% (12 patterns)

Table 4.8: Overall test accuracy of agent1 by earlier stopping criteria.

Number of patterns	**Overall Accuracy**	**Error Rate**
81	95.06% (77 patterns)	4.93% (4 patterns)

The results of testing test set which is 20% of dataset are shown in the confused matrix in table 4.9. The details accuracy by class is shown in table 4.10.

To compare the earlier stopping criteria with regularization and to actual measure of generalization, the noise added to test set, the results of testing the agent1 on test set (with noise) are performed with different ranges of noise, appeared in table 4.11, where each percent is an average of five experiments.

Table 4.9: Confused matrix of test agent1 by earlier stopping criteria.

	High	Medial	Low	Very Low	Actual No.
High	26	0	0	0	26
Medial	2	17	0	0	19
Low	0	2	18	0	20
Very Low	0	0	0	16	16
Total number of predicate	28	19	18	16	81

Table 4.10: Accuracy by class of agent1 using earlier stopping criteria technique.

Class	TP rate	FP rate	FN rate	Precision	Recall
High	1.00	0.08	0.00	0.93	1.00
Medial	0.89	0.11	0.11	0.89	0.89
Low	0.90	0.00	0.1	1.00	0.90
Very Low	1.00	0.00	0.00	1.00	1.00

Table 4.11: Estimate the generalization of agent1 by earlier stopping criteria where number of patterns is 81.

Noise range	Overall Accuracy	Error Rate (Misclassification Rate)
0_0.03	92.60%	7.40%
0_0.05	91.36%	8.64%
0_0.07	86.42%	13.58%
0_0.09	83.95%	16.05%
0_0.1	85.20%	14.80%

ii. Regularization Technique

Select the best training and testing sets by k-fold cross validation (with k=5). The results of training agent1 by cross validation will be described in table A (1) in the appendix. Every training set contain 322 (80% of data) patterns. The results of testing agent1 by cross validation will be described in table A (2) in the appendix. Where, each testing set contains 81 (Roughly 20% of data) patterns.

The results of testing test set of experiment1 appeared in confused matrix in table 4.12, and the details of accuracy by class are shown in table 4.13.

Table 4.12: Confused matrix of test agent1 (experiment 1).

	High	**Medial**	**Low**	**Very Low**	**Actual No.**
High	26	0	0	0	26
Medial	1	17	1	0	19
Low	0	0	20	0	20
Very Low	0	0	0	16	16
Total number of predicates	27	17	21	16	81

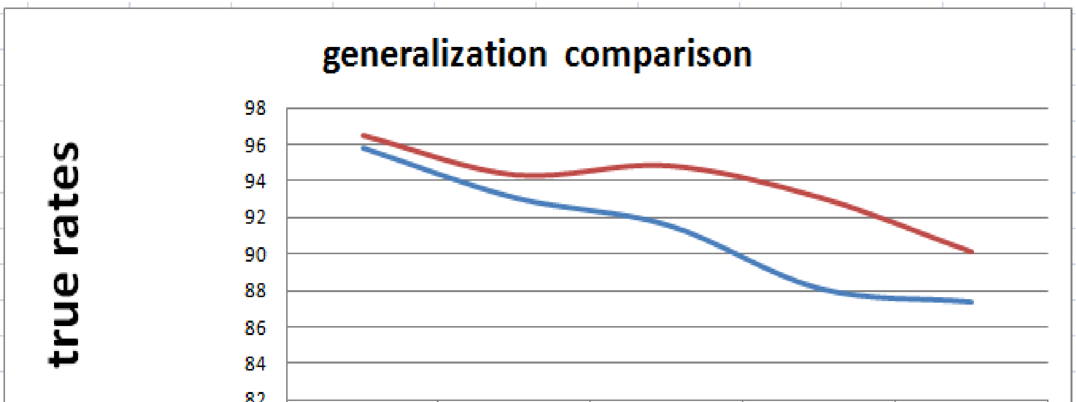
Table 4.13: Accuracy by class of agent1 (experiment 1).

Class	**TP rate**	**FP rate**	**FN rate**	**Precision**	**Recall**
High	1	0.04	0.00	0.96	1
Medial	0.89	0.00	0.11	1	0.89
Low	1	0.05	0.00	0.95	1

The best result is experiment1, where the rate of classification test set is 97.53 % (79 patterns). According to this, experiment1 will be adopted after adding noise to estimate generalization.

The noise take range suitable to each dataset and generated randomly, the confused and original data is trained by the agent1, and then test the original and confused tested data to estimate and compare the generalization. The results will be shown in table A (3) in the appendix.

All outcomes in table A (3) represent overall accuracy of classification and each percent is an average of five experiments. Enhancing generalization is not required when the added noise is greater than 0.05, as shown in figure 4.1.



	1	2	3	4	5
training with noise	95.79	93.08	91.6	88.14	87.4
training without noise	96.45	94.31	94.81	93.08	90.11

Figure 4.1: Generalization measurements (experiment1).

iii. The result of generalization's comparison between earlier stopping criteria and regularization technique (with noised test data) is shown in table 4.14 and figure 4.2.

Table 4.14: Compare the behavior of agent1 by earlier stopping criteria and regularization technique.

Noise range	Earlier Stopping Criteria	Regularization
0_0.03	92.60%	96.45%
0_0.05	91.36%	94.31%
0_0.07	86.42%	94.81%
0_0.09	83.95%	93.08%
0_0.1	85.20%	90.11%

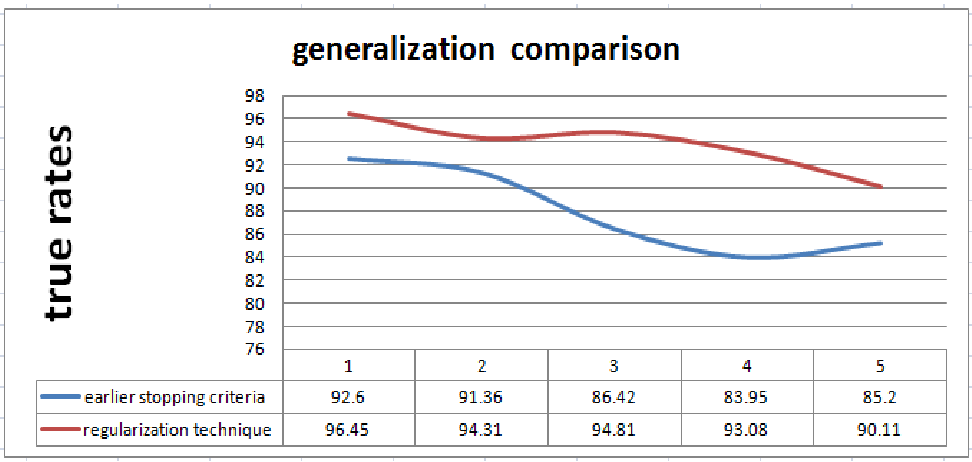


Figure 4.2: Comparison earlier stopping criteria and regularization technique.

This comparison demonstrates that the regularization technique is superior to earlier stopping criteria, where the generalization degree in regularization technique is larger than earlier stopping criteria at all amounts of noise. Presently the basic condition which is adopted to pick the best technique to build agent1 is the ability to generalization and the results of analysis shown that the regularization technique by adding noise to enhance generalization is the best, as indicated by this investigation, regularization technique will be adopted for classifying agent1.

B. Classification Agent2

Back propagation learning algorithm in pattern mode is used to train agent2 on iris dataset.

i. Earlier Stopping Criteria

The results of training agent2 by earlier stopping criteria technique is shown in table 4.15, where the dataset divided into 60% for training, 20% for validation and 20% for testing. Also the overall accuracy of testing is shown in table 4.16.

Table 4.15: Training the agent2 by earlier stopping criteria.

Number of patterns	90
Training time	34.0 sec
Good Iteration	10402
Total Iteration	20403
MSE(training set)	0.56 E-04
MSE(validation set)at good iteration	0.20 E-04
MSE(validation set)at last iteration	0.56 E-04
Accuracy (training set)	98.89% (89)
Error Rate(training set)	1.11% (1)
Accuracy (validation set)	96.67%(29)
Error Rate(validation set)	3.33%(1)

Table 4.16: Overall test accuracy of agent2 by earlier stopping criteria.

Number of patterns	Accuracy	Error Rate
30	100% (30 patterns)	0.0% (0 patterns)

The results of testing test set which is 20% of dataset shown in confused matrix in table 4.17. And the details accuracy by class is shown in table 4.18.

To compare the earlier stopping criteria with regularization and to actual measure of generalization, the noise added to test set, the results of testing agent2 on test set (with noise) are performed with different ranges of noise, appeared in table 4.19, where each percent is an average of five experiments.

Table 4.17: Confused matrix of test agent2 by earlier stopping criteria.

	Setosa	Versicolor	Virginica	Actual no.
Setosa	10	0	0	10
Versicolor	0	10	0	10
Virginica	0	0	10	10
Total number of predicate	10	10	10	30

Table 4.18: Accuracy by class of agent2 with Earlier Stopping criteria.

class	TP rate	FP rate	FN rate	Precision	Recall
Setosa	1.00	0.00	0.00	1.00	1.00
Versicolor	1.00	0.00	0.00	1.00	1.00
Virginica	1.00	0.00	0.00	1.00	1.00
The Average	**1.00**	**0.00**	**0.00**	**1.00**	**1.00**

Table 4.19: Estimate the generalization of agent2 by earlier stopping criteria.

Noise range	Overall Accuracy	Error Rate (Misclassification Rate)
0.05_0.1	100%	0%
0.05_0.2	99.33%	0.67%
0.05_0.3	97.34%	2.66%
0.05_0.4	92.60%	7.40%
0.05_0.5	92.60%	7.40%

ii. Regularization Technique

Select the best training and testing sets by k-fold cross validation (with k=5) to add the noise. The results of training agent2 by cross validation will be depicted in table A (4) in the appendix. Each training set contain 120 patterns. The results of testing agent2 by cross validation will be described in table A (5) in the appendix, where each testing set is contain 30 patterns.

The results of testing test set of experiment4 appeared in confused matrix in table 4.20, and the accuracy by class is shown in table 4.21.

Table 4.20: Confused matrix of test agent2 (experiment4).

