MACHINE LEARNING ALGORITHMS AND FUZZY NEURAL NETWORKS: AN EXPERIMENTAL COMPARISON

By

KHIAN THONG LIM

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1993

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 1996

MACHINE LEARNING ALGORITHMS AND

FUZZY NEURAL NETWORKS: AN

EXPERIMENTAL COMPARISON

Thesis Approved:

Blayne E. Mayfield

Thesis Advisor

Mitchell L Neilsen

J Chandler

Thomas C. Collins

Dean of the Graduate College

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor, Dr. Blayne Mayfield for his excellent guidance, encouragement, support, and patience. My sincere appreciation extends to my other committee members Dr. Mitch Neilsen and Dr. John Chandler, whose guidance, assistance, and encouragement are also invaluable. I would like to thank Dr. Donald Tveter, Dr. Ron Kohavi, and Mr. Marc Theisen for providing the programs and assistance.

Most of all, I would like express my special appreciation to my wife, Lynn, for her strong encouragement and support at times of difficulty, love, and understanding throughout this whole process. Thanks also go to my mother, brothers and sisters for their endless support and encouragement.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF GRAPHS

Graph Page

CHAPTER 1

INTRODUCTION

What is Artificial Intelligence (AI)? One of the most widely accepted definitions of AI is by Minsky, as quoted by Yazdani [Yazdani86]: *"AI is the science of making machines do things that would require intelligence if done by men."* In other words, AI is concerned with constructing programs that behave like people. The programs will have the ability to associate with human beings, such as understanding natural language, reasoning, solving problem, and learning.

One area of AI research is machine learning. Much research has been done to understand the nature of learning and implement learning capabilities in machines. Research has shown that learning manifests itself as a spectrum of information processing activities ranging from the direct memorization of facts and acquisition of simple skills by imitation to very intricate inferential processing leading to creation of new concepts and discovery of new knowledge [Michalski86a].

Machine learning is one of the methods that has been used to reduce uncertainty in problem solving and decision making by capturing knowledge from data or examples. Krisar [Krisar95] quotes Simon's definition of learning: "*Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task* *or tasks drawn from the same population more efficiently and more effectively the next time.*"

Algorithms that learn from examples can be divided into two categories [Zheng93]:

- symbolic algorithms, such as ID3 [Quinlan86], C4.5 [Quinlan93], CART [Breiman84], PLS1 [Rendell83], the AQ family of algorithms [Michalski80], and
- subsymbolic algorithms, such as backpropagation [Rumelhart86], Perceptron learning procedure [Rosenblatt62], IB1, IB2, IB3 [Aha91], MDLA B, C [Cameron-Jones92], Nearest Neighbor, Bayes, and Linear Discriminant [Duda73].

A general definition for both symbolic and subsymbolic algorithms can be stated as follows: symbolic algorithms are those in which the learned theories and representation of knowledge can be understood by a human being; other algorithms are subsymbolic. The problem most often addressed by both types of algorithms is the inductive acquisition of concepts from examples. Mooney [Mooney91] defines the problem as follows: "*Given descriptions of a set of examples each labeled as belonging to a particular class, determine a procedure for correctly assigning new examples to these classes.*"

1.1 The Problem

Despite the fact that many learning algorithms and Fuzzy Neural Networks (FNNs) [Halgamuge94b] learn from classified examples, very little is known regarding their comparative strengths and weaknesses. Many comparisons have been done to study the differences between the symbolic and neural networks algorithms (Cheng94, Dietterich90, Fisher89, Holte93, Mooney91, Weiss89, etc.). However, until recently there has been little comparison done between the Induction of Decision Trees (ID3), Feedforward Neural Networks that are trained with Backpropagation (FFNBPs), and Fuzzy Neural Networks (FNNs) learning algorithms (Halgamuge93a, Halgamuge93b, Halgamuge94a, and Halgamuge94b).

1.2 The Purpose of the Experiment

The main purpose of this experiment is to compare the ID3, FFNBP, and FNN algorithms. These algorithms are tested using twelve data sets obtained from the University of California-Irvine Machine Learning Database Repository [Murphy91]. The collected results are analyzed and used to compare these algorithms.

1.3 The Objective of the Experiment

The main objective of the research is to understand more about the comparative strengths and weaknesses between ID3, FFNBP, and FNN. By comparing the algorithms, this experiment may assist others in determining the best algorithm to use in problem solving.

1.4 Outline of Work

This work is organized into five chapters. The next chapter contains discussions of the learning algorithms used in the experiment and some related works that have been done by others. The third chapter introduces the data sets and method of analysis and metrics for comparing for the algorithms used in this work. Chapter Four discusses the results of the experiments and Chapter Five is a summary and discusses future work.

CHAPTER 2

LITERATURE REVIEW

All learning systems have the same goals: to deal with complex, real-world decision-making problems and to solve these problems in the sense of reaching correct conclusions [Weiss90]. Many different approaches to machine learning have been developed and applied to problems from a variety of fields such as medical science, biology, controls, and linguistics.

The learning algorithms chosen for this experiment are Induction of Decision Trees, Feedforward Neural Network with Backpropagation, and FuNe-I .

2.1 Induction of Decision Trees

Induction of Decision Trees (ID3) [Quinlan86] is a simple and widely used symbolic learning algorithm. Symbolic learning techniques are based on learning strategies such as rote learning, learning from examples, learning by being told, learning by analogy, and so on [Carbonell83]. The learning strategy used by ID3 is learning from examples.

ID3 takes objects of a known class and describes them in terms of a fixed collection of properties or attributes. It produces a decision tree that incorporates these attributes to correctly classify other objects. Quinlan describes the basic structure of ID3 as follows [Quinlan86]:

> *"The basic structure of ID3 is iterative. A subset of the training set called window is chosen at random and a decision tree formed from it; this tree correctly classifies all objects in the window. All other objects in the training set are then classified using the tree. If the tree gives the correct answer for all these objects then it is correct for the entire training set and the process terminates. If not, a selection of the incorrectly classified objects is added to the window and the process continues."*

ID3 employs top-down induction of decision trees (a greedy, divide-and-conquer method) to induce decision trees [Quinlan86]. The method outline is as follows [Murthy94]:

1. Begin with a set of examples called the training set, T. If all examples in T belong to one class, then halt.
2. Consider all tests that divide T into two or more subsets. Score each test according to how well it splits up the examples.
3. Choose ("greedily") the test that scores the highest.
4. Divide the examples into subsets and run this procedure recursively on each subset.

A decision tree consists of nodes and branches. A new node is added to the tree by partitioning the training examples based on their value along a single, most-informative attribute [Mooney91]. The attribute chosen is the one that minimizes the following function:

$$E(A) = -\sum_{i=1}^{V}\frac{S_i}{S}\sum_{j=1}^{N}\frac{k_{ji}}{S_i}\log_2\frac{k_{ji}}{S_i} \qquad \text{(Eq. 1)}$$

where v = number of values for attribute A,

k_{ji} = number of examples in the *j*th category with the *i*th value for attribute A,

S = total number of examples,

S_i = number of examples with the *i*th value for attribute A, and

N = number of categories.

E(A) is the expected information (see Section 3.1.11 for more details about the expected amount of information such as entropy) required for selecting an attribute as the root of a decision trees or partitioning the training examples. The partitioning process is recursive; it stops when each partition contains examples of only a single category. Then, the algorithm creates a leaf for each partition and labels it with the category.

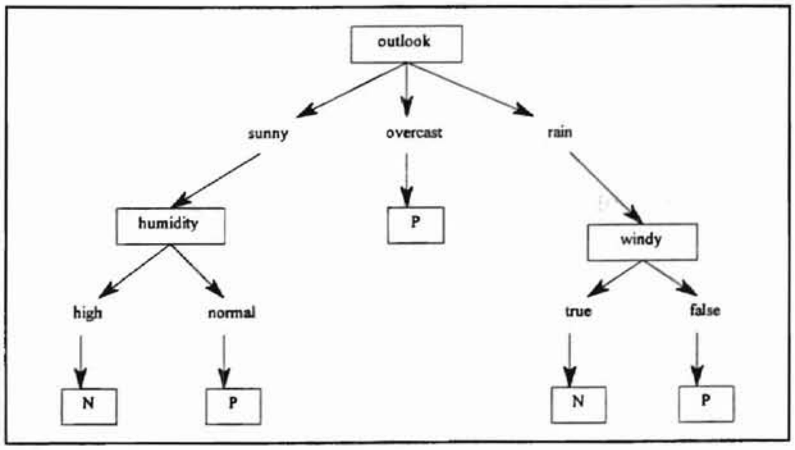


Figure 1: A simple decision tree [Quinlan86]

Table 1: A small training set [Quinlan86]

No.	Attribute Outlook	Attribute Temperature	Attribute Humidity	Attribute Windy	Class
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	sunny	hot	high	false	P
4	overcast	mild	high	false	P
5	rain	cool	normal	false	P
6	rain	cool	normal	true	N
7	rain	cool	normal	true	P
8	overcast	mild	high	false	N
9	sunny	cool	normal	false	P
10	sunny	mild	normal	false	P
11	rain	mild	normal	true	P
12	overcast	mild	high	true	P
13	overcast	hot	normal	false	P
14	rain	mild	high	true	N

Figure 1 illustrates an example of a simple decision tree. The tree is built based on a small training data that uses the 'Saturday Morning' training set given in Table 1 [Quinlan86]. P and N are the classes or categories. The objects of classes P and N are referred to as positive instances and negative instances, respectively, of the concept being learned.

2.1.1 Author

The ID3 program used in this experiment is provided by Dr. Ron Kohavi, who wrote it when he was a Ph.D. student at Stanford University. It is part of the Machine Learning library in C++ (MLC++) package developed by Dr. Kohavi and his colleagues. MLC++ is a library of C++ classes and tools for machine learning algorithms that employ *supervised learning* techniques (see Section 2.2). MLC++ is written in the C++.

2.2 Feedforward Neural Network with Backpropagation

Artificial neural networks are densely interconnected networks of many simple computational neurons. Figure 2 depicts one such neuron. The input to a neuron consists of n values, x_0, x_1, ..., x_{n-1}, each of which is associated with a weight, w_0, .., w_{n-1}, respectively. The neuron computes the weighted sum x of its inputs and passes the result to its activation function. The weighted sum is calculated using the following equation:

$$x = \sum_{i=0}^{n-1} x_i w_i \qquad \text{(Eq. 2)}$$

The activation function is usually the sigmoid function as shown below:

$$f(x) = \frac{1}{1+\exp(-x)} \qquad \text{(Eq. 3)}$$

The value of the activation function is the output, $y=f(x)$, of the neuron as shown in Figure 2.

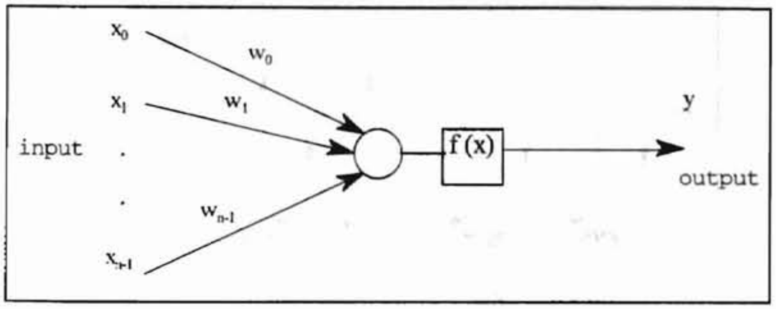


Figure 2: A neuron

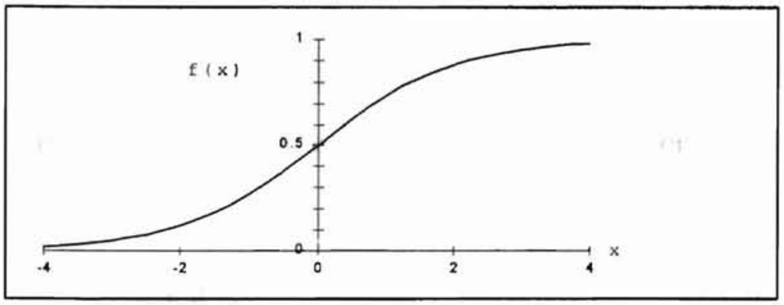


Figure 3: A sigmoid function

Supervised learning is a procedure that adjusts weights on the basis of the difference (or error) between the *actual output* (the network's output) and the *desired output*, given an input pattern and the desired output pattern [Zeidenberg90]. The input pattern and desired output pattern together are called a *training pair*. *Backpropagation* (also called the *generalized delta rule*) is a supervised learning procedure. It uses a *gradient descent* method in an attempt to minimize the sum of squares of the errors across all the training pairs.

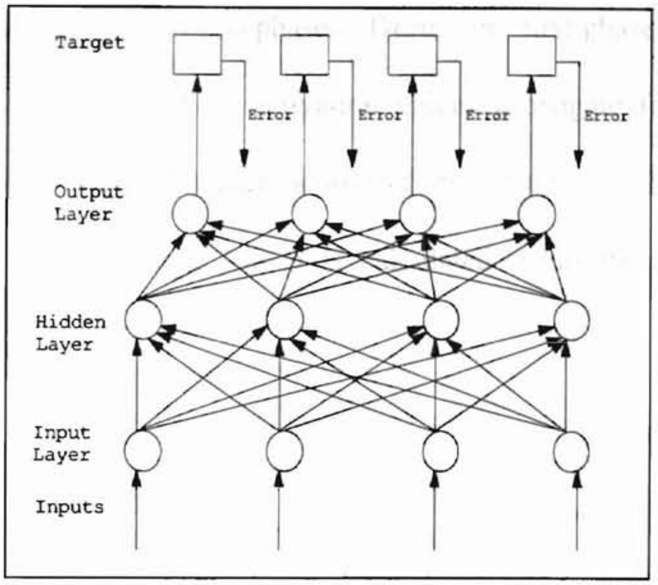


Figure 4: Feedforward neural network with backpropagation [Wasserman89]

A feedforward neural network (FFNN) is a multilayer neural network with three or more layers. FFNNs are usually trained with the backpropagation procedure, and are called FFNBP in this paper. Figure 4 depicts a three-layer FFNBP.

Each neuron in any given layer is like the one shown in Figure 2: it accepts input from previous layer and passes its activation value (or output) to each neuron in the next layer. The first layer (the bottom layer in Figure 4) is called the input layer. The input for each neuron in this layer is one of the values that comprises the input pattern of a training pair. Thus, the number of neurons in the input layer is equal to the size of the input pattern. The input layer neurons are non-computational; that is, they simply pass their input value to the neurons of the next layer. The last layer (the top layer in Figure 4) is called the output layer. The neurons in this layer compute the outputs of the network. All layers between the input and output layers are called hidden layers. The neural net in Figure 4 has one hidden layer.