	Setosa	Versicolor	Virginica	Actual No.
Setosa	10	0	0	10
Versicolor	0	9	1	10
Virginica	0	0	10	10
Total number of predicate	10	9	11	30

Table 4.21: Accuracy by class of agent2 (experiment 4).

class	TP rate	FP rate	FN rate	Precision	Recall
Setosa	1	0	0	1	1
Versicolor	0.9	0	0.1	1	0.9
Virginica	1	0.1	0	0.91	1

The best results is experiment4, where the rate of classification test set is 96.67 % (29 patterns) like experiment5 but the rate of classifying training set is higher than experiment5. According to this, experiment4 will be adopted.

To estimate the generalization, a noise is added to the original data; each noised and original data is trained by the agent, and then test the original and confused tested data. Finally test the confused data (test data) by an agent trained with the original data to estimate the generalization. The results will be shown in table A(6) in the appendix.

All the results in table A (6) represent overall accuracy of classification and each percent is an average of five experiments, also enhancing generalization is not required when the added noise is (0.05 to 0.5) as shown in figure 4.3.

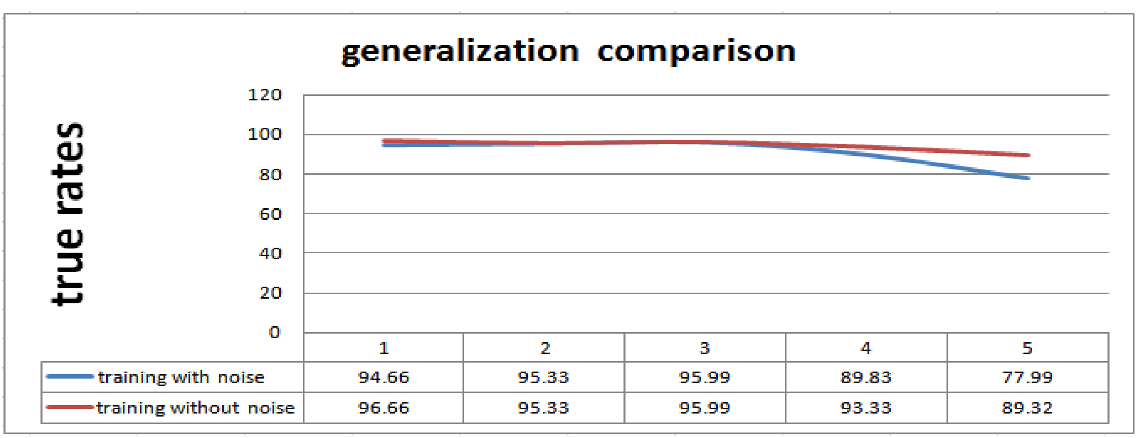


Figure 4.3: Generalization measurements (experiment4).

iii. The results of generalization's comparison between earlier stopping criteria and regularization technique (which is adding noise) showed in table 4.22 and figure 4.4.

Table 4.22: Compare the behavior of agent2 by earlier stopping criteria and regularization technique (agent2).

Noise range	Earlier Stopping Criteria	Regularization
0.05_0.1	100%	96.66%
0.05_0.2	99.33%	95.33%
0.05_0.3	97.34%	95.99%
0.05_0.4	92.60%	93.33%
0.05_0.5	92.60%	89.32%

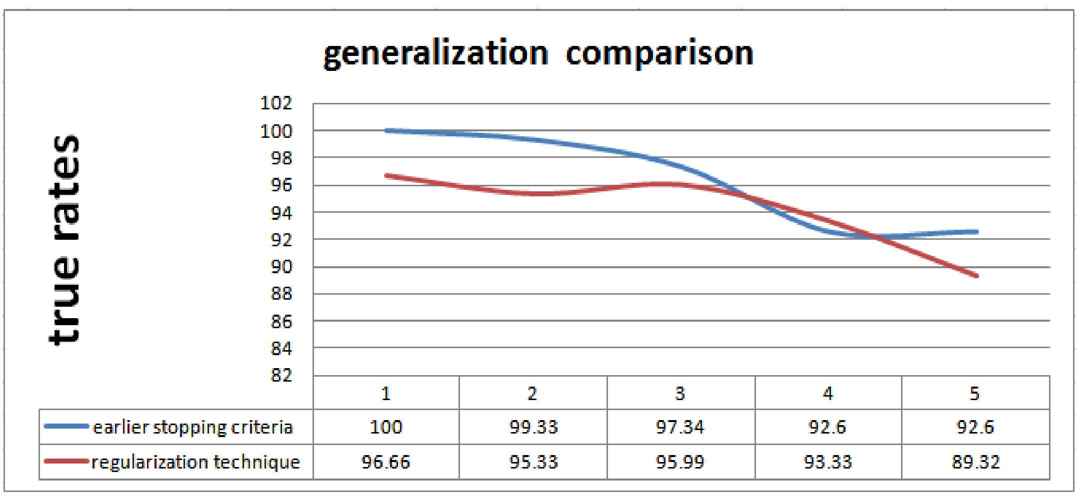


Figure 4.4: Comparison earlier stopping criteria and regularization technique.

This comparison demonstrate that the earlier stopping criteria technique is superior to regularization, where the generalization degree in regularization technique is less than earlier stopping criteria at most amounts of noise, this critical condition is adopted to pick the best technique to construct agent2. As indicated by this investigation, earlier stopping criteria technique will be adopted for classifying agent2.

C. Classification Agent3

Back propagation learning algorithm in pattern mode is used to train agent3 on banknote authentication dataset.

i. Earlier Stopping Criteria

The results of training agent3 by earlier stopping criteria technique is shown in table 4.23, where the dataset divided into 60% for training, 20% for validation and 20% for testing. Also the overall accuracy of testing is shown in table 4.24.

Table 4.23: Training the agent3 with earlier stopping criteria.

Number of patterns	823
Training time	42.00 s
Good Iteration	10000
Total Iteration	10001
MSE(training set)	1.26E-09
MSE(validation set)at good iteration	0.0037
MSE(validation set)at last iteration	0.0037
Accuracy (training set)	100% (823 patterns)
Error Rate(training set)	0% (0 patterns)
Accuracy (validation set)	100% (274 patterns)
Error Rate(validation set)	0% (0 patterns)

Table 4.24: Overall test accuracy of agent3 by earlier stopping criteria.

Number of patterns	**Accuracy**	**Error Rate**
274	100% (274 patterns)	0.0 % (0 patterns)

The results of testing test set which is 20% of dataset shown in confused matrix in table 4.25. And the details accuracy by class is shown in table 4.26.

To compare the earlier stopping criteria with regularization and to actual measure of generalization, the noise added to test set, the results of testing the agent3 on test set (with noise) are performed with different ranges of noise, appeared in table 4.27.

Table 4.25: Confused matrix of test agent3 by earlier stopping criteria.

	0	**1**	**Actual no.**
0	155	0	155
1	0	119	119
Total number of predicate	155	119	274

Table 4.26: Accuracy by class of agent3 by earlier stopping criteria.

Class	**TP rate**	**FP rate**	**FN rate**	**Precision**	**Recall**
0	1.00	0.00	0.00	1.00	1.00
1	1.00	0.00	0.00	1.00	1.00
The Average	1.00	0.00	0.00	1.00	1.00

Table 4.27: Estimate the generalization of agent3 by earlier stopping criteria.

Noise Range	Overall Accuracy	Error Rate (Misclassification Rate)
0_0.1	100%	0%
0_0.3	99.63%	0.37%
0_0.7	98.68%	1.32%
0_1	97.95%	2.05%
0_1.5	95.25%	4.75%

ii. Regularization Technique

Select the best training and testing sets by k-fold cross validation (with k=5) to add the noise. The results of training agent3 by cross validation will be described in table A (7) in the appendix. Each training set contain 1097 patterns, and the results of testing agent3 by cross validation will described in table A (8) in the appendix, where each testing set contain 274 patterns.

The results of testing test set of experiment5 appeared in confused matrix in table 4.28. And the accuracy by class is shown in table 4.29.

Table 4.28: Confused matrix of test agent3 (experiment 5).

	0	1	Actual no.
0	142	1	143
1	2	129	131
Total number of predicate	144	130	274

Table 4.29: Accuracy by class of agent3 (experiment 5).

Class	TP rate	FP rate	FN rate	Precision	Recall
0	0.99	0.01	0.01	0.99	0.99
1	0.98	0.01	0.02	0.99	0.98

The best results are experiment5, where the rate of classification test set is 98.91 % (271 patterns), according to this result experiment5 will be adopted.

To estimate the generalization, a noise is added to the original data; each noised and original data is trained by the agent, and then test the original and confused tested data. Finally test the confused data (test data) by an agent trained with the original data to estimate the generalization. The results will be shown in table A (9) in the appendix. All results in table above represent overall accuracy of classification and each percent is an average of five experiments, also enhancing generalization is not required when the added noise is (0.1 to 0.7) as shown in figure 4.5.

iii. The results of generalization's comparison between earlier stopping criteria and regularization technique (which is adding noise) showed in table 4.30 and figure 4.6.