Backpropagation works in two phases. During the first phase, an input pattern is presented to the input layer and all activation values propagate forward through the network to compute the output for each neuron in the output layer. For each unit *j*, its actual output o_{pj} for an training pair *p* is calculated using the following equation [Mooney91]:

$$o_{pj} = \frac{1}{1+e^{-(\sum_i w_{ji} o_{pi} + \theta_j)}} \qquad \text{(Eq. 4)}$$

where w_{ji} = the weight from neuron *i* (previous layer) to neuron *j* (current layer) and

θ_j = a tunable threshold or bias for neuron *j*.

Equation 4 is a modified version of Equation 2 that works in a layered network. The actual output of the neurons in the output layer is then compare with the desired output pattern (the target in Figure 4) to compute the error.

The second phase is the backpropagation of errors through the network. Beginning at the output layer, error is measured layer by layer and passed backwards through the network. The weights are changed appropriately to reduce the error. The following functions show changes of weight to the unit [Mooney91]:

$$\Delta_p w_{ji} = \eta \delta_{pj} o_{pi} + \alpha \Delta_{p-1} w_{ji} \qquad \text{(Eq. 5)}$$

where $$\delta_{pj} = o_{pj}(1 - o_{pj})(t_{pj} - o_{pj}) \qquad \text{if j is an output unit} \quad \text{(Eq. 6)}$$

and $$\delta_{pj} = o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{kj} \qquad \text{if j is a hidden unit} \quad \text{(Eq. 7)}$$

where η = learning rate,

α = momentum term,

k = neurons (next layer) that neuron j directly connects to,

t_{pj} = target output for the output neuron j for pattern p, and

δ_{pj} = error of the output of neuron j for pattern p.

The process is repeated for each training pair. When all the training pairs have been presented to the network, this is called one epoch (one iteration). The learning process terminates when the overall error of the network is acceptably low or training reaches some predetermined maximum number of epochs (iterations).

One drawback of the FFNBP is that it requires extensive experimentation to find values for parameters such as the learning rate and momentum to achieve optimal results. The learning rate is a constant that expresses the proportionality of *the change in weight to the derivative of the error of training pair p with respect to the weight* [Rumelhart86]. In other words, the learning rate is a constant that controls the amount of changes made to the weights in a single pass. Momentum is a constant that determines the effects of past weight changes on the current direction of movement in weight values.

The FFNBP also has problems with local minima and oscillation. Backpropagation employs a type of gradient descent technique. It follows the slope of the error surface downward, adjusting the weights toward a minimum error. The network can become trapped in a local minimum even though there may be a much deeper minimum nearby. Statistical training methods can help avoid entrapment by local minima, but such processes are slow. Oscillation can occur when the learning rate is too large. The network may oscillate around the global minimum without reaching it. The momentum parameter helps prevent oscillation by adding a term to the weight adjustment that is proportional to the amount of the previous weight change.

2.2.1 Author

The FFNBP program used in this experiment is provided by Dr. Donald Tveter. The program is written in C.

2.3 Fuzzy Neural Networks

Fuzzy set theory was first introduced by Zadeh [Zadeh65]. Fuzzy set theory is a superset of conventional set theory. Fuzzy logic, a subset of fuzzy set theory, is conventional logic extended to handle the concept of partial truth by incorporating "a degree of truth". One area of fuzzy logic application is the integration of fuzzy logic and neural networks known as Fuzzy Neural Networks (FNNs). FNNs seek to maximize of the desirable properties and reduce the disadvantages of each of these systems. The integration provides the low-level learning and computational power of neural networks and the high-level reasoning of fuzzy systems [Lin92]. Neural networks and fuzzy systems have the ability to deal with uncertainty, imprecision, and noise in real-world environment [Lin92].

Over the years, many fuzzy neural networks have been proposed and developed using different learning techniques (i.e., supervised, unsupervised, and competitive) [Halgamuge94b]. Wang et al. [Wang92a] used a backpropagation algorithm to train their three-layer feedforward fuzzy network as an identifier for nonlinear dynamic systems. Roan et al. [Roan93] developed a supervised fuzzy neural network called Fuzzy Restricted Columb Energy (Fuzzy RCE) network for classification problems. Fuzzy RCE can learn very fast and has good recognition performance.

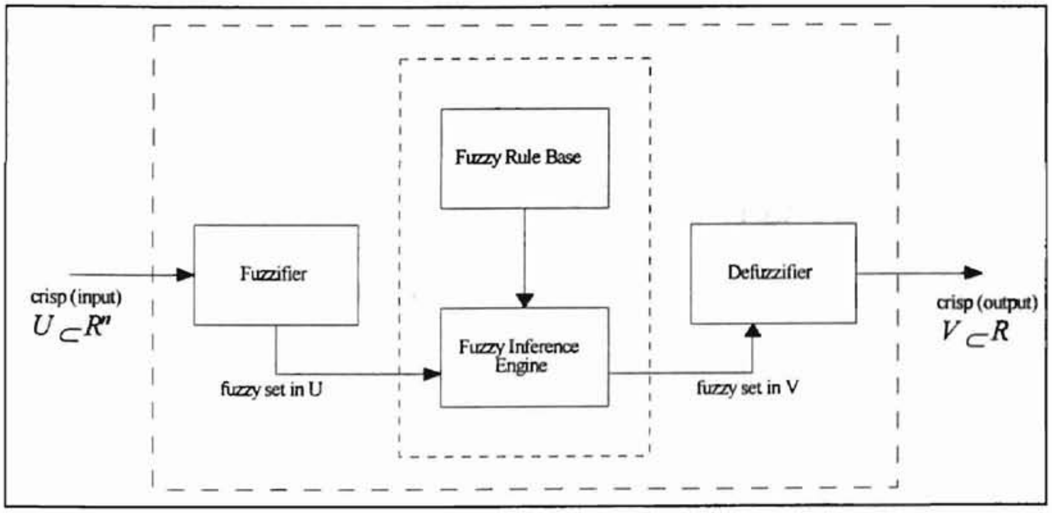


Figure 5: Basic configuration of fuzzy systems

2.3.1 Fuzzy System

A fuzzy system usually has four principal elements [Wang92a] [Wang92b] as depicted in Figure 5: a fuzzifier, a fuzzy rule base, a fuzzy inference engine, and a defuzzifier. The fuzzy systems defined by Wang have multiple inputs and a single output: $U \subset R^n \rightarrow R$, where U is compact and R is real number [Wang92a]. A compact set is defined as [Rudin64]: *"A subset K of a metric space X is said to be compact if every open cover of K contains a finite subcover."* A multiple outputs system can always be divided into a group of single-output systems.

The fuzzifier maps the non-fuzzy inputs ($U \subset R^n$) to fuzzy sets defined in U. A fuzzy set [Zadeh65] is characterized by a membership function μ_F: $U \rightarrow [0,1]$ and labeled by a linguistic term F such as "Low", "Medium" or "High". The *Singleton fuzzifier* is the most commonly used fuzzifier [Wang92a]. It maps $x \in U$ into fuzzy set A_x in U with $\mu_{Ax}(x)=1.0$ and $\mu_{Ax}(x')=0.0$ for all $x' \in U$ with $x' \neq x$.

The fuzzy rule base is a set of linguistic rules in the form of *"IF a set of conditions are satisfied, THEN a set of consequences are inferred."* Given the fuzzy rule base consists of M rules, the format of the rules can be shown as follows:

Rule j: IF x_1 is A_1 and x_2 is A_2 and ... and x_n is A_n, THEN z is B,

where j = 1,2,..,M, x_i (i=1,2,..,n) are the input variables to the fuzzy system, z is the output variable of the fuzzy system, and A_i and B are linguistic terms characterized by membership function $\mu_{A_i}(x_i)$ and $\mu_B(z)$, respectively. Three example rules with two input variables and an output variable:

Rule$_1$: IF x_1 is *high* and x_2 is *high*, THEN z is *high*,

Rule$_2$: IF x_1 is *low* and x_2 is *low*, THEN z is *low*, and

Rule$_3$: IF x_1 is *low* and x_2 is *high*, THEN z is *medium*.

Linguistic terms *low, medium,* and *high* are characterized by their membership functions $\mu_{low}(x_i)$, $\mu_{medium}(x_i)$, and $\mu_{high}(x_i)$ respectively.

The fuzzy inference engine uses fuzzy rules from the fuzzy rule base to determine the fuzzy outputs. The defuzzifier transforms the fuzzy outputs to non-fuzzy (i.e., crisp) outputs, which are the actual outputs of the system.

2.3.2 FuNe-I

FuNe-I is used as the representative of FNNs in this experiment due to its flexibility and good performance. FuNe-I is an integration of fuzzy system and feedforward neural networks [Halgamuge94b]. It extracts fuzzy rules from training data set and tunes its parameters to obtain optimal results by supervised learning.

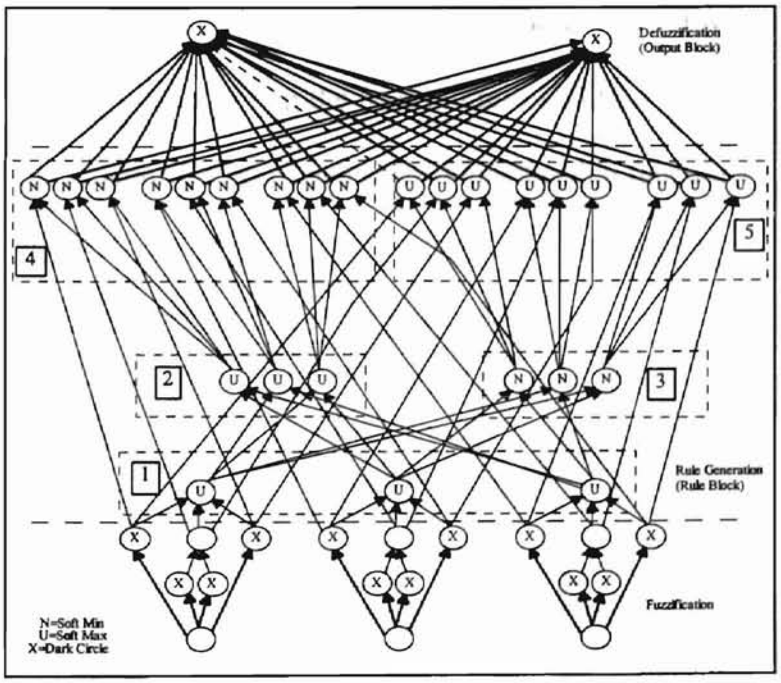


Figure 6: FuNe-I training network [Halgamuge94b]

Basically, FuNe-I works in two phases. In the first phase, FuNe-I generates rules from a training network. It tunes the parameters and the extracted rules in the second phase using the fuzzy system. Figure 6 depicts the basic structure of the FuNe-I training network. FuNe-I training network consists of four neuron blocks: the input block, the fuzzification block, the rule block, and the output block. The input block is not shown in Figure 6.

2.3.2.1 Fuzzification

The input block distributes and feeds the input signals to the fuzzification block where the membership functions are realized. A multilayer neural network is able to obtain virtually any membership function [Halgamuge94b]. However, using a smaller number of neurons, and exploiting the possibilities of shifting, scaling, and reflecting the sigmoid transfer functions, a set of bell shaped membership functions can be created.

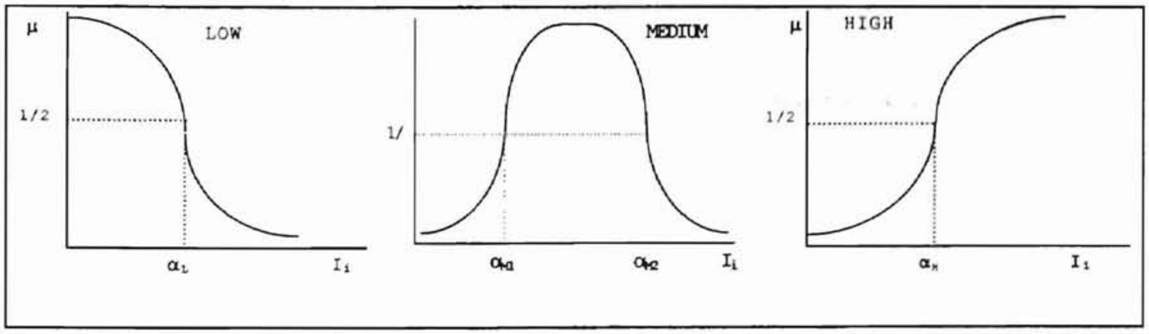


Figure 7: Membership functions [Halgamuge94b]

Considering three possible adjectives: Low (L), Medium (M), and High (H), the formation of the memberships is illustrated in Figure 7. Only one sigmoid function is needed for creating a low or high membership function. However, two sigmoid functions are needed to build a membership function in the range. The following example is taken from Halgamuge [Halgamuge94b]:

$$a_1[I_i,C,\alpha]=\frac{1}{1+e^{-C(I_i-\alpha)}} \quad , \quad a_2[I_i,C,\alpha]=\frac{1}{1+e^{C(I_i-\alpha)}} \qquad \text{(Eq. 8, 9)}$$

Sigmoid a_1 (Equation 8) is the mirror reflection of the sigmoid a_2 (Equation 9) on the Y axis. C is a positive variable used to change the steepness of the sigmoid curves. α is a positive variable employed in shifting the sigmoids. I is the input.

- Low:

  $L = a_2[I_i,C_L,\alpha_L]$

- High:

  $H = a_1[I_i,C_H,\alpha_H]$

- Medium:

  First case: Uses two sigmoid neurons from both types. A third linear neuron with F=Min is connected to the two sigmoid neurons by fixed connection weights of unity.

  $M1 = a_1[I_i,C_{M1},\alpha_{M1}]$
  $M2 = a_2[I_i,C_{M2},\alpha_{M2}]$
  $M = \text{Min}\{M1,M2\}$

  Second case: Subtracts one shifted sigmoid neuron from another shifted sigmoid neuron using a third linear neuron.

  $M1 = a_1[I_i,C_{M1},\alpha_{M1}]$

$M2 = a_1[I_i, C_{M2}, \alpha_{M2}]$

$M = \Sigma\{M1, -M2\}$

It is to be noted that $\alpha_{M1} < \alpha_{M1}$ in both cases.

2.3.2.2 Inference Process and Defuzzification

The rule block receives all the membership values. It creates the rules and passes the rule strengths (described in later paragraph) to the defuzzification block. The rule strengths are weighted and directly summed up to the output neurons with sigmoid activation functions [Halgamuge94b].

FuNe-I constructs and tunes the fuzzy system from data sets. The fuzzy system used by FuNe-I is trained with a gradient-descent method, the backpropagation algorithm. Three types of fuzzy rules are considered for describing the fuzzy system [Halgamuge93a]:

- simple rule with premises containing a single fuzzy variable,
- conjunctive rules with two or more fuzzy variables in premises, and
- disjunctive rules with two or more fuzzy variables in premises.

The following example (taken from Halgamuge) illustrates the inference process and defuzzification of the weighted sum of the rules [Halgamuge93a]:

> Assume for example rules R1, R2, and R3 and inputs X=A1 and Y=B1. Out1, Out2, and Out3 are the outputs and L, M, and H are Low, Medium, and High respectively:

R1: If X is L AND Y is M THEN Out1 is H

R2: If X is H OR Y is L THEN Out2 is M

R3: If X is M THEN Out3 is L

If the membership functions of X and Y are μ_{Lx}, μ_{Mx}, μ_{Hx}, $\mu_{Ly,}$ and μ_{My}, then the evaluation of the antecedents for rules R1, R2, and R3 are:

$K_1 = \cap\ (\mu_{Lx}(A1), \mu_{My}(B1))$

$K_2 = \cup\ (\mu_{Hx}(A1), \mu_{Ly}(B1))$

$K_3 = \mu_{Mx}(A1)$

where $\cap$ and $\cup$ are fuzzy set conjunction and disjunction denoted as T-norm and T-Conorm, and K1 K2, and K3 denote the strengths of rule R1, R2 and R3, respectively. Soft Min (Equation 10) operation is used as fuzzy set conjunction and Soft Max (Equation 11) operation is used as fuzzy set disjunction.

$$Min_K(I_1,..,I_n) = \frac{\sum_{i=1}^{n} I_i \cdot e^{-K \cdot I_i}}{\sum_{i=1}^{n} e^{-K \cdot I_i}} \qquad \text{(Eq. 10) and}$$

$$Max_K(I_1,..,I_n) = \frac{\sum_{i=1}^{n} I_i \cdot e^{K \cdot I_i}}{\sum_{i=1}^{n} e^{K \cdot I_i}} \qquad \text{(Eq. 11)}$$

where $K \geq 1$ and

I_n = nth inputs.

The parameter K controls the hardness of the Soft Min and Soft Max operations. As $K \rightarrow \infty$, Soft Max (Soft Min) operates virtually like usual max (min) operation. The weighted sum of the fired rules is translated into the crisp output (i.e., defuzzified) using the following sigmoid function:

$$\text{Out}_{\text{i}} = a_i(\sum_{j=1}^{r} W_{ij} * K_j) \qquad \text{(Eq. 12)}$$

where $\quad \text{a}_{\text{i}}$ = activation of output neuron (sigmoid function),

r = number of rules,

W_{ij} = weight of the connection from *j*th rule node to the *i*th output, and

K_{j} = strength of *j*th rule node.

2.3.2.3 Automatic Rule Generation

As described earlier, FuNe-I is capable of extracting fuzzy rules from an input data set. The FuNe-I fuzzy-neural model is similar to the Horikawa method of evaluation of the premise and the creation of membership functions except the initial rule base [Halgamuge94b]. A different approach is used to find the initial rule base by FuNe-I. It first identifies the conjunctive and disjunctive rules for each output neuron. This method reduces the size of the initial rule drastically. Then, the training network of FuNe-I is trained with its membership functions to generate rules learn from the input data set.

Figure 6 depicts the FuNe-I training network. The dark lines in the fuzzification and defuzzification blocks represent variable weights; other connections have fixed unity weights. The circles with U(N) represent neurons that have Soft Max (Soft Min) operations. The circles with X represent neurons with sigmoid activation functions; other neurons have linear activation functions.

The rule generation block in Figure 6 contains three layers of neurons for extracting conjunction and disjunction rules. Five sections (1-5) in the rule block have been identified for discussion. The first layer (first section) consists of M neurions (the number of inputs to the network). Each neuron in this section selects the maximum of the membership values of its inputs by using Soft Max. The second layer consists sections 2 and 3. Each of these sections contains M neurons that estimate the maximum (Soft Max) and minimum (Soft Min) of all the neurons in the first layer with some exceptions since the first and second layers are not fully connected. The third layer also has two sections (4 and 5), each of them with 3*M sigmoid neurons. Each of these sections builds the antecedents for conjunction and disjunction rules for the fuzzy rule base. Based on the weights between the output block and the last layer of the rule block, two tables containing the nodes for conjunction and disjunction rules are created.

During the second phase, all layers in the rule block are deleted and two new layers are created for conjunction and disjunction rules. All possible conjunction and disjunction rules with two variables are created using the tables obtained during the first phase. After both phases, the generated rules can be analyzed and weak inputs can be removed.

2.3.2.4 Other Features

Optimization is needed if the number of initial rules generated by a fuzzy system is too large for particular applications. Membership functions illustrated in section 2.3.2.1 can be tuned using training data by changing α values to create a shift in membership function. *C* values can be used to tune the slopes of sigmoid curves of the membership functions. Expert knowledge can be integrated into FuNe-I as fuzzy rules and membership functions to speed up the training and increase performance of the system.

2.3.3 Author

FuNe-I was developed by Dr. S.K. Halgamuge and his colleagues at Technical University of Darmstadt (THD), Darmstadt, Germany. The program was provided by Marc Theisen at THD. The program is written in C.

2.4 Related Works

Many comparisons of learning algorithms have been reported since the development of the first learning algorithms and other statistical based algorithms. Mooney et al. [Mooney91] compared ID3 [Quinlan86] with perceptron [Rosenblatt62] and FFNBP algorithms [Rumelhart86]. Mooney et al. found that the overall performance of the algorithms is comparable. However, FFNBP is more adaptive on noisy data sets though it is slower. Fisher and McKusick compared batch learning backpropagation and ID3; their results support Mooney's conclusion [Chen94]. Chen et al. [Chen94] compared the prediction performance of human experts with ID3 and backpropagation neural networks to predict winners in greyhound racing. In terms of prediction accuracy and monetary payoff, both the learning algorithms performed better than human experts. They found that ID3 predicts more conservatively, and backpropagation is slow but makes excellent predictions for long shots. Weiss and Kapouleas [Weiss89] used a resampling technique such as leave-one-out for evaluation and found those discriminant analysis methods, FFNBPs, and decision trees based learning algorithms achieve comparable performance. Dietterich et al. [Dietterich90] compared ID3 and backpropagation on the NetTalk data sets [Murphy91] and found that backpropagation can capture statistical information that is not captured by ID3. Holte [Holte93] concluded that on most data sets, very simple rules based on a single attribute are as accurate as the rules created by the majority of machine learning systems when he compared 1R with C4.5 [Quinlan93]. C4.5 is a decision tree learning algorithm that is based on ID3.

CHAPTER 3

METHODOLOGY

Many learning algorithms have been compared and evaluated. However, the comparisons often share one common problem: only a few domains are used in each comparison and different comparisons use different domains. Most of the algorithms perform differently on different domains. "*It's very hard to judge an algorithm by seeing its performance on only a few arbitrarily selected domains*", said Zheng [Zheng93]. For that reason, Zheng developed a benchmark for comparing and evaluating learning algorithms.

3.1 Benchmark for Classifier Learning

Learning algorithms are developed to solve real-world problems. Thus, data sets from real-world domains are used when analyzing and comparing learning algorithms. Zheng selected thirteen real-world and synthetic data sets from different domains from the University of California-Irvine Repository of Machine Learning Databases (UCI-RMLD) [Murphy91] to form a benchmark for comparing and evaluating learning algorithms. Table 2 shows the thirteen data sets and their short descriptions. Real-world data sets reflect the situations of real-world applications. Breast cancer and Diabetes data sets are examples of real-world data sets. Synthetic data sets reflect situations of real-world applications through simulation. LED-24 and Waveform-40 data sets are examples of synthetic data sets. Both LED-24 and Waveform-40 data sets are generated by programs written in C.

Table 2: Data sets in the benchmark [Zheng93]

Name	Description
Breast Cancer (W)	Medical diagnosis applied to breast cytology (Wisconsin)
Diabetes	Pima Indians diabetes databases for diagnosing diabetes
Hepatitis	Predicting whether a patient will die from hepatitis
LED-24	LED display with 24 segments (17 irrelevant)
LED-7	LED display with 7 segments
Lymphography	Lymphography database
Monks-2	The second Monk's problem
Mushroom	Mushrooms classified as poisonous or edible
NetTalk (Phoneme)	NetTalk Corpus for the phonetic transcription of the 1000 most common English words (prediction of phoneme)
Promoter	Promoter gene sequences (DNA)
Soybean	Large soybean database
Thyroid	Hypothyroid database (thyroid disease records)
Waveform-40	Waveform database with 40 attributes (19 irrelevant)

In general, the data set that describes a classification problem has three aspects: the form of the attributes, the form of the instances, and the forms of the classes [Zheng93]. Appendix A shows the dimensions of each data set and their values. Zheng uses sixteen dimensions to describe these data sets (see Appendix A):

- Four dimensions concern attributes: the type of attributes, the number of attributes, the number of different nominal attribute values, and the number of irrelevant attributes.

- Five dimensions regard instances: the data set size, the data set density, the level of noise in attribute values, the level of noise in class membership or indeterminacy, and the frequency of missing attribute values.
- Seven dimensions relate to classes: the number of classes, the default accuracy, the entropy, the predictive accuracy, the relative accuracy, the average information score, and the relative information score.

These dimensions are described below.

3.1.1 Type of Attributes

There are four types of attributes: *binary, nominal, continuous*, and *mixed*. Binary attribute means its attribute value is either 1 or 0, YES or NO, + or -, and so on. Nominal value is also known as discrete value. The attribute value is within a specify range, such as '1, 2, 3, or 4 ', or 'yellow, red, or blue'. Continuous attribute values are 1, 45.1, 999.12, and so on; they do not have any range or boundary. A mixed attribute is a combination of at least two type of these attributes: binary, nominal, and continuous.

3.1.2 Number of Attributes

The number of attributes for a data set can be classified as *small* (less than 10), *medium* (between 10 and 30), or *large* (more than 30). For example, LED-7 has 7 attributes (small), LED-24 has 24 attributes (medium), and Promoter has 57 attributes (large).

3.1.3 Number of Different Nominal Attributes Values (#DNAV)

The #DNAV is the total number of different nominal attribute values for a data set. The #DNAV also can be classified as *small* (less than 5), *medium* (between 5 and 10), or *large* (more than 10). For example, Promoter has 4, Soybean has 7, and Mushroom has 12.

3.1.4 Number of Irrelevant Attributes (#IAtt)

Irrelevant attributes occur both in real-world and synthetic data sets. However, it is quite difficult to identify irrelevant attributes in a real-world data set. The synthetic data sets such as LED-24 and Waveform-40 have irrelevant attributes. The #IAtt can affect the performance of learning algorithms.

3.1.5 Data Set Size

Data set size is classified as *small* (less than 210 instances), *medium* (between 210 and 3170 instances), and *large* (more than 3170 instances). In most cases, the size of a data set directly affects the training time and classification accuracy of a learning algorithm.

3.1.6 Data Set Density

Learning algorithms usually achieve a higher accuracy from a large number of training examples (larger data set size). However, because different domains have different description space sizes, it is very hard to conclude those data sets that contain more than *N* examples are large and those that contain less than *N* examples are small for some *N*, according to Zheng [Zheng93]. Thus, the density of description space is a more useful way to characterize a data set. Density is defined as:

$$Density = \frac{Number\ of\ examples}{Size\ of\ description\ space}$$

$$Size\ of\ description\ space = \prod_{i=1}^{n} N_i$$

where n = the number of attributes,

N_i = for the *i*-th attributes, the number of different values (if it is a binary or nominal attribute), or the number of different values in the data set (if it is a continuous attribute).

There are three different classifications for the data set density: *low* (less than $1.0*10^{-18}$), *medium* (between $1.00*10^{-18}$ and $6.00*10^{-7}$), and *high* (greater than $6.00*10^{-7}$). For examples, data set density for Promoter is low, Soybean is medium, and LED-7 is high.

3.1.7 Level of Noise in Attribute Values and Class Membership

Noise affects the performance of learning algorithms. Noise in attribute values and classes is inevitable especially in the real-world domains such as medical domains. Noise occurs due to errors introduced when measuring and diagnosing, and indeterminacy with respect to attributes.

Both LED-24 and Waveform-40 data sets have noise in their attribute values. Monks-2 does not has noise in its attribute values. Breast Cancer data set contains noise in its class memberships, but LED-24 does not.

3.1.8 Frequency of Missing Attribute Values (FMAV)

The Frequency of Missing Attribute Values is the ratio of the number of missing attribute values with respect to the total attribute values in a data set. The frequency is divided into three classes, *none*, *few* (between 0 and 5.6%), and *many* (more than 5.6%).

3.1.9 Number of Classes

The number of classes is categorized into *binary* (2), *small* or *medium* (between 3 and 10), and *large* (more than 10). Hepatitis and Promoter data sets have binary classes, Lymphography and Waveform-40 data sets have medium number of classes, and Soybean and NetTalk data sets have large number of classes.

3.1.10 Default Accuracy

The default accuracy is the relative frequency of the most common class in a data set. It is classified as: *low* (less than 40%), *medium* (between 40% and 75%), and *high* (more than 75%). Soybean data set has the lowest default accuracy (13.7%), Thyroid data set has the highest default accuracy with 95.2%, and most of the other data sets have medium default accuracy.

3.1.11 Entropy

In general, the default accuracy is used to describe the major class of a data set. In addition to the default accuracy, this benchmark includes entropy to account for the class distribution of a data set. Entropy is the expected amount of information for classifying an instance. It can be defined as:

$$-\sum_{i=1}^{N} P(C_i) * log_2 P(C_i) \text{ bits}$$

where N = the number of classes and

$P(C_i)$ = the prior probability of class C_i.

"In many cases, the probabilities of the occurrence of certain results are known a priori," said Fast [Fast68]. The prior probability of class x is the probabilities of the occurrence of class x in a data set. The unit of information (by entropy) using base 2 logarithms is called a "bit". This word is first suggested by John W. Tukey, being a condensation of "binary digit" [Shannon64].

Entropy is also described by three terms, *low* (less than 0.80 bits), *medium* (between 0.80 bits and 1.58 bits), and *high* (more than 1.58 bits). The Thyroid data set has the lowest entropy with 0.28 and NetTalk has the highest entropy with 4.72.