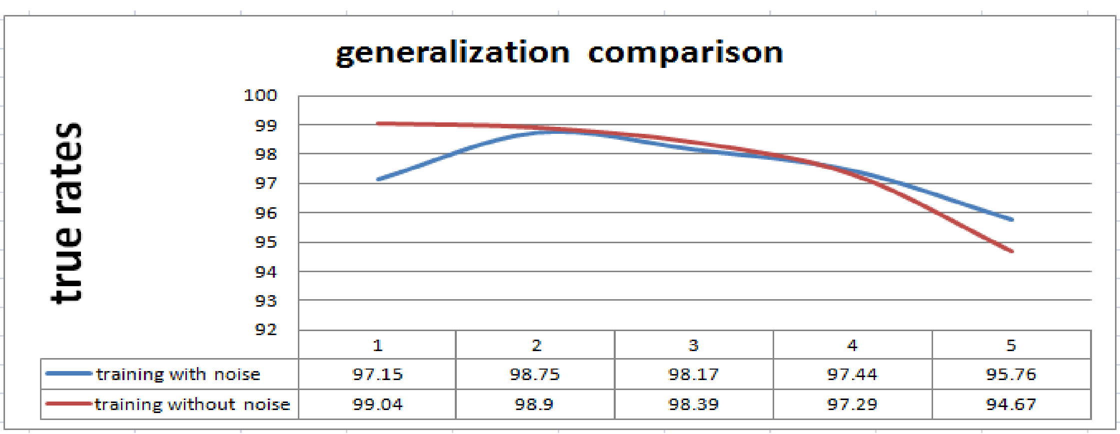
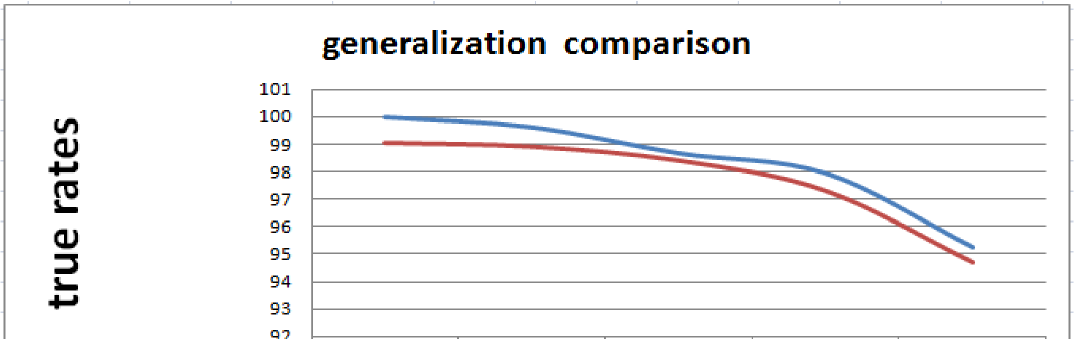


Figure 4.5: Generalization measurements (experiment5).

Table 4.30: Compare agent3 behavior by earlier stopping criteria and regularization technique (agent3).

Noise range	Earlier Stopping Criteria	Regularization
0.05_0.1	100%	99.04%
0.05_0.2	99.63%	98.90%
0.05_0.3	98.68%	98.39%
0.05_0.4	97.95%	97.29%
0.05_0.5	95.25%	94.67%



	1	2	3	4	5
earlier stopping criteria	100	99.63	98.68	97.95	95.25
regularization technique	99.04	98.9	98.39	97.29	94.67

Figure 4.6: Comparison earlier stopping criteria and regularization technique.

This comparison demonstrates that the earlier stopping criteria technique is superior than regularization, where the generalization degree in regularization technique is less than earlier stopping criteria at all amounts of noise, Presently the basic condition which is adopted to pick the best technique to build agent3 is the ability to generalization and the results of analysis shown that the earlier stopping criteria technique is the best, as indicated by this investigation,, earlier stopping criteria technique will be adopted for classifying agent3.

D. Classification Agent4

Back propagation learning algorithm in pattern mode is used to train agent4 on seeds dataset. All experiments in all techniques gave classification rate 33.33% of training and testing which represent only one class from three classes.

4.2.3 Experiments of Control Agent

The control agent train and test will be on data from all dataset as fallowing:

A. Control agent's training

The results above are shown the best training sets of each dataset, in state of earlier stopping criteria technique there is validation set, First the validation set will be added to test set, second the validation set will be added to training set, then select the best state of them.

First the results of training sets (without validation sets) which are representing 60% of each dataset are shown in table 4.31.

Table 4.31: Control agent's training (without validation sets).

Number of Patterns	1235
True rate	100% (1235 patterns)
False rate	0% (0 pattern)
Iterations	162
Time	3.17sec
Net error	0.005

Second the training sets (with validation sets) which are representing 80% of each dataset are shown in table 4.32.

Table 4.32: Control agent's training (with validation sets).

Number of patterns	1509
True rate	99.87% (1507 pattern)
False rate	0.13% (2 pattern)
Iterations	156
Time	3.81s
Net error	0.005

B. Hard test of control agent

Test sets of classification agents are represent test set of control agent, first (without validation sets) and Second (with validation sets) as shown in table 4.33.

Table 4.33: Control agent hard test (without and with validation sets).

	Number of patterns	**True rate**	**False rate**
Test set with validation	689	100% (689 patterns)	0% (0 patterns)
Test set without validation	385	99.48% (383 patterns)	0.52% (2 patterns)

According to the results in table 4.31 and 4.32, the best state that will be adopted to build the control agent is training the control agent on training set only without validation set which is added to test set.

4.3 Results comparison with weka software

Weka software will be used to classify each dataset then compare its results with the results of the corresponding agent; the class that use is

MultilayerPerceptron which is the classifier that uses backpropagation to classify instances. The nodes in this network are all sigmoid.

4.3.1 User knowledge level dataset

The train set of classification agent1 consider as training set in weka, the results of training are shown in table 4.34.

Table 4.34: Results of training (agent1's data) with weka.

Number of patterns	322
Iteration	782
MSE(training set)	0.1115
Accuracy (training set)	97.205%(313 patterns)
Error Rate(training set)	2.795%(9 patterns)

The test set of classification agent1 consider as test set to weka, The overall accuracy of testing test set which is 20% of dataset is shown in table 4.35.

While testing details are shown by confused matrix in table 4.36. And details of accuracy by class are shown in table 4.37. Then overall accuracy comparison is shown in table 4.38.

Table 4.35: Test 20% of data (agent1) with weka.

Number of patterns	Overall Accuracy	False rate
81	97.53 % (79 pattern)	2.47 % (2 pattern)

Table 4.36: Confused matrix of test (agent1) by weka.

	High	Medial	Low	Very low	Actual no.
High	26	0	0	0	26
Medial	1	18	0	0	19
Low	0	0	19	1	20
Very Low	0	0	0	16	16
Total number of predicate	27	18	19	17	81

Table 4.37: The accuracy by class (agent1) with weka.

Class	TP rate	FP rate	Precision	Recall
High	1	0.018	0.963	1
Medial	0.947	0	1	0.947
Low	0.950	0	1	0.950
Very low	1	0.015	0.941	1

Table 4.38: Overall accuracy comparison.

	Weka	Agent1
Overall Accuracy	97.53 % (79 patterns)	97.53 % (79patterns)
False rate	2.47 % (2 patterns)	2.47 % (2 patterns)

4.3.2 Iris Dataset

The train set of classification agent2 consider as training set in weka, the results of training are shown in table 4.39.

Table 4.39: Results of training on 60 % of iris dataset by weka.

Number of patterns	90
Iteration	10402
MSE(training set)	0.0028
Accuracy (training set)	100%(90 patterns)
Error Rate(training set)	0%(0 patterns)
MSE(validation set)	0.2079
Accuracy (validation set)	93.33%(28 patterns)
Error Rate(validation set)	6.67%(2 patterns)

The test set of classification agent2 consider as test set to weka, results of testing test set which is 20% of dataset shown in confused matrix in table 4.40. The overall accuracy of testing by weka is shown in table 4.41. The details of accuracy by class are shown in table 4.42, and the overall accuracy comparison is shown in table 4.43.

Table 4.40: Confused matrix of test iris data by weka.

	Setosa	**Versicolor**	**Virginica**	**Actual no.**
Setosa	10	0	0	10

Versicolor	0	10	0	10
Virginica	0	0	10	10
Total number of predicate	10	10	10	30

Table 4.41: Test on 20% of iris data with weka.

Number of patterns	**Overall Accuracy**	**False rate**
30	100 % (30 patterns)	0 % (0 patterns)

Table 4.42: The accuracy by class.

Class	**TP rate**	**FP rate**	**Precision**	**Recall**
High	1.00	0	1.00	1.00
Medial	1.00	0	1.00	1.00
Low	1.00	0	1.00	1.00
Very Low	1.00	0	1.00	1.00

Table 4.43: Overall accuracy comparison.

	Weka	**Agent2**
Overall Accuracy	100 % (30 patterns)	100 % (30 patterns)
False rate	0 % (0 patterns)	0 % (0 patterns)
Accuracy validation	93.33%(28 patterns)	96.67%(29)
False rate validation	6.67%(2 patterns)	3.33%(1)

4.3.3 Banknote Authentication Dataset

Weka does not classify banknote authentication dataset in original data format.

4.3.4 Seeds Dataset

Weka does not classify seeds dataset in original data format.

4.4 Normalize agents

The data converted to normalize form between 0 and 1 for all agents to training and testing.

4.4.1 Normalize agent1

The results of training agent1 on normalized data are shown in table 4.44. And results of testing test set which is 20% of dataset shown in confused matrix in table 4.45. Also the overall accuracy of testing is shown in table 4.46, while the details of accuracy by class are shown in table 4.47.

Table 4.44: Agent1 training results (normalize data).

Number of patterns	322
Training time	0.000
Iteration	2039
MSE(training set)	0.00977
Accuracy (training sets)	96.89%(312 patterns)
Error Rate(training sets)	3.10%(10 patterns)

Table 4.45: Confused matrix of test agent1 (normalize data).

	High	**Medial**	**Low**	**Very low**	**Actual no.**
High	26	0	0	0	26
Medial	2	14	3	0	19
Low	0	0	19	1	20
Very Low	0	0	1	15	16
Total number of predicate	28	14	22	16	81

Table 4.46: Overall accuracy of test the agent1 (normalize data).

Number of patterns	**Overall Accuracy**	**False rate**
81	91.36 % 74 (patterns)	8.64 % 7 (patterns)

Table 4.47: Accuracy by class of agent1 (normalize data).

Class	**TP rate**	**FP rate**	**FN rate**	**Precision**	**Recall**
High	1	0.08	0	0.93	1
Medial	0.74	0	0.26	1	0.74
Low	0.95	0.2	0.05	0.83	0.95
Very Low	0.94	0.06	0.06	0.9	0.94

The results of normalize data will comparison with original data as shown in table 4.48.

Table 4.48: Agent1 overall accuracy comparison with normalize data.

	With normalize	**Without normalize**
Overall Accuracy	91.36%(74 patterns)	97.53 % (79 patterns)
False rate	8.64 % (7 patterns)	2.47% (2 patterns)

According to results in table 4.48 normalize agent1 will do not improve the overall accuracy of classification, therefore will save the agent1 without normalize.

4.4.2 Normalize agent2

The results of training agent2 on normalized data are shown in table 4.49, the results of testing test set (normalize data) which is 20% of dataset shown in confused matrix in table 4.50, the overall accuracy of testing (normalize data) is shown in table 4.51, while the details of accuracy by class are shown in table 4.52.