3.1.12 Predictive Accuracy and Relative Accuracy

Predictive accuracy and relative accuracy are used to discuss the difficulty of a domain. Predictive accuracy is defined as the highest accuracy achieved by some existing algorithms, recorded in UCI-RMLD (past usage). If such an accuracy of a domain is not available, the accuracy achieved by the C4.5 decision trees [Quinlan86] learning algorithm is used. Relative accuracy is defined as:

$$Relative\ accuracy = \frac{Predictive\ accuracy\ -\ Default\ accuracy}{100\% -\ Default\ accuracy} * 100\%$$

where 100% in the denominator is the accuracy that would be achieved by a perfect learning algorithm.

Predictive accuracy is divided into three classes: *low* (less than 80%), ***medium*** (between 80% and 98.5%), and *high* (more than 98.5%). Relative accuracy is also divided into three classes: *low* (less than 52%), *medium* (between 52% and 88.5%), and *high* (more than 88.5%).

3.1.13 Average Information Score and Relative Information Score

Predictive accuracy is the most commonly used evaluation criteria, but it does not consider the prior probabilities of a class and class distribution of a data set. Zheng uses the average information score and relative information score introduced by Kononenko and Bratko [Kononenko91] to overcome the shortcoming of predictive accuracy.

Average information score and relative information score are defined as follows,

$$I_{average} = \frac{1}{T} * \sum_{j=1}^{T} I_{ej}$$

$$I_{relative} = \frac{I_{average}}{Entropy_{test}} * 100\%$$

where $I_{average}$ = the average information score,

$I_{relative}$ = the relative information score,

$Entropy_{test}$ = the entropy of the test set,

T = the number of test examples , and

I_{ej} = the information score of the classifier on the test example *ej*, and defined as:

$$I_{ej} = \begin{cases} -\log_2 P(C_{ej}) + \log_2 Q(C_{ej}) & \text{if} Q(C_{ej}) \geq P(C_{ej}) \\ -(-\log_2 (1 - P(C_{ej})) + \log_2 (1 - Q(C_{ej}))) & \text{if} Q(C_{ej}) < P(C_{ej}) \end{cases}$$

where C_{ej} = the correct class of text example *ej*,

$P(C_{ej})$ = the prior probability of class C_{ej}, and

$Q(C_{ej})$ = the posterior probability returned by the classifier.

Average information score is classified into three levels: *low* (less than 0.25 bits), *medium* (between 0.25 bits and 1.30 bits), and *high* (more than 1.30 bits). Relative information score has three levels: *low* (less than 4.50%), *medium* (between 45.0% and 85.5%), and *high* (more than 85.5%).

3.2 Data Sets

This experiment uses all the data sets suggested by Zheng's benchmark except the NetTalk data set. The NetTalk data set requires a huge amount of training time especially for FFNBP and FuNe-I. Due to the time constraints of the research, the NetTalk data set was dismissed. The Waveform-40 data set was provided by Zijian Zheng [Zheng93]. The other eleven data sets are obtained from UCI-RMLD: Breast Cancer (W), Diabetes, Hepatitis, LED-24, LED-7, Lymphography, Monks-2, Mushroom, Promoter, Soybean, and Thyroid.

The following sections introduce each of the data sets, in general. The information is mostly summarized from the individual "*readme.txt*" that is included with the data sets at UCI-RMLD. Table of Appendix A provides more information about the characteristics of each data set that are defined using the sixteen dimensions.

3.2.1 Breast Cancer (Wisconsin) 2

The *Breast Cancer data set (Wisconsin)* originates from the University of Wisconsin Hospitals, Madison. The data was collected by Dr. William H. Wolberg. The data set is dated January 8, 1992.

The Breast Cancer data set (BCDS) has 699 instances. The data set has nine attributes and a class attribute. All of its attributes are continuous. The number of classes is binary or 2, and its class distributions are 65.5% (458 instances) for *Benign* and 34.5% (241 instances) for *Malignant*.

The past usage of the data set has applied mostly to the older version of the BCDS, which has only 369 instances. Zhang applied four instance-based learning algorithms on the data set, as cited by Murphy [Zhang92] [Murphy91]. The algorithms were trained on 200 instances and tested on the other 169 instances using 10-fold cross-validation. The best classification accuracy was obtained by 1-nearest neighbor algorithm with 93.7%. Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on the data set, which has 699 instances. The reported accuracies are 96.0%, 94.8%, and 94.7% respectively.

3.2.2 Diabetes

The *Pima Indians Diabetes data set* (DDS) is the formal name for this data set. The owner of the DDS is the National Institute of Diabetes, Digestive, and Kidney Diseases.

The Diabetes data set has 768 instances. It has eight continuous attributes plus the class attributes. It does not have any missing attributes. It has two classes (binary classes). Its class distributions are 65.1% (500 instances) for class 0 and 34.9% (268 instances) for class 1.

Smith et al. used the ADAP learning on the older version of the DDS to forecast the onset of diabetes mellitus [Smith88]. The older version of DDS has only 576 instances. The algorithm was trained with 384 instances and tested with 192 instances. The classification accuracy is 76%. Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on the new version data set which has 768 instances. Their accuracies were 70.6%, 71.5%, and 70.4% respectively.

3.2.3 Hepatitis

The *Hepatitis data set* was donated to UCI-RMLD by G. Gong from Carnegie-Mellon University.

Hepatitis data set (HDS) has 155 instances. HDS has twenty attributes that consist of thirteen binary attributes, six continuous attributes, and a class attribute. The data set has two classes (binary classes) with 32 instances (21.6%) of class *DIE* and 123 instance (79.4%) of class *LIVE*.

Cestnik et al. reported a 83% classification accuracy using Assitand-86, as cited by Murphy [Cestnik87] [Murphy91]. Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on the data set. The reported accuracies are 81.9%, 78.2%, and 82.1% respectively.

3.2.4 LED-24 and LED-7 10

The *LED Display Domain* (LDD) is divided into two parts, LED-24 and LED-7. The LDD is created by Breiman and his associates. The data sets are generated by programs that written in C.

In this experiment, each data set (LED-24 and LED-7) has 200 instances. Each domain has ten classes. LED-24 has seven binary attributes, seventeen irrelevant attributes, and noise in its attribute values. The LED-7 consists of seven binary attributes and no noise. Neither of the data sets have missing values in their attributes. Their class distributions are 10% theoretically since each class has the same theoretical probability of occurrence.

Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on LED-24 and LED-7. The reported accuracies on LED24 are 32.0%, 60.5% and 60.5% respectively. The reported accuracies on LED-7 are 71.0%, 69.5% and 70.5% respectively.

3.2.5 Lymphography 4

Lymphograhy data set was collected by M. Zwitter and M. Soklic, University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia.

Lymphography data set (LDS) consists of nine binary attributes, nine nominal attributes, and a class attribute. It has eight different nominal attribute values. Irrelevant attributes, missing attribute values, and noise are not present in LDS. LDS has 148 instances. It has four classes and its class distributions are *normal find* (2 instances), *metastases* (81 instances), *malign lymph* (61 instances), and *fibrosis* (4 instances).

LDS has been tested by various algorithms. Cestnisk et al. used Assistant-86 and reported a 76% classification accuracy [Cestnik87]. Clark et al. used Simple Bayes (83%) and CN2 (82%) on LDS [Clark87]. Michalski et al. tested AQ15 with LDS and reported a 80-82% accuracy [Michalski86b]. Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on the data set. Their accuracies were 82.4%, 78.4%, and 81.1% respectively.

3.2.6 Monks-2 2

The *Monks-2 Problems data set* (M2DS) originates from Sebastian Thrun, Carnegie-Mellon University, Pittsburgh, Philadelphia, USA.

M2DS contains a total of six attributes. The attributes consist of two binary attributes and four nominal attributes. The nominal attribute consists of four different nominal attribute values. Irrelevant attributes, noise, and missing attribute values are not present in M2DS. M2DS has 432 instances and two classes.

Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 3-fold cross-validation on the data set. The reported accuracies are 70.4%, 60.5%, and 72.7% respectively.

3.2.7 Mushroom 2

The *Mushroom data set* (MDS) is drawn from *The Audubon Society Field Guide to North American Mushrooms*. It was prepared by Jeff Schlimmer.

MDS consists of four binary attributes and eighteen nominal attributes, for a total of 22 attributes. The number of different nominal attribute values is 12. Irrelevant attribute and noise are not present in MDS. However, MDS has about 2480 missing values, all for attribute #11. There are 8124 instances and two classes in MDS. Its class distributions are 51.8% (4208 instances) for *edible* and 48.2% (3916 instances) for *poisonous*.

Murphy cited Schlimmer's STAGGER reported a 95% accuracy after reviewing 1000 instances [Murphy91]. Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on the data set. All three algorithms achieved 100% classification accuracy.

3.2.8 Promoter 2

The official name for *Promoter data set* is *E. coli promoter gene sequence (DNA)* with associated imperfect domain theory. The creators of the promoter instances are C. Harley and R. Reynolds. The creator of the non-promoter instances and the domain theory is M. Noordewier.

Promoter data set (PDS) contains 57 nominal attributes and a class attribute. The nominal attributes have four values. Irrelevant attribute, noise, and missing value are not present in PDS. PDS has 106 instances and two classes. The instances are distributed equally between two classes.

Towell et al. [Towell90] has used PDS with KBANN (Knowledge-Based Artificial Neural Net), Standard Backpropagation Neural Nets with one hidden layer, ID3, and Nearest Neighbor with k is 3. Using leave-one-out methodology, the accuracies achieved by the algorithms are 96.23%, 92.45%, 82.08%, and 87.74% respectively. Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on the data set. The reported accuracies are 83.0%, 76.0% and 81.1% respectively.

3.2.9 Soybean 19

Soybean data set (large) was collected R.S. Michalski and R.L. Chilausky.

Soybean data set (SDS) contains seventeen binary attributes and nineteen nominal attributes. The number of nominal attribute values is seven. SDS has 2337 missing values in its data set. It also has noise in both of its attribute values and class membership. SDS has nineteen classes.

Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on the data set. The reported accuracies are 91.1%, 91.5%, and 99.1%.

3.2.10 Thyroid θairɔid 甲状腺 2

The formal name of *Thyroid data set* is *Hypothyroid.* This data set was left at University of California at Irvine by Dr. Ross Quinlan during his visit in 1987 for the 1987 Machine Learning Workshop.

The Thyroid data set consists of eighteen binary attributes, seven continuous attributes, and a class attribute, a total of 25 attributes. It has noise at its attribute level, which 5329 of its attribute values are missing. The total number of instances in this data set is 3163. It has two classes.

Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on the data set. The reported accuracies are 97.1%, 99.1%, and 99.1% respectively.

3.2.11 Waveform-40 3

Waveform-40 is an artificial data set. The data set is generated by Waveform Database Generator, which is written in C. The authors of the program are Breiman and his colleagues.

Waveform-40 has 40 continuous attributes. Missing values are not present in the data set. However, it has noise in its attributes. It also has nineteen irrelevant attributes. 300 instances were used for this experiment. Waveform-40 has three classes. The instances are distributed equally among the three classes.

Waveform-40 was used by Optimal Bayes classification (86%), CART decision tree algorithm (72%), and Nearest Neighbor Algorithm (38%) [Murphy91]. Zheng [Zheng93] used IB1, C4.5, and CI2-2L algorithms with 10-fold cross-validation on the data set. The reported accuracies are 67.7%, 69.4%, and 72.7% respectively.

3.3 Cross-Validation

Cross-validation refers to the practice of partitioning the data set into folds of as equal-size as possible. A partition is called *fold*. The smallest number of partitions is 2-fold. The largest number of partitions is n-fold where *n* is the size of the data set. This special case is also known as *jackknife* or *leave-one-out* cross-validation. For n-fold, *n* runs of the learning algorithm are made. During each run, all but one of the folds (n-1) are used for training, and the remaining fold is used for testing.

4-fold cross-validation refers to the case where the data set is randomly divided into 4 folds. For example (see Figure 8), the LED7 data set has 200 instances. The data set is divided into 4 folds (1, 2, 3, and 4). Each fold has 50 instances. Four runs are made. During each run, three of the folds are selected as the training set and the fourth is used as testing set. The process is repeated four times, each time with a different fold used as the testing set. The final result is the average of the four separate runs.

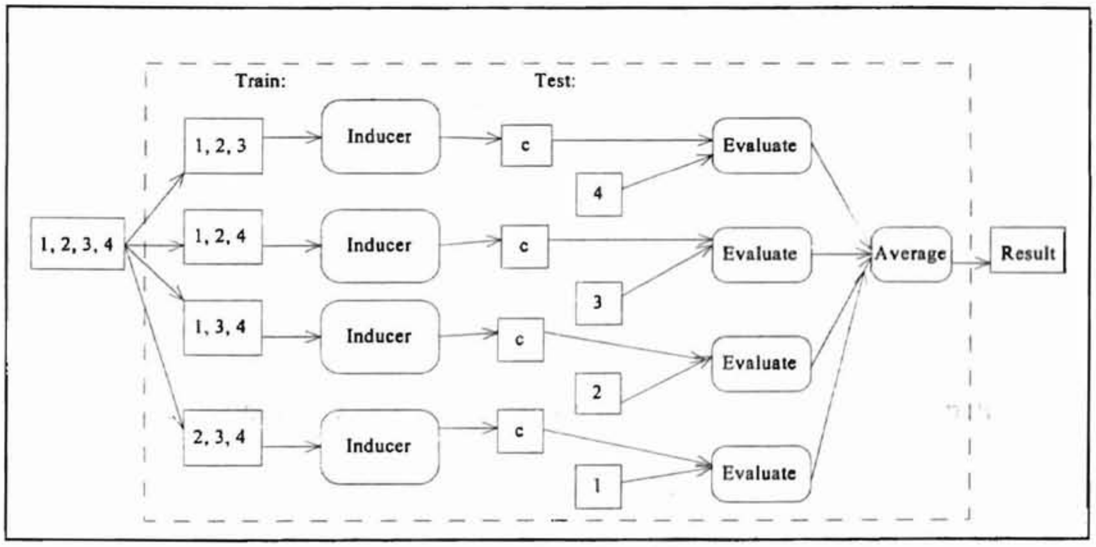


Figure 8: An example of 4-fold cross-validation.

Many experiments have used various partitioning such as 2-folds, 3-folds, 10-fold, 3/5 of data set used for training and 2/5 of data used for testing, etc. In this experiment, both 10-fold and 4-fold cross-validations were applied on the data sets.

10-fold cross-validation was chosen because it had proven itself better than other methods. Experimental results [Kohavi95] indicate that for real-world data sets, the best method to use is 10-fold even if computational power allows more folds. 4-fold cross-validation was chosen because the sizes of some data sets are smaller than 500 instances, such as Hepatitis (155), LED7 (200), LED24 (200), Lymphography (148), Promoter (106), and Waveform-40 (300). The testing data set might not represent accurately the original data sets using 10-fold cross-validation.

For all subsequent testing of *dropping features* and *reducing the amount of training data* (as described in sections 4.3 and 4.4), 4-fold cross-validation was used. This decision is based on two factors: 4-fold cross-validation takes less time in training, and the results obtained by this method have shown to be comparable with 10-fold cross-validation in initial tests. Refer to Chapter 4 for further discussion of the results.

3.4 Analysis and Comparison

The benchmark for classifier learning defined by Zheng [Zheng93] is used in evaluating the algorithms. Zheng uses sixteen dimensions to form the benchmark. The benchmark provides a standardized basis for evaluating and comparing the learning algorithms: ID3, FFNBP, and FNN.

The algorithms are analyzed and compared based on the following criteria:

1. classification accuracy,
2. learning time,
3. dependence on the amount of training data, and

4. ability to handle imperfect data of various types, such as missing attributes and irrelevant attributes.

The first and second criteria are standard criteria in comparing learning algorithms. The third and fourth criteria also are important because data sets are different as described by Zheng's sixteen dimensions. Some learning systems slow down rapidly as the numbers of attributes increase, but others do not. Some algorithms can deal with discrete values, continuous values, noise, more than two classes, missing attributes or irrelevant attributes, but some others have difficulty with them [Zheng93]. With the help of the benchmark, it is possible to characterize the learning algorithms and pinpoint particular strengths or weaknesses of different learning algorithms with respect to the four criteria.

3.5 Experiment Platform

The platforms for testing the learning algorithms are Sparcstations 20 running the SunOS Released 5.4 Version Generic_101945-36 (Solaris 2.4) operating system. The SunOS is a UNIX system port that is compatible with UNIX(R) System V Release 4.0. There are fifteen Sparcstations. Each machine is equipped with 64 MB of memory.

CHAPTER 4

RESULTS AND DISCUSSIONS

This chapter reports the results of four experiments. The first and second experiments compared the classification accuracy and learning time of the Induction of Decision Trees (ID3), Feedforward Neural Network with Backpropagation (FFNBP), and Fuzzy Neural Network (FuNe-I) learning algorithms. The effects of the number of training examples and imperfect data on the classification performances of the algorithms were studied in the third and fourth experiment.

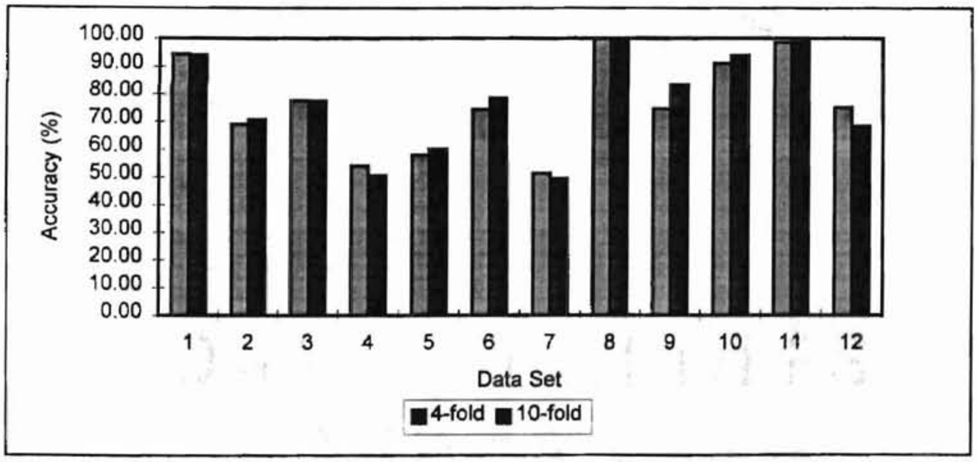
As explained in Chapter 2, FFNBP and FuNe-I have several parameters including learning rate and momentum. A few experimental runs were conducted to find the best parameters for FFNBP and FuNe-I. The FFNBP parameters were set as follows:

- Learning rate = *1/n*, where ***n*** is the number of training instances (as suggested by Dr. Tveter),
- Momentum = 0.9,
- One hidden layer with *m* neurons, where *m* is the number of inputs (attributes) and,
- Each training run stops when the network correctly classifies at least 99.0% of the training data set or when the *number of passes (epochs)* through the data set reaches 5000.

The parameters for FuNe-I were set as follows:

- Learning rate = *1/n* (same as FFNBP),
- Momentum = 0.9, and
- Each *training* and *optimization* run (if applicable) makes 500 iterations (epochs). Optimization of the rules is applicable only if the number of inputs (attributes) is less than or equal to 20.

The learning rates chosen in these experiments are very small (low). FFNBP and FuNe-I take a longer time to be trained with a small learning rate. However, a small learning rate helps avoid oscillation and assures that the algorithms achieve better classification performances.

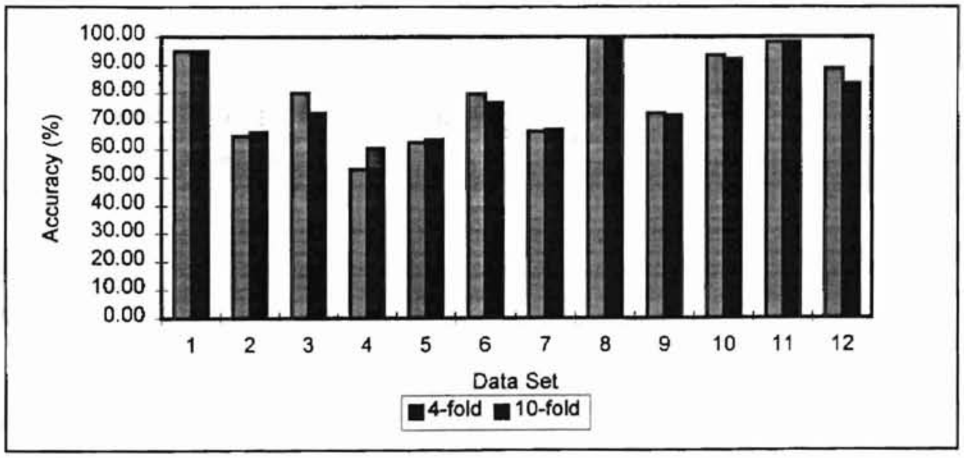


Graph 1: Classification performance of ID3 as a function of data set (4-fold vs. 10-fold).

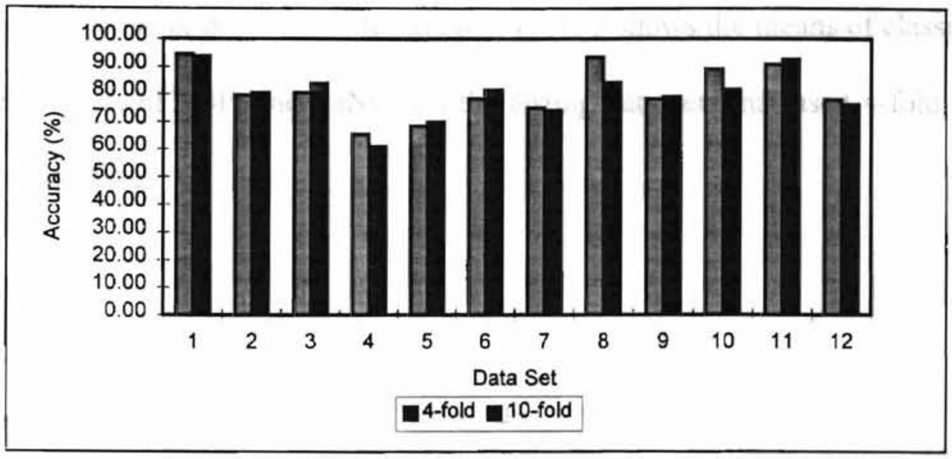
All three algorithms were tested with twelve data sets. The data sets were obtained from University of California-Irvine Machine Learning Database Repository [Murph91]. To perform the experiments, 4-fold cross-validation method was applied to partition each data sets into training and testing data sets. Besides 4-fold cross-validation, 10-fold cross-validation was also used to partition the data sets in the first and second experiments. Chapter 3.3 discusses cross-validation method in more details.

The results show that the algorithms perform as well with 4-fold cross-validation as they do with 10-fold cross-validation. 4-fold cross-validation is even better than 10-fold cross-validation in some cases (see Graph 1, 2, and 3). The results also show that 4-fold cross-validation took less time than 10-fold cross validation for learning from training data sets (see Table B2, B3, B4, B6, B7, and B8 of Appendix B).

ID3 performed better under 10-fold cross-validation. On the other hand, FFNBP and FuNe-I performed better under 4-fold cross-validation (see Table D1 and D5 of Appendix D). Overall, most results obtained by algorithms that used 4-fold and 10-fold cross-validation are within 5% of each other. The following discussions focus on algorithms that used 4-fold cross-validation method.



Graph 2: Classification performance of FFNBP as a function of data set (4-fold vs. 10-fold).



Graph 3: Classification performance of FuNe-I as a function of data set (4-fold vs. 10-fold).

4.1 Experiment One: Classification Accuracy

Classification accuracy on a testing data set is one of the most commonly used evaluation criteria in comparing and evaluating learning algorithms. The accuracy of a classifier is the probability of correctly classifying a randomly selected instance from the data set [Kohavi95]. In this experiment, the classification accuracy on a testing data set is calculated as follow:

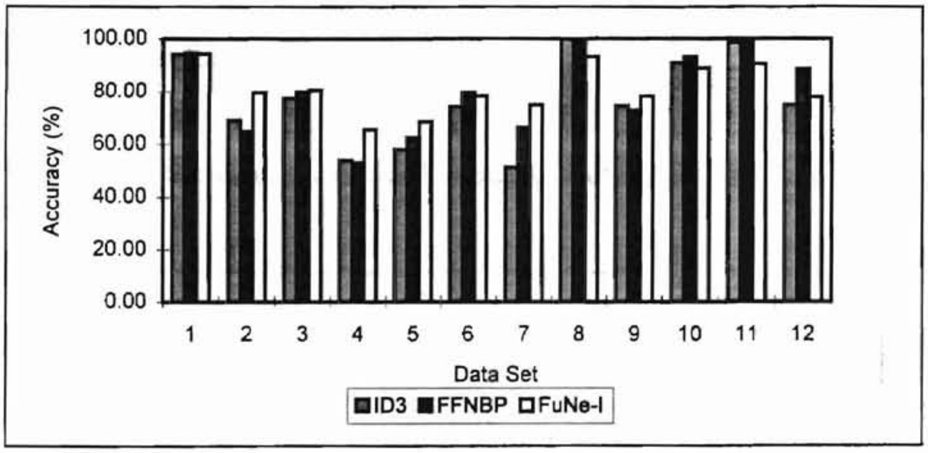
$$\frac{\text{Number of examples correctly classify}}{\text{Total number of examples}} * 100\%.$$

4.1.1 Results

Graph 4 and 5 report the classification accuracies of the three learning algorithms that used 4-fold and 10-fold cross-validation, respectively. The actual numbers and their averages appeared in the Table D1 through D8 of Appendix D. The italic and bold numbers shown in Table D1 and D5 of Appendix B reflected the highest classification accuracy achieved among the three algorithms. Table 3 shows the means of classification accuracies of ID3, FFNBP, and FuNe-I on the testing data sets that used 4-fold and 10-fold cross-validation.

4.1.2 Discussion

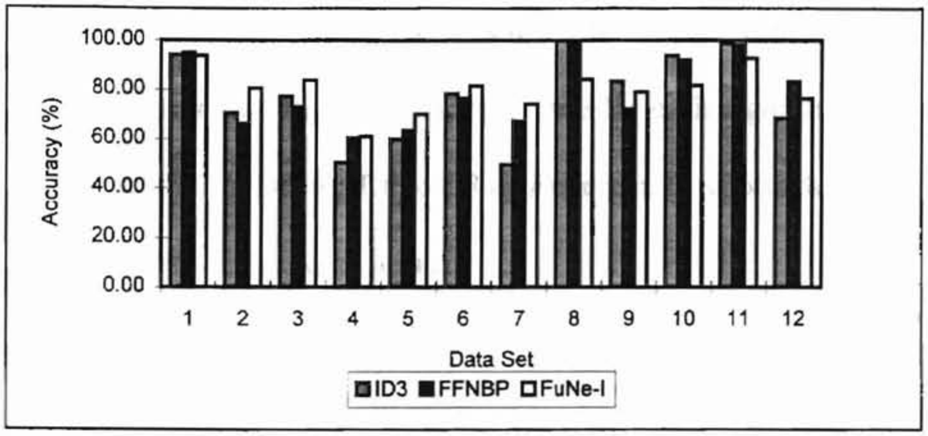
In this experiment, the results show ID3 and FFNBP learning algorithms are similar with respect to accurately classifying examples. A majority of the results are within 5% of each other except the data sets Lymphography, Monks-2, and Waveform-40. This conclusion coincides with the conclusions from other experiments [Mooney91] [Weiss89] [Atlas90] [Towell90] [Chen94].



Graph 4: Classification performance of ID3, FFNBP, and FuNe-I as a function of data sets using 4-fold cross-validation.

FuNe-I consistently outperformed or performed as well as (within 5%) ID3 and FFNBP on most of the data sets except Mushroom, Thyroid, and Waveform-40. According to the classification accuracies of the experiment (see Table D1 of Appendix

D), FuNe-I is the best, followed by FFNBP and ID3. FuNe-I outperformed ID3 and FFNBP on six of the twelve data sets. FuNe-I performed better on *small* and ***medium*** data sets. It did poorly on *large* data sets. Chapter 3 described the characteristic of each data set and the sixteen dimensions used to define each data set.



Graph 5: Classification performance of ID3, FFNBP, and FuNe-I as a function of data sets using 10-fold cross-validation.

Table 3: Averages of the accuracy of the 12 data sets for ID3, FFNBP, and FuNe-I algorithms.

	4-fold	10-fold
ID3	76.48 %	77.04 %
FFNBP	79.55 %	78.96 %
FuNe-I	80.98 %	79.88 %

4.2 Experiment Two: Learning Time

Learning time is another commonly used evaluation criteria in comparing learning algorithms. Learning time is the time learning algorithms take to learn the theory or concept from examples or training data sets. In this experiment, the learning time included the time (testing time) used to classify testing data sets after learning. The testing time is very small, approximately 2% of the overall time. The UNIX korn shell command *time* was used to keep track of the time.