Table 4.49: Agent2 training results (normalize data).

Number of patterns	90
Training time	13.0 s
Good Iteration	99999
Total Iteration	100000
MSE(training set)	2.66E-04
MSE(validation set) at good iteration	2.90E-03
MSE(validation set)at last iteration	2.90E-03
Accuracy (training set)	100%(90 patterns)
Error Rate(training set)	0%(0 patterns)
Accuracy (validation set)	90%(27 patterns)
Error Rate(validation set)	10%(3 patterns)

Table 4.50: Confused matrix of test agent2 (normalize data).

	Setosa	**Versicolor**	**Virginica**	**Actual no.**
Setosa	10	0	0	10
Versicolor	0	10	0	10
Virginica	0	0	10	10
Total number of predicate	10	10	10	30

Table 4.51: Overall accuracy of test the agent2 (normalize data).

Number of patterns	**Accuracy**	**Error Rate**
30	100% (30 patterns)	0.0% (0 patterns)

Table 4.52: Accuracy by class of agent2 (normalize data).

Class	**TP rate**	**FP rate**	**Precision**	**Recall**
Setosa	1.00	0.00	1.00	1.00
Versicolor	1.00	0.00	1.00	1.00
Virginica	1.00	0.00	1.00	1.00
The Average	1.00	0.00	1.00	1.00

The results of normalize data will comparison with original data as shown in table 4.53.

Table 4.53: Agent2 overall accuracy comparison with normalize data.

	With normalize	**Without normalize**
Overall Accuracy	100% (30 patterns)	100% (30 patterns)
False rate	0 % (0 patterns)	0 % (0 patterns)

According to results in tables 4.53 normalize agent2 given same overall accuracy of agent2 (100%) for test set, therefore will compare the results of training in table 4.49, these results shown that the number of error patterns in training and validation sets is three with normalize, while it is two without normalize, therefore the agent2 will not normalize.

4.4.3 Normalize agent3

The results of training agent3 on normalized data are shown in table 4.54, the results of testing test set (normalize data) which is 20% of dataset shown in confused matrix in table 4.55, also overall accuracy of testing is shown in table 4.56, while details of accuracy by class are shown in table 4.57.

Table 4.54: Agent3 training results (normalize data).

Number of patterns	823
Training time	44.0 s
Good Iteration	10000
Total Iteration	10001
MSE(training set)	9.56E-09
MSE(validation set)at good iteration	3.66E-03
MSE(validation set)at last iteration	3.66E-03
Accuracy (training set)	100% (823)
Error Rate(training set)	0%(0)
Accuracy (validation set)	97.45%(267)
Error Rate(validation set)	2.55%(7)

Table 4.55: Confused matrix of test agent3 (normalize data).

	0	**1**	**Actual no.**
0	155	0	155
1	0	119	119
Total number Of predicate	155	119	274

Table 4.56: Overall accuracy of test the agent3 (normalize data).

Number of patterns	Overall Accuracy	Error Rate
274	100% (274 patterns)	0.0% (0 patterns)

Table 4.57: Accuracy by class of agent3 (normalize data).

Class	TP rate	FP rate	FN rate	Precision	Recall
0	1.00	0.00	0.00	1.00	1.00
1	1.00	0.00	0.00	1.00	1.00
The Average	1.00	0.00	0.00	1.00	1.00

The results of normalize data will comparison with original data as shown in table 4.58.

Table 4.58: Agent3 overall accuracy comparison with normalize data.

	With normalize	Without normalize
Overall Accuracy	100% (274 patterns)	100% (274 patterns)
False rate	0 % (0 patterns)	0 % (0 patterns)

According to results in tables 4.58 normalize agent3 given same overall accuracy of agent3 (100%) for test set, therefore will compare the results of training in table 4.54.

These results shown that the number of error patterns in training and validation sets is seven with normalize, while it is zero without normalize, therefore the agent3 will not normalize.

4.4.4 Normalize agent4: not give good results (like original data).

4.5 Fuzzified agents

All data converted to fuzzy form before processed by agents as fallowing:

4.5.1 Fuzzified classification agent1

The results of training agent1 with fuzzy data by earlier stopping are shown in table 4.59; the results of testing fuzzy test set which is 20% of dataset are shown in confused matrix in table 4.60, and overall accuracy of fuzzy testing is shown in table 4.61, while details accuracy by class of fuzzy test is shown in table 4.62.

Table 4.59: Training the agent1with fuzzy by earlier stopping criteria.

Number of patterns	241
Training time	0.00001 s
Good Iteration	15365
Total Iteration	35366
MSE(training set)	0.16
MSE(validation set)at good iteration	3.58E-10
MSE(validation set)at last iteration	0.00016
Accuracy (training set)	7.05 % (17 patterns)
Error Rate(training set)	92.946 % (224patterns)
Accuracy (validation set)	8.64 % (7patterns)
Error Rate(validation set)	91.36 % (74patterns)

Table 4.60: Confused matrix of fuzzy test agent1 by earlier stopping criteria.

	High	**Medial**	**Low**	**Very Low**	**Actual no.**
High	15	1	10	0	26

Medial	13	0	6	0	19
Low	13	0	7	0	20
Very low	13	0	3	0	16
Total number of predicate	36	0	16	0	81

Table 4.61: Overall accuracy of test the agent1 earlier stopping criteria.

Number of patterns	Overall Accuracy	Error Rate
81	27.16 % (22)	72.84 % (59)

Table 4.62: Accuracy by class of fuzzy agent1 by earlier stopping criteria.

Class	TP rate	FP rate	FN rate	Precision	Recall
High	0.58	1.5	0.42	0.28	0.58
Medial	0	0.05	1	0	0
Low	0.35	0.95	0.65	0.27	0.35
Very Low	0	0	1	-	0

The comparison the original results with fuzzy results of agent1 with earlier stopping are shown in table 4.63.

Table 4.63: Agent1orginal and fuzzy comparison by earlier stopping.

	With fuzzy	Original data
Number of patterns	241	241
Training time	0.00001 s	0.00001 s

Good Iteration	15365	3763
Total Iteration	35366	23764
MSE(training set)	0.16	6.253E-04
MSE(validation set)at good iteration	3.58E-10	9.4567E-09
MSE(validation set)at last iteration	0.00016	3.544E-08
Accuracy (training set)	7.05 % (17 patterns)	97.5% (235 patterns)
Error Rate(training set)	92.946 % (224patterns)	2.5% (6 patterns)
Accuracy (validation set)	8.64 % (7patterns)	85.2 % (69patterns)
Error Rate(validation set)	91.36 % (74patterns)	14.8%(12 patterns)
Overall Accuracy of hard test	27.16 % (22 patterns)	95.06% (77 patterns)

According to results in tables 4.63 fuzzified agent1 will not given best result for overall accuracy, accuracy of training set and validation set, when earlier stopping criteria is used, therefore will not adopted fuzzifid the agent1 with earlier stopping criteria.The results of training agent1 with fuzzy data with cross validation are shown in table 4.64, while the overall accuracy of fuzzy testing by cross validation is shown in table 4.65.

Table 4.64: Agent1 fuzzy training results with cross validation.

Experiments	Training time	Iteration	MSE	Accuracy	Error Rate
Experiment1	0.000	20000	0.22	51.86%(167)	48.14%(155)
Experiment2	0.000	20000	0.17	49.38%(159)	50.62%(163)
Experiment3	0.000	20000	0.10	57.76%(186)	42.24%(136)
Experiment4	0.004	20000	0.089	60.25%(194)	39.75%(128)
Experiment5	0.005	20000	0.15	52.48%(169)	47.52%(153)
Average	**0.0018**	**20000**	**0.15**	**54.53%(175)**	**45.65%(147)**

Table 4.65: Agent1 fuzzy testing results with cross validation.

Experiments	Overall Accuracy	False rate
Experiment1	45.68 % (37 patterns)	45.32 % (44 patterns)
Experiment2	51.85%(42 patterns)	48.15%(39 patterns)
Experiment3	37.04%(30 patterns)	62.97%(51 patterns)
Experiment4	24.69%(20 patterns)	75.30%(61 patterns)
Experiment5	39.51%(32 patterns)	60.49%(49 patterns)
Average	**39.75%(32.2patterns)**	**60.25%(48.8patterns)**

The comparison the original results with fuzzy results of agent1 with cross validation are shown in table 4.66, and 4.67.

Table 4.66: Agent1orginal and fuzzy comparison by cross validation.

	Training time	Iteration	MSE	Accuracy	Error Rate
Fuzzy average	0.0018	20000	0.15	54.53%(175)	45.65%(147)
Original average	0.0032 s	973.8	0.00978	96.832% (311.8)	3.167% (10.2)

Table 4.67: Hard test agent1orginal and fuzzy comparison by cross validation.

	Overall Accuracy	False rate
fuzzy average	39.75%(32.2 patterns)	60.25%(48.8 patterns)
Original average	92.839% (75.2patterns)	7.48%(5.8patterns)

According to results in tables 4.66 and 4.67 also fuzzified agent1 will not given best result for overall accuracy, accuracy of training set, when cross validation is used, therefore will not adopted fuzzifid the agent1 with cross validation technique.

4.5.2 Fuzzified classification agent2

The results of training the agent2 on fuzzy training set by earlier stopping criteria technique shown in table 4.68, the results of testing test set which is 20% of dataset shown in confused matrix in table 4.69, and overall accuracy of fuzzy testing is shown in table 4.70, while The details accuracy by class is shown in table 4.71.

Table 4.68: Training the agent2 with fuzzy data.

Number of patterns	90
Training time	19.0 s
Good Iteration	626
Total Iteration	10627
MSE(training set)	1.14 E-03
MSE(validation set)at good iteration	6.07 E-03
MSE(validation set)at last iteration	9.36 E-02
Accuracy (training set)	98.89% (89)
Error Rate(training set)	1.11%(1)
Accuracy (validation set)	83.33%(25)
Error Rate(validation set)	16.66%(5)

Table 4.69: Confused matrix of test agent2 with fuzzy data.

	Setosa	**Versicolor**	**Virginica**	**Actual no.**
Setosa	8	0	2	10
Versicolor	0	10	0	10
Virginica	1	0	9	10
Total number of predicate	10	10	10	30

Table 4.70: Overall accuracy of fuzzy test the agent2.

Number of patterns	Accuracy	Error Rate
30	90% (27 patterns)	10.0% (3 patterns)

Table 4.71: Accuracy by class of agent2 with fuzzy data.

Class	TP rate	FP rate	FN rate	Precision	Recall
Setosa	0.8	0.1	0.2	0.89	0.8
Versicolor	1.00	0.00	0.00	1.00	1.00
Virginica	0.9	0.2	0.1	0.82	0.9

The comparison the original results with fuzzy results of agent2 with earlier stopping are shown in table 4.72.

Table 4.72: Agent2 original and fuzzy comparison by earlier stopping.

	With fuzzy	Original data
Number of patterns	90	90
Training time	19.0 s	34.0 s
Good Iteration	626	10402
Total Iteration	10627	20403
MSE(training set)	1.14 E-03	0.56 E-04
MSE(validation set)at good iteration	6.07 E-03	0.20 E-04
MSE(validation set)at last iteration	9.36 E-02	0.56 E-04
Accuracy (training set)	98.89% (89)	98.89% (89)
Error Rate(training set)	1.11%(1)	1.11%(1)
Accuracy (validation set)	83.33%(25)	96.67%(29)
Error Rate(validation set)	16.66%(5)	3.33%(1)
Overall Accuracy of hard test	90% (27 patterns)	100%(30 patterns)

According to results in tables 4.72 fuzzified agent2 will not given best result for overall accuracy, accuracy of training set and validation set, when earlier stopping criteria is used, therefore will not adopted fuzzifid the agent2 with earlier stopping criteria. The results of training the agent2 on fuzzy training set by cross validation technique shown in table 4.73, and overall accuracy of fuzzy testing with cross validation is shown in table 4.74, while comparison the original results with fuzzy results of agent2 with cross validation are shown in table 4.75, and 4.76.

Table 4.73: Agent2'S training results with cross validation (fuzzy data).

Experiments	Training time	Iteration	MSE	Accuracy	Error Rate
Experiment1	42.0 s	14852	0.0009	98.33%(118 patterns)	1.67%(2 patterns)
Experiment2	41.0 s	63337	0.0009	100%(120 patterns)	0%(0 patterns)
Experiment3	37.0 s	90000	0.017	96.67%(116 patterns)	3.33% (4 patterns)
Experiment4	25.0 s	90000	0.0011	98.33% (118 patterns)	1.66% (2 patterns)
Experiment5	10.0 s	90000	0.0091	99.16% (119 patterns)	0.83% (1 patterns)
Average	**31.0 s**	**69637.8**	**0.00578**	**98.5%(118.2 patterns)**	**1.83%(2.2 patterns)**

Table 4.74: Overall accuracy of fuzzy test the agent2 with cross validation.

Experiments	Overall Accuracy	False rate
Experiment1	90 % (27 patterns)	10 % (3 patterns)
Experiment2	86.66% (26 patterns)	13.33%(4 patterns)
Experiment3	90% (27 patterns)	10%(3 patterns)

Experiment4	76.66% (23 patterns)	23.33%(7 patterns)
Experiment5	83.33%(25 patterns)	16.66%(5 patterns)
Average	**85.33% (25.6 patterns)**	**14.67%(4.4 patterns)**

Table 4.75: Agent2 original and fuzzy comparison by cross validation.

	Training time	Iteration	MSE	Accuracy	Error Rate
Fuzzy average	31.0 s	69637.8	0.00578	98.5%(118.2 patterns)	1.83% (2.2 patterns)
Original average	9.2 s	5149.8	0.00087	96.17%(115.4)	3.83% (4.6)

Table 4.76: Hard test agent2 original and fuzzy comparison by cross validation.

	Overall Accuracy	False rate
Fuzzy average	85.33%(25.6 patterns)	14.67% (4.4 patterns)
Original average	93.33 % (28 patterns)	6.67% (2 patterns)

According to results in tables 4.75 and 4.76 fuzzified agent2 will not given best result for test set overall accuracy, but improve the accuracy of training set, when cross validation is used, therefore if the accuracy of training set within the critical condition can used fuzzifid data to improve results. In this state can adopted fuzzifid the agent2 with cross validation technique.

4.5.3 Fuzzified classification agent3

The results of training agent3 by earlier stopping criteria fuzzy data are shown in table 4.77, The results of testing test set which is 20% of dataset with fuzzy data shown in confused matrix in table 4.78, and overall accuracy of testing fuzzy data is shown in table 4.79, while details of accuracy by class are shown in table 4.80.

Table 4.77: Training the agent3 with fuzzy by earlier stopping.

Number of patterns	823
Training time	29.00s
Good Iteration	9976
Total Iteration	10001
MSE(training set)	2.10E-15
MSE(validation set)at good iteration	5.95E-16
MSE(validation set)at last iteration	7.25E-16
Accuracy (training set)	85.91%(707 patterns)
Error Rate(training set)	14.09%(116 patterns)
Accuracy (validation set)	63.50%(174 patterns)
Error Rate(validation set)	36.50 %(100 patterns)

Table 4.78: Confused matrix of test agent3 by fuzzy data.

	0	**1**	**Actual no.**
0	145	10	155
1	23	96	119
Total number of predicate	168	106	274

Table 4.79: Overall accuracy of test the agent3 with fuzzy data.

Number of patterns	Accuracy	Error Rate
274	87.96%(241patterns)	12.04%(33patterns)

Table 4.80: Accuracy by class of agent3 by fuzzy data.

Class	TP rate	FP rate	FN rate	Precision	Recall
0	0.94	0.15	0.06	0.86	0.94
1	0.81	0.08	0.19	0.91	0.81

The comparison the original results with fuzzy results of agent3 with earlier stopping are shown in table 4.81.

Table 4.81: Agent3 original and fuzzy comparison by earlier stopping.

	With fuzzy	Original data
Number of patterns	823	823
Training time	29.00s	42.00 s
Good Iteration	9976	10000
Total Iteration	10001	10001
MSE(training set)	2.10E-15	1.26E-09
MSE(validation set)at good iteration	5.95E-16	0.0037
MSE(validation set)at last iteration	7.25E-16	0.0037
Accuracy (training set)	85.91%(707 patterns)	100% (823 patterns)
Error Rate(training set)	14.09%(116 patterns)	0%(0 patterns)
Accuracy (validation set)	63.50%(174 patterns)	100%(274 patterns)
Error Rate(validation set)	36.50 %(100 patterns)	0%(0 patterns)
Overall Accuracy of hard test	87.96%(241patterns)	100%(274 patterns)

According to results in tables 4.81 fuzzified agent3 will not given best results for overall accuracy, accuracy of training set and validation set, when earlier stopping criteria is used, therefore will not adopted fuzzifid the agent3 with earlier stopping criteria. The results of training the agent3 on fuzzy training set by cross validation technique shown in table 4.82, while overall accuracy of fuzzy testing with cross validation is shown in table 4.83.

Table 4.82: Agent3's training results with cross validation (fuzzy data).

Experiments	Training time	Iteration	MSE	Accuracy	Error Rate
Experiment1	0.00	3	0.0013	50.32%(552)	49.68%(545)
Experiment2	0.00	2	0.00046	48.95%(537)	51.05%(560)
Experiment3	0.00	2	0.0018	48.68%(534)	51.32%(563)
Experiment4	0.00	2	0.0012	51.69%(567)	48.31%(530)
Experiment5	0.00	2	0.0018	51.41%(564)	48.59%(533)
Average	**0.00**	**2.2**	**0.0013**	**50.20%(550.8)**	**49.79%(546.2)**

Table 4.83: Overall accuracy of fuzzy test the agent3 with cross validation.

Experiments	Overall Accuracy	False rate
Experiment1	50 % (137 patterns)	50 % (137 patterns)
Experiment2	51.82%(142patterns)	48.17%(132patterns)
Experiment3	49.27%(135patterns)	50.73%(139patterns)
Experiment4	51.82%(142patterns)	48.17%(132patterns)
Experiment5	47.81%(131patterns)	52.19%(143patterns)
Average	50.2% (137.4patterns)	49.85%(136.6patterns)

The comparison the original results with fuzzy results of agent3 with cross validation are shown in table 4.84, and 4.85.

Table 4.84: Agent3 original and fuzzy comparison by cross validation.

	Training time	Iteration	MSE	Accuracy	Error Rate
Fuzzy average	0.00	2.2	0.0013	50.20%(550.8)	49.79%(546.2)
Original average	0.00	2.2	0.0013	96.57%(1059.4)	3.43%(37.6)

Table 4.85: Hard test agent3 original and fuzzy comparison by cross validation.

	Overall Accuracy	False rate
Fuzzy average	50.15% (137.4 patterns)	49.85%(136.6 patterns)
Original average	97.08%(266 patterns)	2.92%(8 patterns)

According to results in tables 4.84 and 4.85 fuzzified agent3 will not given best results for test set overall accuracy, accuracy of training set, when cross validation is used, therefore will not adopted fuzzifid the agent3 with cross validation technique.

4.5.4 Fuzzified Classification Agent4

This agent4 trained on seeds dataset with back propagation learning algorithm in pattern mode but in original data format the agent4 don’t give good results. When fuzzified the agent4 the results become excellent.

The same methodology of the previous agents will be performed, first if generalization is not required (conventional divisions are used), second if generalization required The two techniques (*Earlier Stopping Criteria* and *Regularization*) will applied to select the best one to build the agent4.

4.2.1 Experiments of classification agents (conventional divisions)

The division (70%, 30% and 80%, 20%), where, the results of training the system on the ratio of (70% and 80%) are shown in table 4.86. While the results of testing the system on (30% and 20%) of data are shown in table 4.87.

Table 4.86: Training the system on 70% of data.

The divisions	Number of patterns	Training time	Iteration	Net error	True rate	False rate
70%	147	0.002	776	0.00398	100%	0%
80%	168	0.002	1629	0.00390	97.62%	2.38%

Table 4.87: Testing the system on 30% of data.

The divisions	Number of patterns	True rate	False rate
30 %	63	73.02% (46 patterns)	26.98% (17 patterns)
20 %	42	92.857% (39 patterns)	7.14% (3 patterns)

The results in table 4.87 show that the best division is 80% for training and 20%for testing.