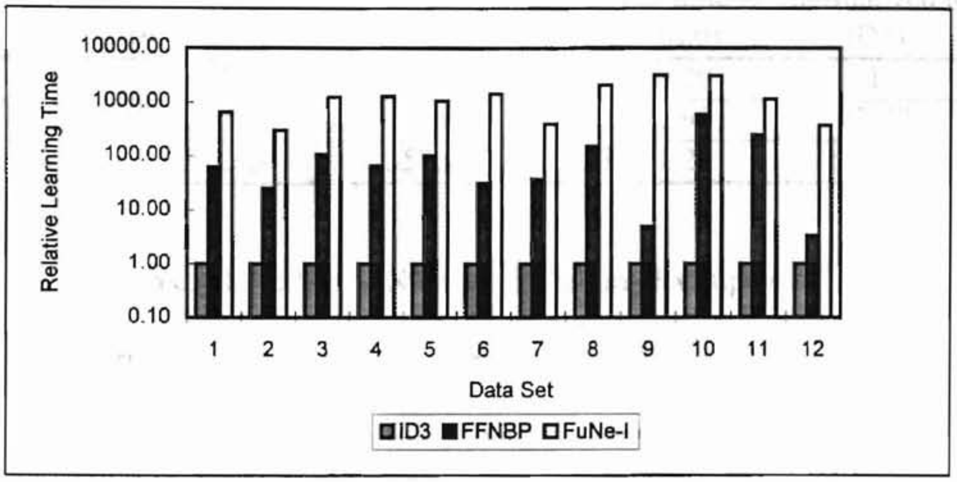
4.2.1 Results

Graph 6 and 7 show the relative learning time of the three learning algorithms (normalized to the time taken by ID3) using 4-fold and 10-fold cross-validation, respectively. Table 4 shows the mean learning time for each algorithm and its relative learning time. Table D2 to D4 and D6 to D8 of Appendix D reports the actual time each algorithm took to learn and test each data set.

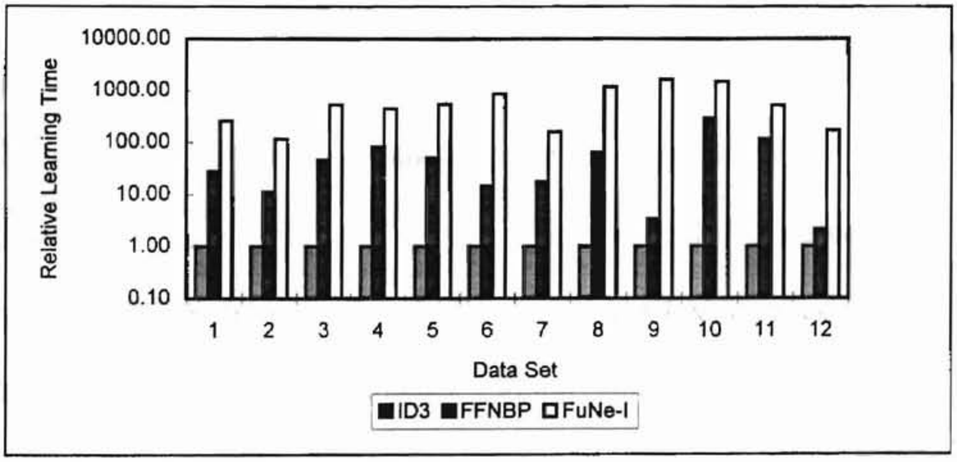
4.2.2 Discussion

ID3 took the least time in learning from the training examples, followed by FFNBP and FuNe-I. FuNe-I generated the worst learning time. As explained in Chapter 2, FuNe-I works in two phases. It first generates rules from training data set through its training network. Then, it tunes the parameters and the extracted rules in the second phase using the fuzzy system. Thus, it was expected to take longer time than the other two algorithms in learning from the training data set. However, the learning times captured from the experiment were unexpectedly high. FuNe-I takes about 1357 times longer than ID3 to learn from training examples (see Table 4).

The relative quickness of ID3 with respect to FFNBP in learning concepts from training examples was also observed in other experiments. Experimental results from Chen et al. [Chen94], Fisher and McKusick [Fisher89], Mooney et al. [Mooney91], Towell et al. [Towell90], Tsaptsinos et al. [Tsaptsinos90], and Weiss and Kapouleas [Weiss89] agree with the above conclusion.



Graph 6: Relative learning time of ID3, FFNBP, and FuNe-I as a function of data sets using 4-fold cross-validation.



Graph 7: Relative learning time of ID3, FFNBP, and FuNe-I as a function of data sets using 10-fold cross-validation.

Table 4: Averages of the learning time and the relative learning time for ID3, FFNBP and FuNe-I algorithms.

	4-fold: average learning time (h:mm:ss.0)	Relative Learning Time (normalized to ID3)	10-fold: average learning time (h:mm:ss.0)	Relative Learning Time (normalized to ID3)
ID3	0:00:08.18	1	0:00:20.20	1
FFNBP	0:21:45.46	159.61	0:24:39.92	73.27
FuNe-I	3:05:02.84	1357.45	3:31:11.26	627.32

4.3 Experiment Three: The Effect of Number of Training Examples

Some learning algorithms perform relatively better with small amount of training examples while others perform relatively better on large training examples. In this experiment, the three algorithms were tested for their dependency on the number of the training examples.

4.3.1 Results

Graph B1 through B12 of Appendix B present classification accuracy of ID3, FFNBP, and FuNe-I as a function of the percentage of the data set used for each of the twelve data sets. Graph B13a through B15b present classification accuracy as a function of percentage of data set for each learning algorithm.

Each data point on the graphs represent the average of three distinct runs with data sets *a*, *b*, and *c*. The composition of data sets *a*, *b*, and *c* are shown in Table 5. The original data set was randomly divided into four parts (1, 2, 3, and 4). Each part represents ¼ of the total number instances of a data set. For example, when a data set is reduced by ¼ of its original size; data set *a* consists of part 1, 2 and 3 of the original data set; data set *b* consists of part 2, 3, and 4 of the original data set; and data set *c* consists of part 1, 2, and 4 of the original data set.

Table 5: Organization of data set *a*, *b*, and *c*.

	Reduced by 1/4	Reduced by 1/2	Reduced by 3/4
a	1, 2, and 3	1 and 2	1
b	2, 3, and 4	2 and 3	2
c	1, 2, and 4	3 and 4	3

4.3.2 Discussion

Compared to FFNBP and ID3, FuNe-I performs as well and better on *small* and *medium* (based on Zheng's benchmark) data sets. The dependency of the algorithm on the amount of the training instances is the least among the three algorithms. The conclusion is supported by the results of the experiment.

From observing the graphs of Appendix B and tables of Appendix D, the classification accuracies for FuNe-I on some of the data sets increased as the numbers of instances were reduced, especially on *small* and *medium* data sets (Hepatitis, LED-24, LED-7, Lymphography, and Promoter). The largest increase was recorded on Promoter data set, ≅9%. The classification accuracy on Thyroid data set increased by almost 4% even though its size is *large*. The classification performance on the rest of the data sets decreased slightly (≤ 2%) except Soybean data set, which dropped about 13% and Mushroom data set lost about 6%.

On the same token, the classification accuracies of ID3 on the data sets decreased consistently as the numbers of instances were reduced. For ID3, the classification accuracies degraded slightly (≤ 3%) for most data sets (Breast Cancer, Diabetes, Lymphography, Monks-2, Mushroom, and Thyroid). The classification accuracy of ID3 on Waveform-40 data set suffered the most lost (≅12%), it is followed by Soybean (≅9%), LED-24 (≅9%), Promoter (≅9%), Hepatitis (≅5%), and LED-7 (≅4%) data sets.

For FFNBP algorithm, its classification accuracies on the data sets degraded slightly for some and plenty for the other as the numbers of instances were reduced. The classification accuracies of six data sets decreased drastically (≥ 5%); the greatest change was on data set LED-24 (≅18%) followed by Soybean (≅10%), Monks-2 (≅9%), Waveform-40 (≅8%), LED-7 (≅8%), and Hepatitis (≅5%). The classification performances on other six data sets decreased only slightly (≤ 3%).

4.4 Experiment Four: The Effect of Imperfect Data

The sensitivity of learning algorithms to imperfect data is also an important aspect in comparing the algorithms. Imperfect data can affect the performance of learning algorithms. Data is improperly represented due to various reasons: for example, mistakes may be made when recording and copying attribute values, or some attribute values may be missing; or instances that are represented or described with insufficient collection of attributes.

This experiment investigated two types of imperfect data and reported their effect on the three learning algorithms. Irrelevant attributes and completely-dropped attributes were used to stimulate imperfect data, and the performances of the three algorithms were compared.

4.4.1 Irrelevant Attributes

The first type of imperfection is *irrelevant attributes*. Irrelevant attributes do exist in the data sets used in these experiments. However, it is very difficult to determine which attributes are irrelevant. Irrelevant attributes can affect the learning performance of a classifier learning. LED-24 and Waveform-40 are the only known data sets with irrelevant attributes, 7 irrelevant attributes for LED-24 data set and 19 irrelevant attributes for Waveform-40 data set. The only known data set that does not has irrelevant attributes is LED-7.

4.4.1.1 Results

Graph 4 and Table B1 of Appendix B show the results obtained by the three algorithms on LED-24, LED-7, and Waveform-40 data sets.

4.4.1.2 Discussion

The results obtained by the three algorithms on LED-24 and LED-7 show that FuNe-I and ID3 are the least affected by irrelevant attributes. ID3 reported 58% and 54% classification accuracies on LED-7 and LED-24, respectively. The difference is very small. FuNe-I shows the same characteristic, and its classification accuracies are better; 68.5% for LED-7 and 65.5% for LED-24. The difference is very small, too. On the other hand, FFNBP is affected the most by the irrelevant attributes. Its classification accuracy different between LED-7 and LED24 is 9.5%.

4.4.2 Completely-Dropped Attributes

The second imperfection investigated is *completely-dropped attributes*. The imperfection arises when an insufficient information is used to describe or represent examples. This experiment investigated the sensitivity of the learning algorithms to the attributes by randomly dropping some of the attributes. If an attribute is dropped, it is dropped from all examples in both the training and testing data set. In this experiment, all of the odd numbered attributes were dropped gradually from 10%, 25% to 50% of the total attributes in the data set. Table 6 shows the data sets and the number of attributes dropped for each category, 10%, 25%, and 50% of the total number of attributes (in a data set).

Table 6: Number of attributes being dropped (10%, 25% and 50%) for each data set.

Data Set #	Data Set Name	Total # of Attributes	Reduced by 10%	Reduced by 25%	Reduced by 50%
1	Breast Cancer	9	1	2	4
2	Diabetes	8	1	2	4
3	Hepatitis	19	2	5	10
4	LED-24	24	2	6	12
5	LED-7	7	1	2	4
6	Lymphography	18	2	4	9
7	Monks-2	6	1	2	4
8	Mushroom	22	2	6	11
9	Promoter	2	6	14	28
10	Soybean	19	4	9	18
11	Thyroid	2	2	6	12
12	Waveform-40	3	4	10	20

4.4.2.1 Results

Graph C1 through C12 of Appendix C present classification performance of ID3, FFNBP, and FuNe-I as a function of percentage of attributes dropped for each of the twelve data sets. Graph C13a through C15b present classification performance of the learning algorithms as function of the percentage of attributes dropped for all twelve data sets.

4.4.2.2 Discussion

All three algorithms' performances degraded the most on data set LED-24 and LED-7. ID3 and FuNe-I performed comparably as the attributes were dropped from the data sets. Both ID3 and FuNe-I performed better than FFNBP. On interesting aspect of the Graphs of Appendix C is the occasion where the classification performance of the three learning algorithms improve when the attributes were dropped. It illustrates that extra attributes can degrade learning algorithms' performances. The conclusion is supported by these experimental results and it also agrees with the experimental results of Mooney et al. [Mooney91].

ID3 performed very well as the numbers of the attributes were being reduced. Its classification performances on most of the data sets degraded gradually (≤ 10%), except LED-24, LED-7, and Waveform-40. Four of the data sets reported better results as the number of attributes decreasing (Diabetes, Hepatitis, Lymphography, Monks-2, and Mushroom). ID3's classification accuracy on Monks-2 data set increased by 25.16%. Classification accuracy of ID3 on Mushroom and Lymphography data sets unchanged.

The experiment results show that FuNe-I performed as well as ID3 and it worked best with fewer attributes. Table D14 of Appendix D and graphs of Appendix C show that the classification performances of FuNe-I on most of the data sets degraded gradually (≤ 10%) except LED-24 and Soybean data sets. Four of the data sets achieved better results as the numbers of attributes were decreased (Hepatitis, Lymphography, Thyroid, and Waveform-40).

FFNBP is affected the most by the reduction of attributes. Tables of Appendix D and graphs of Appendix C show its classification performances on the data sets degraded more drastically than ID3 and FuNe-I. Its classification accuracies on five of the data sets lost are greater than 10%.

CHAPTER 5

SUMMARY AND FUTURE WORK

5.1 Summary

The main objective of the experiment is to study and understand more about the comparative strengths and weaknesses between Induction of Decision Trees (ID3), Feedforward Neural Network trained with Backpropagation (FFNBP), and Fuzzy Neural Network (FuNe-I). All the three algorithms were tested on twelve data sets obtained from the University of California-Irvine Machine Learning Database Repository [Murphy91]. Sixteen dimensions defined by Zheng were used to describe these data sets [Zheng93]. The algorithms were tested and compared based on the classification accuracy, the learning time, the effects of the number of training data, and the effects of imperfect data.

The first experiment tested the classification accuracy of the three algorithms. The experiment found that FuNe-I is superior than ID3 and FFNBP on *small* and *medium* data sets. FuNe-I performance on *large* data sets was poor. The performance of ID3 and FFNBP is comparable.

Learning time of each algorithm was monitored in the second experiment. The results show that FuNe-I generated the worst learning time. It is followed by FFNBP and ID3. ID3 performed extremely fast. In the third experiment, the three algorithms' dependencies on the amount of training examples were tested. FuNe-I outperformed ID3 and FFNBP as the numbers of training examples were reduced. FuNe-I works better with *small* and *medium* data sets. FuNe-I's classification performance increased as the numbers of training examples were reduced. Overall, ID3's classification performance consistently degraded slightly across the twelve data sets. FFNBP's classification performance was not stable. Its performance degraded drastically on six of the twelve data sets.

In the last experiment, two types of imperfect data were tested: irrelevant attributes and completely dropped of attributes. In the first section of the experiment, the results indicated that FuNe-I and ID3 performed better than FFNBP with irrelevant attributes. FFNBP's classification performance degraded as the irrelevant attributes were introduced to the LED-24 data set. FuNe-I achieved higher classification accuracy on that data set and it was least affected by the irrelevant attributes.

The results from the second section of experiment four show the occasions where the classification performances of ID3, FFNBP, and FuNe-I algorithms improved when the number of attributes was reduced. It illustrates that extra attributes can degrade learning algorithms' performance. Both ID3 and FuNe-I performed better than FFNBP as the number of attributes was being reduced. The results show that FuNe-I works well with fewer attributes. Its classification performances on some data sets were better when fewer numbers of attributes were presented. At the same moment, FFNBP was most affected by the attribute reduction. Its classification performances degraded more drastically as compared to ID3 and FuNe-I.

5.2 Future Work

Due to the time constraint, only selected numbers of criteria were used to compare the three algorithms. Thus, there are several directions where future endeavors might pursue to better evaluate and compare the three algorithms.

The first improvement is to include the Nettalk data set. The data set was omitted from the experiment due to the time constraint. The data set requires huge learning time (especially for FFNBP and FuNe-I). This will truly satisfy the requirements for Zheng's learning classifier benchmark.

A second improvement is to test the algorithms with noise and missing values in the data set. Noise and missing values affect the performance of the learning algorithms. Each noise and missing values should be manually applied to the data set. Then, their effects on the algorithms' performances can be measured and compared.

The third improvement is to include the information scores and relative information scores. The information scores can assist the experiment in evaluating and comparing algorithms on different data sets. This measure is introduced by Kononenko and Bratko [Kononenko91] to overcome the shortcoming of predictive accuracy.

Finally, the experiment can be expanded to include different types of Fuzzy Neural Networks. Currently, there are a lot of research work is going on different types of Fuzzy Neural Networks, such as Fuzzy RCE [Roan93], Multilayer Feedforward Net with Pattern Learning [Qin92], Multilayer Feedforward Net with Batch Learning [Qin92], Feedforward Net with External Recurrence and Pattern Learning [Qin92], Feedforward Net with External Recurrence and Batch Learning [Qin92], and many others.

BIBLIOGRAPHY

[Aha91] D.W. Aha, D. Kibler, and M.K. Albert, "Instance-based learning algorithms", *Machine Learning*, Vol. 6, pp. 37-66, 1991.

[Atlas90] Les Atlas, Ronald Code, Jerome Connor, Mohamed El-Sharkawi, Robert Marks II, Yeshwant Muthusamy, and Etienne Barnard, "Performance Comparisons Between Backpropagation Networks and Classification Trees on Three Real-World Applications", *Advances in Neural Information Processing Systems*, Vol.2, pp. 622-629, 1990.

[Anderson35] E. Anderson, The Irises of the Gaspe Peninsula. *Bull. Amer. Iris Soc.*, 59:2-5, 1935.

[Breiman84] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification And Regression Trees*, Wadsworth, Belmont, CA, 1984.

[Cameron-Jones92] R.M. Cameron-Jones, "Minimum description length instance-based learning", *Proceedings of Australian Joint Conference on Artificial Intelligence*, World Scientific Publisher, pp. 368-373, 1992.

[Carbonell83] J.G. Carbonell, R.S. Michalski, and T.M. Mitchell, "An Overview of Machine Learning", *Machine Learning, an Artificial Intelligence Approach*, Tioga Publishing, Palo Alto, CA, pp. 3-23, 1983.

[Chen94] Hsinchun Chen, Peter Butin Rinde, Linlin She, Siunie Sitjahjo, Chris Sommer, and Daryl Neely, "Expert Prediction, Symbolic Learning, and Neural Networks", *IEEE Expert*, pp. 21-27, December 1994.

[Cestnik87] G. Cestnik, I. Konenenko, & J. Bratko, Assistant-86: A Knowledge-Elicitation Tool for Sophisticated Users, *Progress in Machine Learning*, pp. 31-45, Sigma Press, 1987.

[Clark87] Clark,P. & Biblett,T., Induction in Noisy Domains, *Progress in Machine Learning*, pp. 11-30, Sigma Press, 1987.

[Diaconis83] Diaconis,P. & Efron,B., Computer-Intensive Methods in Statistics, *Scientific American*, Vol. 248, 1983.

[Dietterich90] T.G. Dietterich, H. Hild, and G. Bakiri, "A comparative study of ID3 and backpropagation for English text-to-speech mapping", *Proceedings of the Seventh International Workshop on Machine Learning*, pp. 24-31, 1990.

[Duda73] R. Duda and P. Hart, *Pattern Classification and Scence Analysis*, Wiley, New York, 1973.

[Fast68] J.D. Fast, *Entropy. The significance of the concept of entropy and its application in science and technology*, Gordon and Breach, New York, 1968

[Fisher89] Douglas H. Fihser and Kathleen B. McKusick, "An Empirical Comparison of ID3 and Back-propagation", *Machine Learning*, pp. 788-793, 1989.

[Forsyth89] Richard Forsyth, *Machine Learning: principles and techniques*, Chapman and Hall Ltd., New York, NY, 1989.

[Halgamuge93a] S.K. Halgamuge, W. Poechmueller, and M.Glesner. "A Rule based Prototype System for Automatic Classification in Industrial Quality Control", *IEEE International Conference on Neural Networks' 93*, pp. 238-243, San Francisco, CA, March 1993.

[Halgamuge93b] S.K. Halgamuge, W. Poechmueller, S. Ting, M. Hoehn, and M. Glesner, "Identification of Underwater Sonar Image Using Fuzzy Neural Architecture Fune I". *International Conference on Artificial Neural Networks' 93*, pp. 922-925, Amsterdam, The Netherlands, September 1993.

[Halgamuge94a] S.K. Halgamuge and M. Glesner, "Fuzzy Neural Fusion Techniques for Industrial Applications," *ACM Symposium on Applied Computing (SAC'94)*, Phoenix, March 1994.

[Halgamuge94b] S.K. Halgamuge and M. Glesner, "Neural Networks in Designing Fuzzy Systems for Real World Applications," *International Journal for Fuzzy Sets and Systems*, North Holland, 1994.

[Holsheimer94] M. Holsheimer, A. Siebes, *Data Mining: the search for knowledge in database*, Report, Computer Science/Department of Algorithmics and Architecture, Amsterdam, Netherlands, 1994.

[Holte93] R.C. Holte, "Very simple classification rule perform well on most data sets", *Machine Learning*, Vol. 11, pp. 63-90, 1993.

[Horikawa92] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On Fuzzy Modeling Using Fuzzy Neural Networks with the Backpropagation Algorithm", *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, 1992.

[Kawamura92] A. Kawamura, N Watanabe, H. Okada, and K. Asakawa, "A Prototype of Neuro-Fuzzy Cooperation System", *IEEE International Conference on Fuzzy Systems*, pp. 1275-1282, San Diego, USA, 1992.

[Kohavi95] Ron Kohavi, *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*, Computer Science Department, Stanford University, Stanford, CA, 1995.

[Kononenko91] Igor Kononenko and Ivan Bratko, "Information-Based Evaluation Criteria for Classifier Performance", *Machine Learning*, Vol. 6, pp. 67-80, 1991.

[Kosko92] B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice-Hall, USA, 1992.

[Krisar95] Johan Krisar, *Noise Handling in Inductive Learning*, Thesis, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, 1995.

[Lin92] C.T. Lin and C.S.G. Lee, "Real-Time Supervised Structure/Parameters Learning for Fuzzy Neural Network", *IEEE International Conference on Fuzzy Systems*, pp. 1283-1291, San Diego, USA, 1992.

[Liu94] Xiaoji Liu, *A comparison study of feedforward fully connected neural networks vs. cascade correlation networks for prediction of soil moisture content*, Master Thesis, Computer Science Department, Oklahoma State University, Stillwater, OK, 1994.

[Malik94] Naeem Malik, *Application of Fuzzy Logic to Distress Analysis*, Master Thesis, Oklahoma State University, Stillwater, OK, 1994.

[Michalski80] R.S. Michalski and R.L. Chilausky, "Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis", *International Journal of Policy Analysis and Information Systems*, Vol. 4, No. 2, pp. 125-161, 1980.

[Michalski86a] R.S. Michalski, J.G. Carbonell and T.M. Mitchell, *Machine Learing: An Artifical Intelligence Approach - Volume II*, Morgan Kaufmann, CA. USA, 1986.

[Michalski86b] R. Michalski, I. Mosetic, J. Hong, & N. Lavrac, The Multi-Purpoase Incremental Learning System AQ15 and its Testing Applications to Three Medical Domains, *Proceedings of the Fifth National Conference on AI*, pp. 1041-1045, Philadelphia, PA: Morgan Kaufman, 1986.

[Mooney91] R. Mooney et al., "An Experimental Comparison of Symbolic and Connectionist Learning Algorithms", *Proceedings 11th International Conference of AI*, Morgan Kaufman, San Francisco, CA, 1991, pp. 775-780.

[Murphy91] P.M. Murphy and D.W. Aha, *UCI Repository of machine learning database*, Department of Information and Computer Science, University of California, Irvine, CA, 1994.

[Murthy94] S.K. Murthy, S. Kasif, and S. Salzberg, "A System for Induction of Oblique Decision Trees", *Journal of Artificial Intelligence Research*, 2, pp. 1-32, 1994.

[Qin92] Si-Zhao Qin, Hong-Te Su, and Thomas J. McAvoy, "Comparison of Four Neural Net Learning Methods for Dynamic System Identification", *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, 1992.

[Quinlan79] J.R. Quinlan, "Discovery Rules by Induction from Large Collections of Examples", *Expert Systems in the Micro-electronic Age (ed. Michie, D.)*, Edinburgh University Press, Edinburgh, 1979.

[Quinlan82] J.R. Quinlan, "Semi-Autonomous Acquisition of Pattern-Based Knowledge", *Introductory Readings in Expert System*, Gordon and Breach Science Publishers, New York, 1982.

[Quinlan86] J.R. Quinlan, "Induction of decision trees," *Machine Learning*, Vol. 1, pp. 81-106, 1986.

[Quinlan89] J.R. Quinlan, *Learning relations: Comparison of a symbolic and connectionist approach*, University of Sydney, N.S.W., Australia, 1989.

[Quinlan93] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufman, San Meteo, CA, 1993.

[Rendell83] L.A. Rendell, "A new basis for state-space learning system and a successful implementation", *Artificial Intelligence*, 20, pp. 369-392, 1983.

[Roan93] Sing-Ming Roan, Cheng-Chin Chiang, and Hsin-Chia Fu, "Fuzzy RCE Neural Network", *International Conference on Fuzzy Systems*, pp. 629-634, 1993.

[Rosenblatt62] F. Rosenblatt, *Principles of Neuradynamics*, Spartan, New York, 1962.

[Rudin64] Walter Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, New York, 1964.

[Rumelhart86] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation", *Parallel Distributed Processing*, MIT Press, Cambridge, Mass., 1986, pp. 318-362.

[Shannon64] C.E. Shannon, *The mathematical theory of communication*, University of Illinois Press, Urbana, Illinois, 1964.

[Simon91] H. Simon, "Artificial Intelligence: Where Has It Been, and Where is it Going?", *IEEE Transaction Knowledge and Data Engineering*, Vol. 3, No. 2, pp. 128-136, June 1991.

[Smith88] J.W. Smith, J.E. Everhart, W.C. Dickson, W.C. Knowler, & R.S. Johannes., Using the ADAP learning algorithm to forecast the onset of diabetes mellitus, *Proceedings of the Symposium on Computer Applications and Medical Care*, pp. 261-265, IEEE Computer Society Press, 1988.

[Towell90] G. Towell, J. Shavlik, and M. Noordewier, Refinement of Approximate Domain Theories by Knowledge-Based Artificial Neural Networks, *Proceedings of the Eighth National Conference on AI*, 1990.

[Tsaptsinos90] D. Tsaptsinos, A.R. Mirzai, and B.W. Jervis, "Comparison of machine learning paradigms in a classification task", *Proceedings of the 5th International Conference on AI in Engineering*, Vol. 2, Boston, Mass., USA., pp. 323-339, July 1990.

[Wang92a] Li-Xiu Wang and Jerry M. Mendel, "Back-propagation Fuzzy System as Nonlinear Dynamic System Identifiers", *International Conference on Fuzzy Systems*, pp. 1409-1418, San Diego, 1992.

[Wang92b] L.X. Wang and J.M. Mendel, "Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least Square Learning", *IEEE Transactions on Neural Networks*, 3(5), pp. 807-814, 1992.

[Wang94] Shiang-Huey Wang, *Comparison of backpropagation neural networks and general regression neural networks*, Master Thesis, Computer Science Department, Oklahoma State University, Stillwater, OK, 1994.

[Wasserman89] Philip D. Wasserman, *Neural Computing: theory and practice*, Van Nostrand Reinhold, New York, NY, 1989.

[Weiss89] S.M. Weiss and I. Kapouleas, "An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods", *Proc. 11th International Joint Conference of AI*, Morgan Kaufman, San Francisco, CA, 1989, pp. 781-787.

[Weiss90] S.M. Weiss and Casimir Kulikowski, *Computer Systems That Learn*, Morgan Kaufman Publisher, Inc., San Mateo, CA, 1990.

[Wu94] Jian-Kang Wu, *Neural networks and simulation methods*, Marcel Dekker Inc., New York, NY, 1994.

[Yazdani86] Masoud Yazdani, *Artificial Intelligence: principles and applications*, Chapman and Hall Ltd., New York, NY, 1986.

[Zadeh65] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, Vol. 8, pp. 338-353, 1965.

[Zeidenberg90] Matthew Zeidenberg, *Neural network models in artificial intelligence*, Ellis Horwood Limited, West Sussex, 1990.

[Zhang92] J. Zhang, "Selecting typical instances in instance-based learning", *Proceedings of the Ninth International Machine Learning Conference*, pp.470-479. Aberden, Scotland: Morgan Kaufman, 1992.

[Zheng92] Z. Zheng, "Constructing conjunctive tests for decision trees", *Proceedings of Australian Joint Conference on Artificial Intelligence*, World Scientific Publisher, pp. 335-360, 1992.

[Zheng93] Zijian Zheng, *A Benchmark for Classifier Learning*, Technical Report, Basser Department of Computer Science, University of Sydney, N.S.W., Australia, November 1993.