4.2.2 Experiments of classification agents (generalization techniques)

The results of training and testing agent4 with two techniques (*Earlier Stopping Criteria technique* and *Regularization Technique with Jitter*) will be displayed to select the best technique for agent4.

i. Earlier Stopping Criteria

The results of training agent4 by earlier stopping criteria technique are appeared in table 4.88, where the dataset partitioned into 60% for training, 20% for validation and 20% for testing. Likewise the overall accuracy of testing is appeared in table 4.89.

Table 4.88: Training the agent4 by earlier stopping criteria.

Number of patterns	126
Training time	0.00001 s
Good Iteration	693
Total Iteration	20694
MSE(training set)	0.0009
MSE(validation set)at good iteration	0.0098
MSE(validation set)at last iteration	0.03
Accuracy (training set)	98.4 %(124 patterns)
Error Rate(training set)	1.6 %(2 patterns)
Accuracy (validation set)	59.52 %(25 patterns)
Error Rate(validation set)	40.48%(17 patterns)

Table 4.89: Overall test accuracy of agent4 by earlier stopping criteria.

Number of patterns	Overall Accuracy	Error Rate
42	95.23% (40 patterns)	4.77% (2 patterns)

To compare the earlier stopping criteria with regularization and to actual measure of generalization, the noise added to test set, the results of testing the agent4 on test set (with noise) are performed with different ranges of noise, appeared in table 4.90, where each percent is an average of five experiments.

Table 4.90: Estimate the generalization of agent4 by earlier stopping criteria where number of patterns is 42.

Noise range	Overall Accuracy	Error Rate (Misclassification Rate)
0.2_0.1	79.998%	20.002%
0.3_0.1	72.86%	27.14%
0.4_0.1	66.67%	33.33%
0.5_0.1	60.95%	39.05%
0.6_0.1	54.28%	45.72%

ii. Regularization Technique

Select the best training and testing sets by k-fold cross validation (with k=5) to add the noise. The results of training the agent4 by cross validation will be described in table 4.91. Every training set contain 168 (80% of data) patterns. The results of testing the agent4 by cross validation will be described in table 4.92. Where, each testing set contains 42 (20% of data) patterns.

Table 4.91: Agent4 training results with cross validation.

Experiments	Training time	Iteration	MSE	Accuracy	Error Rate
Experiment1	0.002s	3248	0.004	100%(168 patterns)	0%(0 patterns)
Experiment2	0.001s	2787	0.004	100%(168 patterns)	0%(0 patterns)
Experiment3	0.002s	30000	0.019	98.21%(165patterns)	1.79%(3patterns)
Experiment4	0.001s	30000	0.0065	98.2%(165 patterns)	1.78%(3patterns)
Experiment5	0.002s	1675	0.004	99.40%(167patterns)	0.60%(1patterns)
Average	**0.0016**	**13542**	**0.0075**	**99.16% (166.6 patterns)**	**0.83% (1.4 patterns)**

Table 4.92: Agent4 testing results with cross validation.

Experiments	Overall Accuracy	False rate
Experiment1	69.04%(29 patterns)	30.96%(13 patterns)
Experiment2	90.47%(38 patterns)	9.52 %(4 patterns)

Experiment3	92.86%(39 patterns)	7.14%(3 patterns)
Experiment4	88.1%(37 patterns)	11.9%(5 patterns)
Experiment5	100%(42 patterns)	0%(0 patterns)
Average	**88.09%(37 patterns)**	**11.91%(5 patterns)**

The best result is experiment5, where the rate of classification test set is 100 % (42 patterns). According to this, experiment5 will be adopted for this agent to add noise for estimate generalization.

The confused and not confused data is trained by the agent4, and then test the original and confused tested data to estimate and compare the generalization. The results will be shown in table 4.93.

All outcomes in table 4.93 represent overall accuracy of classification and each percent is an average of five experiments. As appeared table 4.93, enhancing generalization is required at all rates of noise, as shown in figure 4.7.

Table 4.93: Estimate the generalization of agent4 (experiment5).

The noise ranges on training and testing data	Soft test	Hard test	Hard test with noise	Hard test with noise (training on original training data)
0.2_0.1	95.14%	94.00%	77.62%	74.28%
0.3_0.1	93.50%	93,33%	72.37%	65.23%
0.4_0.1	88.80%	86.66%	72.37%	60.47%
0.5_0.1	86.19%	83.33%	63.84%	60.47%
0.6_0.1	84.46%	80.95%	61.42%	55.23%

iii. The result of generalization's comparison between earlier stopping criteria and regularization technique (with noised test data) is shown in table 4.94 and figure 4.8.

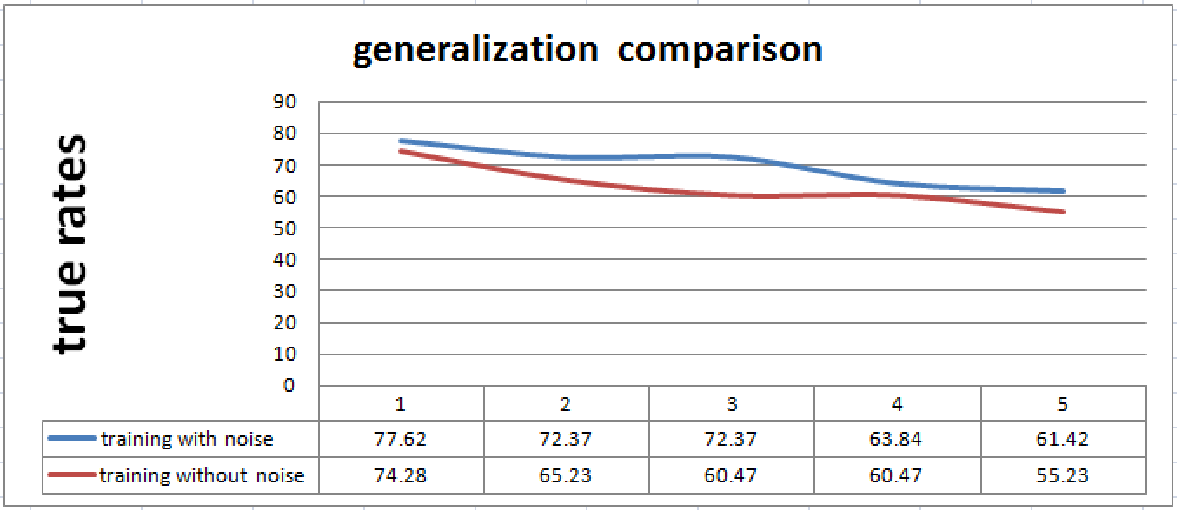


Figure 4.7: Generalization measurements (experiment5).

Table 4.94: Compare the behavior of agent4 by earlier stopping criteria and regularization technique.

Noise range	Earlier Stopping Criteria	Regularization
0.2_0.1	79.998%	77.62%
0.3_0.1	72.86%	72.37%
0.4_0.1	66.67%	72.37%
0.5_0.1	60.95%	63.84%
0.6_0.1	54.28%	61.42%

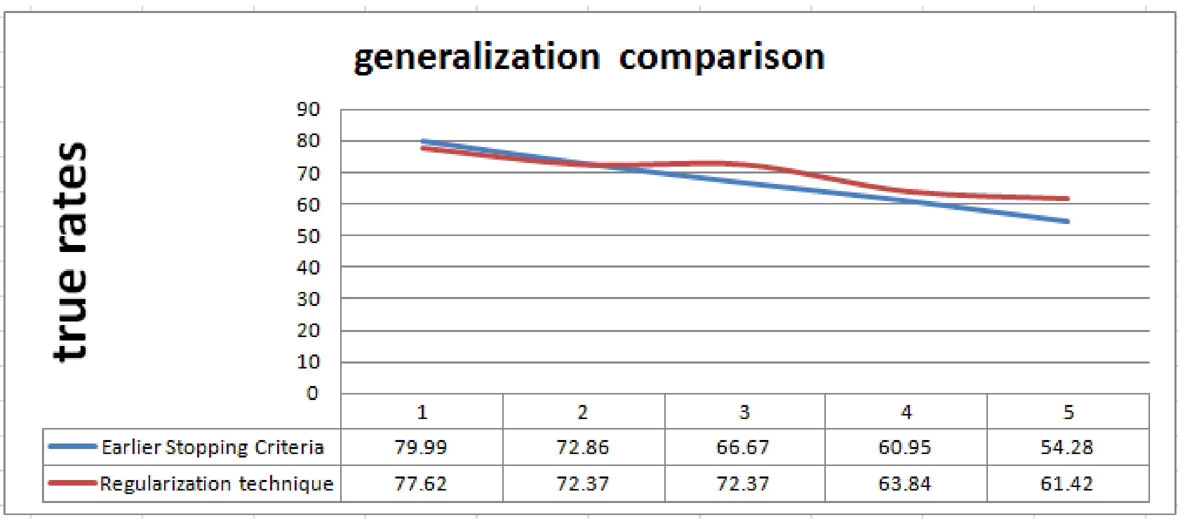


Figure 4.8: Comparison earlier stopping criteria and regularization technique.

This comparison demonstrates that the regularization technique is superior to earlier stopping criteria, where the generalization degree in regularization technique is larger than earlier stopping criteria when the rate of noise (0.1_0.4) and above. Presently the basic condition which is adopted to pick the best technique to build agent4 is the ability to generalization and the results of analysis shown that the regularization technique by adding noise to enhance generalization is the best, as indicated by this investigation, regularization technique will be adopted for classifying agent4.

4.6 Updating control agent data

After building agent4 (with fuzzy data) the control agent data must be updated with seeds data set. Where, the overall accuracy of control agent is 99.76%.

4.7 Summarization of the Experimental Results

Table (4.95) contains summary of the best techniques which adopted to build each agent in the system depend on the experimental results.

The table 4.95 demonstrates the best technique for each classification agent based on generalization criteria .Where; the regularization technique is superior to earlier stopping criteria to build agent1, because the generalization degree in regularization technique is large than earlier stopping criteria at all amounts of noise.

The earlier stopping criteria technique is superior to regularization, to build agent2 and agent3, where the generalization degree in regularization technique is less than earlier stopping criteria at all amounts of noise, therefore earlier stopping criteria technique will be adopted for classifying agent2 and agent3.

Table 4.95: Summary of the best techniques.

Agent number	dataset	Number of patterns	Number of features	Number of classes	Data format	Soft patterns rate	Hard patterns rate	earlier stopping criteria	regularization	Weka results
Agent1	**User knowledge level**	**403**	**5**	**4**	**original**	**80%**	**20%**	**95.06%**	**97.53%**	**97.53%**
Agent2	**Iris**	**150**	**4**	**3**	**original**	**60% training** **20% validation**	**20%**	**100%**	**96.67%**	**100%**
Agent3	**Banknote**	**1372**	**4**	**2**	**original**	**60% training** **20% validation**	**20%**	**100%**	**98.91%**	**-**
Agent4	**Seeds**	**210**	**7**	**3**	**fuzzy**	**80%**	**20%**	**95.23%**	**69.04%**	**-**

With seeds dataset (fuzzy) the regularization technique is superior to earlier stopping criteria, where the generalization degree in regularization technique is larger than earlier stopping criteria therefore regularization technique will be adopted for classifying agent4.

Chapter Five

Conclusions and Future Works

5.1 Conclusions

In the previous chapters, the establishment of the developed classification system was presented, and the effect of various classification techniques on datasets has been illustrated. Several conclusions have been deduced from the obtained test results which are summarized in the following points:

- The control agent classifies collection of datasets and it has ability to recognize each dataset from another in ratio 99.76%.
- The developed system reduces the time and efforts to classify each pattern in a collection of datasets where, best time is the average time of the first agent; worst time is the accumulated time for all agents. In multi layer agent best time = worst time = the average time of the selected agent + time of the control agent.
- The comparisons among experimental results show the regularization technique is suitable for classification agent1 (user knowledge level dataset), as shown in table (4.11).
- Earlier stopping criteria technique is suitable for both classification agent2 and agent3 (iris, banknote datasets), as shown in tables (4.22) and (4.33) respectively.
- Normalize agent1 does not improve the overall accuracy of classification, while normalize agent2 and agent3 give same overall accuracy of original data (100%) for test set.

- Fuzzified agent1 will not enhance overall accuracy, because the nature of data which similar fuzzy form where ranged between zero and one. While fuzzified agent2 will not given best result for overall accuracy, when earlier stopping criteria is used, when cross validation is used fuzzified data improve the accuracy of training set, while do not enhance the overall accuracy of test set, lastly fuzzified agent3 do not enhance overall accuracy, when earlier stopping criteria and cross validation is used.

5.2 Future Work Suggestions

This work can be extended in different directions. In the following some suggested future works are given:

- Use more features of agent technology like increase cooperative by building communication between classification agents and sociality by building communication with other classification systems.
- Try to design a big hierarchal of agents, more than two levels which give amore ability to classify more complex datasets.
- Using different strategies in one agent, like genetic algorithm for optimizing data, and feature selection techniques.

References

[Ali04] Alireza Ferdowsizadeh Naeeni, "**Advanced Multi-Agent Fuzzy Reinforcement Learning**", Master Thesis, Computer Engineering, Dalarna University, Sweden, November 2004.

[Ami11] Amit G., Y. P. Kosta, Gaurang P., Chintan G.," **Initial Classification Through Back Propagation In a Neural Network Following Optimization Through GA to Evaluate the Fitness of an Algorithm**", International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3, No 1, Feb 2011**.**

[Ang09] Agnimitra B., R. Gupta., "**An Artificial Neural Network Based Methodology for the Prediction of Power**", The Open Renewable Energy Journal, Vol. 2, pp. 43-51, 2009.

[Ant13] Antonio Irpino, Mario Rosario Guarracino, and Rosanna Verde.,**" Classification of Data Chunks Using Proximal Vector Machines and Singular Value Decomposition**",, Springer-Verlag Berlin Heidelberg 2013.

[Anu15] Anuj Gupta, **"Classification Of Complex UCI Datasets Using Machine Learning And Evolutionary Algorithms",** International Journal Of Scientific & Technology Research Volume 4, Issue 05, May 2015.

[Chi05] Chia-En Lin, Krishna M. Kavi, "**A Methodology to Evaluate Agent Oriented Software Engineering Techniques**", The International Journal of Multiagent and Grid Systems, Nov. 2005.

[Dav10] David L. Poole, "**Artificial Intelligence Foundations of Computational Agents**", University of British Columbia, David L. Poole and Alan K. Mackworth 2010.

[Dev08] Devendra K. C.," **Soft Computing Techniques and its Applications in Electrical Engineering**", Springer-Verlag Berlin Heidelberg**,** 2008.

[Far00] Farooq A.,**" Biologically Inspired Modular Neural Networks "**, Ph.D. thesis, Virginia Polytechnic Institute and State University, 2000.

[Flo10] Florin LEON **., "Strategy Management In A Multi-Agent System Using Neural Networks For Inductive And Experience-Based Learning",** Management & Marketing Challenges for Knowledge Society Vol. 5, No. 4, pp. 3-28., 2010.

[Fra13] Franck Dernoncourt, "**Introduction to fuzzy logic**", January 2013.

[Ger99] Gerhard Weiss, "**Multiagent Systems:A Modern Approach to Distributed Modern Approach to Artificial Intelligence**", The MIT Press Cambridge, Massachusetts, London, England, 1999.

[Guo00] Guoqiang P. Z.,**" Neural Networks for Classification: A Survey ",** Transactions On Systems, Man, And Cybernetics - Part C: Applications And Reviews, IEEE, Vol. 30, No. 4, November 2000.

[Hai01] Haiping Xu, Sol M. Shatz,**"A Framework for Modeling Agent-Oriented Software"**, IEEE, 2001.

[Han14] Hanaa Mohammed Mashjel., **" General Classification Model Using Cooperative Neural Networks",** Thesis Submitted to the College of Science University of Baghdad 2014.

[Hug15] Hugh Cartwright, **"Artificial Neural Networks "**,Second Edition, Springer Science + Business Media New York, 2015.

[Hya96] Hyacinth S. N.,**" Software Agents: An Overview",** Knowledge Engineering Review, Vol. 11, No 3, pp.205_244, 1996.

[Ins08] Insung J., Gi-Nam W.**," Pattern Classification of Back-Propagation Algorithm Using Exclusive Connecting Network ",** International

Journal of Electrical and Electronics Engineering**,** 2008.

[Jef11] Jeff Heaton., "**Programming Neural Networks with Encog3 in Java**", Heaton Research, 2011.

[Jia15] Jian Jin,1 Ming Li,2 and Long Jin3, "**Data Normalization to Accelerate Training for Linear Neural Net to Predict Tropical Cyclone Tracks**" , Hindawi Publishing Corporation, Article ID 931629, 2015.

[Jia16] Jianye Hao · Ho-fung Leung, **"Interactions in Multiagent Systems: Fairness, Social Optimality and Individual Rationality",** Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg 2016.

[Jye99] J. Yen and R. Langari, PrenticeHall .,**"Fuzzy Logic: Intelligence, Control, and Information",** 1999.

[Kel14] Ke-Lin Du · M. N. S. Swamy., **"Neural Networks and Statistical Learning"**, Springer-Verlag London 2014.

[Ken11] Kenji Suzuki., "**Artificial Neural Networks Methodological Advances And Biomedical Applications**", InTech., 2011.

[Kha08] Khaing W. M., Thinn T. N., **"Optimum Neural Network Architecture for Precipitation Prediction of Myanmar",** World Academy of Science, Engineering and Technology 48, 2008**.**

[Kld06] K. -L. Du , M. N. S. Swamy, "**Neural Networks in a Softcomputing Framework**", Springer-Verlag London 2006.

[Lab14] Labhya Sharma , Utsav Sharma ., **"Neural network based classifier (pattern recognition) for classification of iris data set",** International Journal of Recent Development in Engineering and Technology, Vol. 3, Issue 2, August 2014.

[Lui15] Luis Magdalena José Luis Verdegay Francesc Esteva., **" Enric Trillas:**

A Passion for Fuzzy Sets", Springer International Publishing Switzerland 2015.

[Mic05] Michael Negnevitsky., **"Artificial Intelligence A Guide to Intelligent Systems**", Second Edition Pearson Education Limited 2005.

[Mir15] Miroslav Kubat., "**An Introduction to Machine Learning**", Springer International Publishing Switzerland 2015.

[Moh15] Mohit, Rohit R. V **"Classification of Complex UCI Datasets Using Machine Learning Algorithms Using Hadoop",** International Journal of Computer Science and Software Engineering (IJCSSE), Volume 4, Issue 7, July 2015.

[Muh13] Muhammet Ünal, Vedat Topuz, "**Optimization of PID Controllers Using Ant Colony and Genetic Algorithms**", Springer-Verlag Berlin Heidelberg 2013.

[Mzr11] M. Z. Rehman, N. M. Nawi,**" Improving the Accuracy of Gradient Descent Back Propagation Algorithm (GDAM) on Classification Problems",** International Journal on New Computer Architectures and Their Applications (IJNCAA), 2011.

[Noo13] Noor A. J. Mohammed Almadhkor, "**A Content-Based Spam Filtering Approach Using Artificial Neural Networks**", Master Thesis, Baghdad University, August 2013.

[Oma10] Omaima N. A. AL-Allaf, **"Improving the Performance of Backpropagation Neural Network Algorithm for Image Compression/Decompression System",** Journal of Computer Science , 2010.

[Pat05] Patricia M., Oscar C.,**" Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing"**, Springer-Verlag Berlin Heidelberg, 2005**.**

[Pre12] Preksha Pareek, Dr. Prema K.V, "**classifying the population as BPL or non_BPL using multilayer neural network** ", International Journal of Scientific and Research Publications, Volume 2, Issue 12, December 2012.

[Rem16] Remco R. Bouckaert, **"WEKA Manual"**, University of Waikato, Hamilton, New Zealand,April 14, 2016.

[Rob95] Robert Full´er, "**Neural Fuzzy Systems**", Donner Visiting professor, Abo Akademi University, 1995.

[Ron07] Ronald J. Brachman, Yahoo Research , **"Synthesis Lectures on Artificial Intelligence and Machine Learning",** Intelligent Autonomous Robotics , Peter Stone 2007.

[Rui05] Rui X., Donald W., **"Survey of Clustering Algorithms ",** IEEE Transactions on Neural Networks, Vol. 16, No. 3, May 2005.

[Rus10] Stuart Russel, Peter Norving, "**Artificial intelligence modern approach**", 2010.

[San06] Sandhya Samarasinghe., "**Neural networks for applied science and engineering**", Taylor & Francis Group, 2006.

[San13] Sanghamitra B. - Sriparna S., **"Unsupervised Classification**", Springer-Verlag Berlin Heidelberg 2013.

[Shi10] Shigeo Abe., **"Support Vector Machines for Pattern Classification",** Second Edition, Springer- Verlag London Limited 2010.

[Sol97] J. Sola and J. Sevilla , "**Importance of input data normalization for the application of neural networks to complex industrial problems**", IEEE Transactions on Nuclear Science, vol. 44, no. 3, 1997.

[Stu95] Stuart J. Russell and Peter Norvig., **"Artificial Intelligence A Modern Approach"**, Prentice-Hall, Inc., 1995.

[Sub08] Subrahmanyam Muthukumar.," **The Application Of Advanced Inventory Techniques In Urban Inventory Data Development To Earthquake Risk Modeling And Mitigation In Mid america**", Ph.D thesis in City and Regional Planning, College of Architecture, Georgia Institute of Technology December, 2008.

[Taj11] T.Jayalakshmi, Dr.A.Santhakumaran,**" Statistical Normalization and Back Propagation for Classification ",** International Journal of Computer Theory and Engineering, Vol.3, No.1, February, 2011.

[Tim10] Timothy J. Ross**.," Fuzzy Logic With Engineering Applications"**, John Wiley & Sons, Ltd. 2010.

[Tos08] Toshinori M., **"Fundamentals of the New Artificial Intelligence Neural, Evolutionary, Fuzzy and More",** Second Edition, Springer-Verlag London Limited, 2008.

Appendix A

Table A (1): Agent1 training results with cross validation.

Experiments	Training time (sec)	Iteration	MSE	Accuracy	Error Rate
Experiment1	0.003	782	0.0099	98.137% (316)	1.863% (6 patterns)
Experiment2	0.003	562	0.0099	96.584% (311)	3.416% (11 patterns)
Experiment3	0.003	906	0.0097	96.584% (311)	3.416% (11 patterns)
Experiment4	0.004	2462	0.0099	99.378% (320)	0.621% (2 patterns)
Experiment5	0.003	156	0.0095	93.478% (301)	6.522% (21 patterns)
Average	**0.0032**	**973.8**	**0.00978**	**96.83% (311.8)**	**3.167% (10.2)**

Table A (2): Agent1 testing results with cross validation.

Experiments	Overall Accuracy	False Rate
Experiment1	97.53 % (79 patterns)	2.46 % (2 patterns)
Experiment2	93.83 % (76 patterns)	6.17% (5 patterns)
Experiment3	93.83 % (76 patterns)	6.172 % (5 patterns)
Experiment4	85.19 % (69 patterns)	14.81 % (12 patterns)
Experiment5	93.83 % (76 patterns)	6.17 % (5 patterns)
Average	**92.839%(75.2patterns)**	**7.48%(5.8patterns)**

Table A (3): Estimate the generalization of agent1 (experiment1).

The noise ranges on training and testing data	Soft test	Hard test	Hard test with noise	Hard test with noise (training on original training data)
0_0.03	97.10%	95.80%	95.79%	96.45%
0_0.05	97.13%	96.04%	93.08%	94.31%
0_0.07	96.73%	94.81%	91.60%	94.81%
0_0.09	94.28%	93.82%	88.14%	93.08%
0_0.1	95.46%	95.79%	87.40%	90.11%

Table A (4): Agent2 training results with cross validation.

Experiments	Training time (sec)	Iteration	MSE	Accuracy	Error Rate
Experiment1	6.0	1855	0.00082	90% (108)	10% (12)
Experiment2	2.0	932	0.00088	97.5% (117)	2.5% (3)
Experiment3	2.0	665	0.00088	95.83% (115)	4.16% (5)
Experiment4	31.0	17782	0.0009	99.17% (119)	0.83% (1)
Experiment5	8.0	4515	0.00089	98.33% (118)	1.66% (2)
Average	9.2	5149.8	0.00087	96.17% (115.4)	3.83% (4.6)

Table A (5): Agent2 testing results with cross validation.

Experiments	Overall Accuracy	False rate
Experiment1	90 % (27 patterns)	10 % (3 patterns)
Experiment2	93.33 % (28patterns)	6.67% (2 patterns)
Experiment3	90 % (27 patterns)	10.0% (3 patterns)

Experiment4	96.67 % (29 patterns)	3.33 % (1patterns)
Experiment5	96.67 % (29 patterns)	3.33 % (1patterns)
Average	**93.33 % (28pattern)**	**6.67%(2 patterns)**

Table A (6): Estimate the generalization of agent2 (experiment4).

The noise ranges on training and testing data	Soft test	Hard test	Hard test with noise	Hard test with noise (training on original training data)
0.05_0.1	97.99%	95.33%	94.66%	96.66%
0.05_0.2	97.16%	95.32%	95.33%	95.33%
0.05_0.3	97.24%	97.32%	95.99%	95.99%
0.05_0.4	89.83%	90.66%	89.83%	93.33%
0.05_0.5	82.40%	72.66%	77.99%	89.32%

Table A (7): Agent3 training results with cross validation.

Experiments	Training time (sec)	Iteration	MSE	Accuracy	Error Rate
Experiment1	0.00	3	0.0013	98.27% (1078)	1.73%(19)
Experiment2	0.00	2	0.00046	97.17% (1066)	2.83%(31)
Experiment3	0.00	2	0.0018	94.26% (1034)	5.74%(63)
Experiment4	0.00	2	0.0012	95.72% (1050)	4.28%(47)
Experiment5	0.00	2	0.0018	97.45% (1069)	2.55%(28)
Average	**0.00**	**2.2**	**0.0013**	**96.57% (1059.4)**	**3.43%(37.6)**

Table A (8): Agent3 testing results with cross validation.

Experiments	Overall Accuracy	False rate
Experiment1	97.45% (267patterns)	2.55 % (7 pattern)
Experiment2	98.18 % (269pattern)	1.82 % (5 patterns
Experiment3	94.16 % (258pattern)	5.84 %(16pattern)
Experiment4	96.72 %(265pattern)	3.28 %(9 patterns)
Experiment5	98.91 % (271pattern)	1.09% (3 patterns)
Average	97.08% (266 pattern)	2.92%(8 patterns)

Table A (9): Estimate the generalization of agent3 (experiment5).

The noise ranges on training and testing data	Soft test	Hard test without noise	Hard test with noise	Hard test with noise (training on original training data)
0_0.1	96.74%	97.36%	97.15%	99.04%
0_0.3	97.96%	98.61%	98.75%	98.90%
0_0.7	97.16%	98.83%	98.17%	98.39%
0_1	96.95%	98.83%	97.44%	97.29%
0_1.5	95.53%	98.17%	95.76%	94.67%

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة النهرين
كلية العلوم

مقترح نظام الوكلاء متعدد التصنيف بأستخام الشبكات العصبية

رسالة

مقدمة إلى كلية العلوم / جامعة النهرين
كجزء من متطلبات نيل درجة الماجستير في علوم الحاسبات

من قبل

محمد عبدالعزيز محمد
بكالوريوس جامعة كربلاء ٢٠٠٩

بأشراف
أ .د .بان نديم ذنون

١٤٣٧ **٢٠١٦**

الخلاصة

أخذت تكنولوجيا الوكيل المتعدد دورا هاما في مجال صناعة القرار و تعلم الالة لحل المسائل المعقدة في العالم الحقيقي، فهي تحاكي قابلية الانسان على صنع القرار، حيث أن لها القدره على الاستنتاج والتصرف بشكل مستقل لحل المسائل او لدعم المستخدم البشري.

في هذه الرسالة، تم تطوير نظام تصنيف كفوء يستخدم تكنولوجيا الوكيل المتعدد المعتمده على الشبكات العصبيه و المنطق المضبب، حيث كل وكيل نفذ كشبكة عصبية (دربت باستخدام خوارزمية تعلم الرجوع العكسي). النظام يصنف مجموعه من مجاميع البيانات بشكل كفوء مع درجه معينه من التعميم.

يتكون النظام من مستويين من الوكلاء، المستوى الاعلى يحوي وكيل واحد يعمل كوكيل سيطره، مسئوليته هي تحديد الوكيل المناسب من الوكلاء في المستوى الأدنى الذي يقوم بتصنيف العينه ذات الصله بالاعتماد على خصائص البيانات. إذا لم يتم التعرف على العينه من قبل وكيل السيطرة فسيعلن أنها عينه غيرمعروفه.

النظام المطور أختبر بأستخدام مجموعات بيانات قياسية مختلفة تم الحصول عليها من مخزن تعلم الماكنة (UCI) للتحليل التجريبي لخوارزميات تعلم الماكنة. وهم مستوى معرفة المستخدم، نبات السوسن، قواعد بيانات مصادقة الأوراق النقدية وقواعد بيانات البذور.

تم استخدام تقنيتين هما (معيار التوقف المبكر و تقنية التنظيم) لتقدير تعميم الوكلاء. تم إضافة الضوضاء إلى البيانات لتحسين التعميم، أجريت عدد من التجارب وقدمت مقارنات بين هذه التجارب لتحديد افضل تقنيه لبناء كل وكيل، قياس التعميم و توضيح تاثير تسويه وتضبيب البيانات الاصليه على النتائج من وجهة نظر الدقة والسرعه. أشارت النتائج النهائية الى أن افضل تعميم تقنية لتصنيف قاعدة بيانات مستوى معرفة المستخدم هي طريقة التسوية مع نموذج اختيار التصديق المتقاطع، وأن معيار التوقف المبكر هي الطريقة الفضلى لتصنيف قاعدتي بيانات نبات السوسن مصادقة الأوراق النقدية . بينما طريقة التسوية مع نموذج اختيار التصديق المتقاطع للبيانات المضببة هي أفضل الطرق لتصنيف قواعد بيانات البذور.

أيضاً كانت دقة أختبار كل وكيل تصنيف كالاتي:وكيل التصنيف الاول 97.53% , وكيل التصنيف الثاني 100%, وكيل التصنيف الثالث 100% و وكيل التصنيف الرابع 96.04%.