APPENDIXES

APPENDIX A

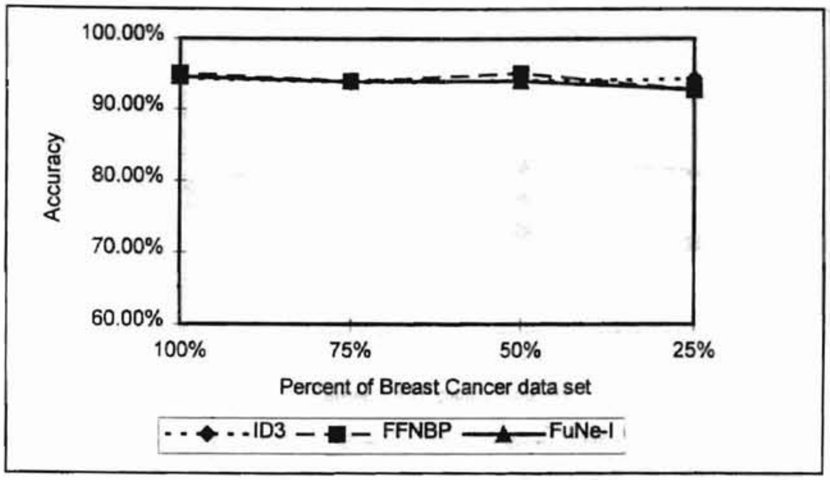
DATA SET DIMENSIONS

DataSet	Size	Missing	Noise Level		# Attributes				#	#	#
		Values	Att.	Cl.	B	N	C	T	IAtt	DNAV	Cl.
Breast Cancer (W)	699	16(0.25)	yes	yes	0	0	9	9			2
Diabetes	768	0			0	0	8	8			2
Hepatitis	155	167(5.67)			13	0	6	19			2
LED-24	200	0	yes	no	24	0	0	24	17(70.8)		10
LED-7	200	0	yes	no	7	0	0	7	0		10
Lymphography	148	0			9	9	0	18		8	4
Monks-2	432	0	no	no	2	4	0	6	0	4	2
Mushroom	8124	2480(1.39)			4	18	0	22		12	2
NetTalk(Phoneme)	5438	0	no	yes	0	7	0	7	0	27	52
Promoter	106	0			0	57	0	57		4	2
Soybean	683	2337(9.78)	yes	yes	16	19	0	35		7	19
Thyroid	3163	5329(6.74)	yes	yes	18	0	7	25			2
Waveform-40	300	0	yes	no	0	0	40	40	19(47.5)		3

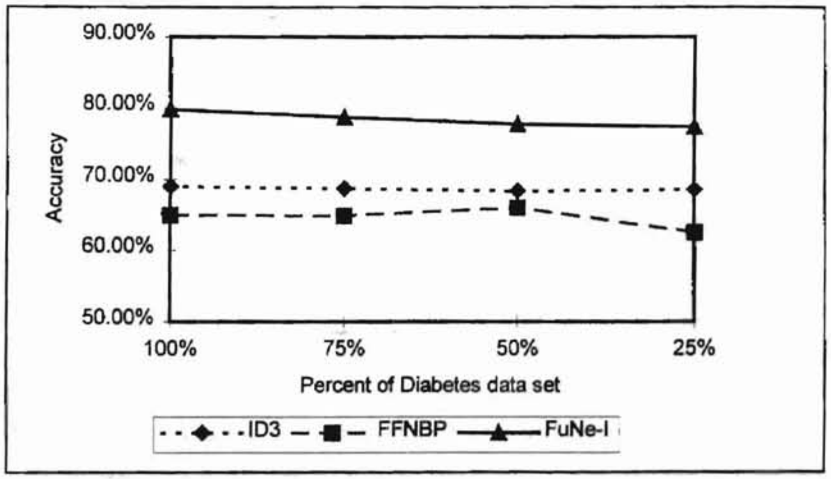
DataSet	Density	Default	Entropy	Highest Acc.		Info. S. of C4.5	
		Acc.	(bits)	Pred.	Rel.	Average.	Rel.
Breast Cancer (W)	7.77×10^{-7}	65.5	0.93	94.8	84.9	0.81	87.5
Diabetes	1.12×10^{-13}	65.1	0.93	78.8	39.3	0.32	34.1
Hepatitis	1.28×10^{-12}	79.4	0.73	83.0	17.5	0.14	19.1
LED-24	1.19×10^{-5}	14.1	3.28	70.0	65.1	1.64	56.1
LED-7	1.56	14.1	3.28	71.0	66.2	1.92	66.2
Lymphography	4.90×10^{-7}	54.7	1.23	78.4	52.3	0.61	54.3
Monks-2	1.00	62.1	0.96	100.0	100.0	0.15	16.6
Mushroom	5.79×10^{-12}	51.8	1.00	100.0	100.0	1.00	100.0
NetTalk(Phoneme)	5.20×10^{-7}	18.7	4.72	84.1	80.4	3.70	79.4
Promoter	5.10×10^{-33}	51.5	1.00	76.3	51.1	0.41	42.1
Soybean	5.47×10^{-13}	13.7	3.84	97.1	96.6	3.35	91.4
Thyroid	1.32×10^{-17}	95.2	0.28	99.1	81.2	0.24	85.6
Waveform-40	7.31×10^{-92}	39.0	1.56	86.0	77.0	0.83	54.0

APPENDIX B:

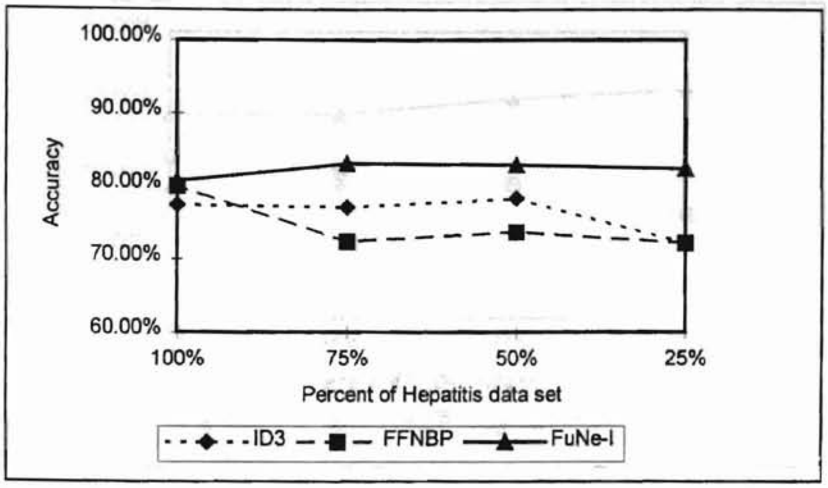
GRAPHS WITH REDUCING DATA



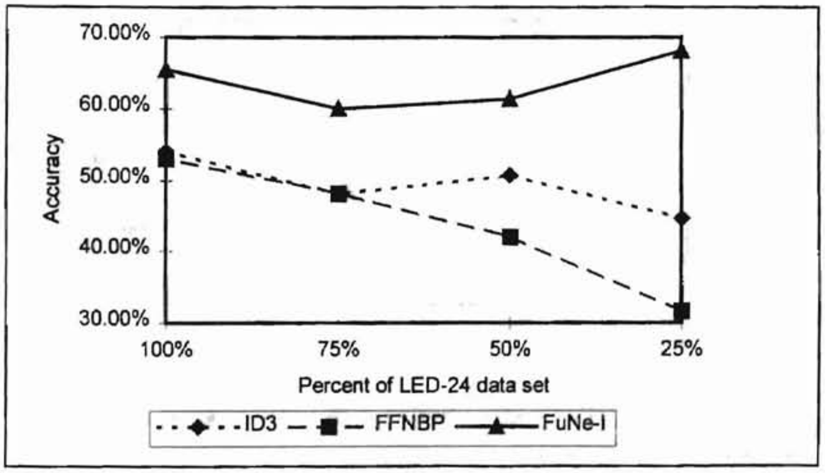
Graph B1: Classification performance as a function of percent of Breast Cancer data set.



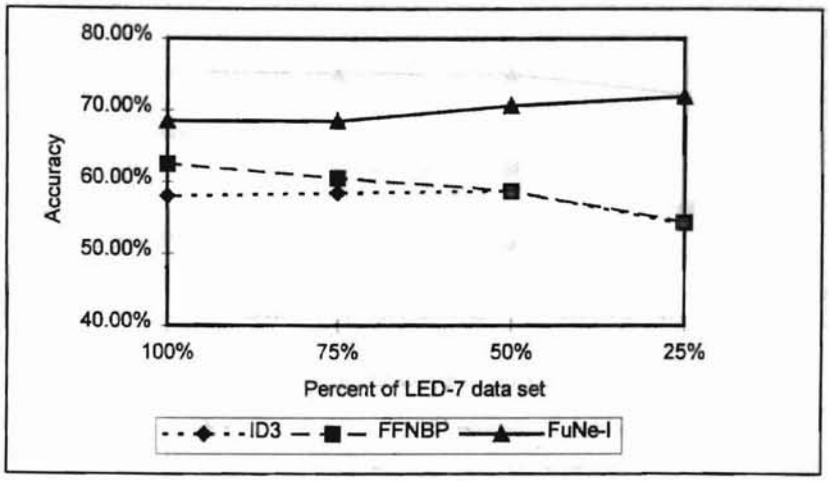
Graph B2: Classification performance as a function of percent of Diabetes data set.



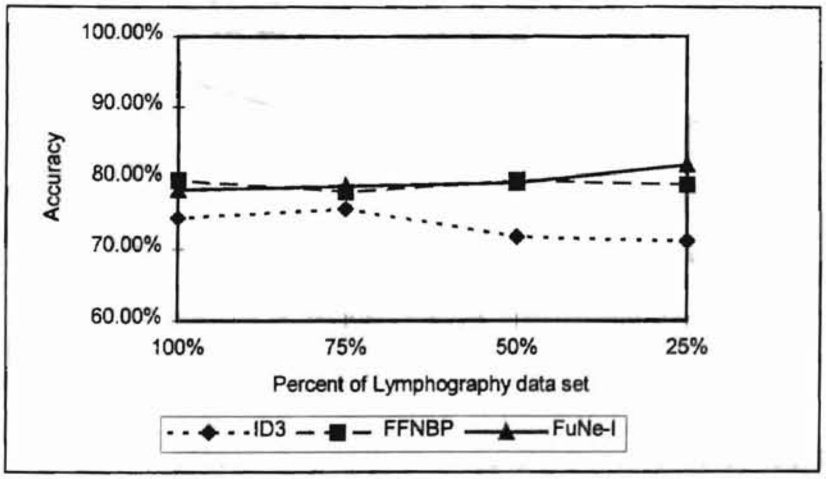
Graph B3: Classification performance as a function of percent of Hepatitis data set.



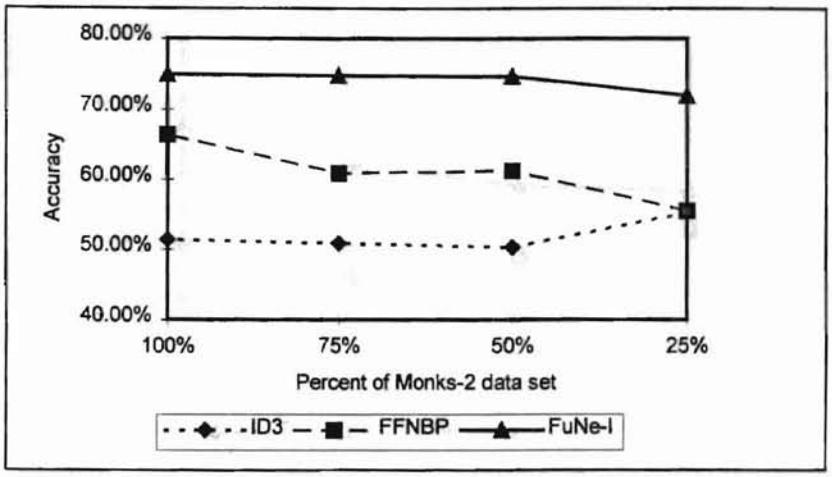
Graph B4: Classification performance as a function of percent of LED-24 data set.



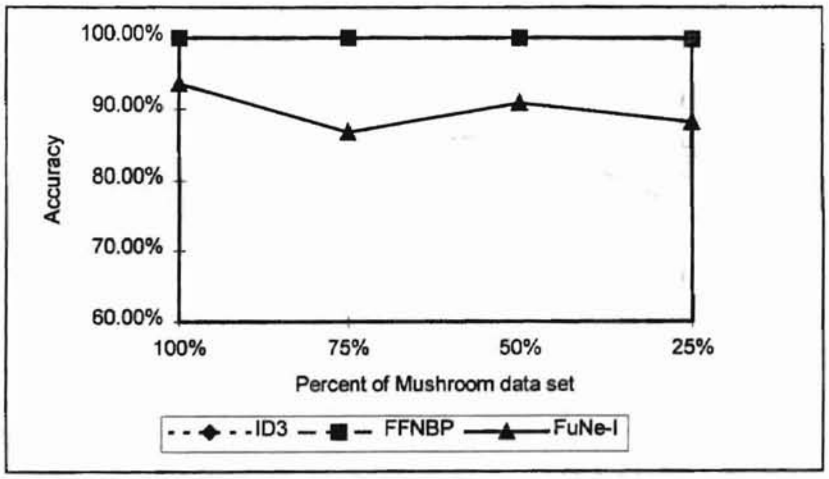
Graph B5: Classification performance as a function of percent of LED-7 data set.



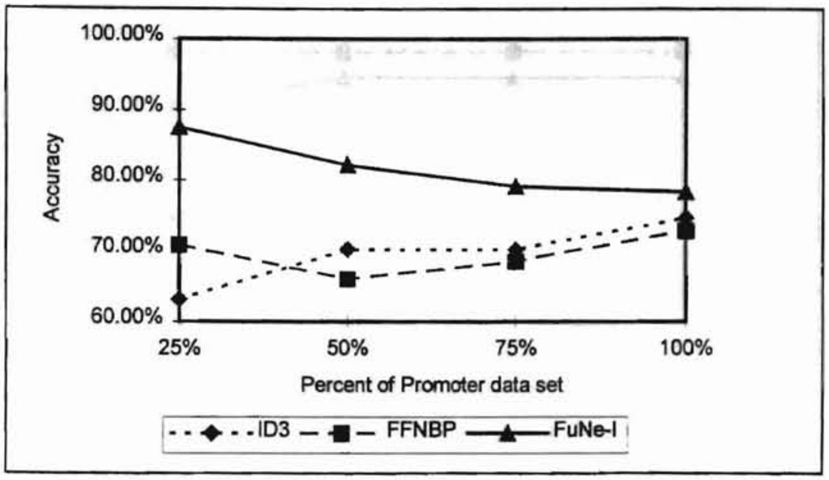
Graph B6: Classification performance as a function of percent of Lymphography data set.



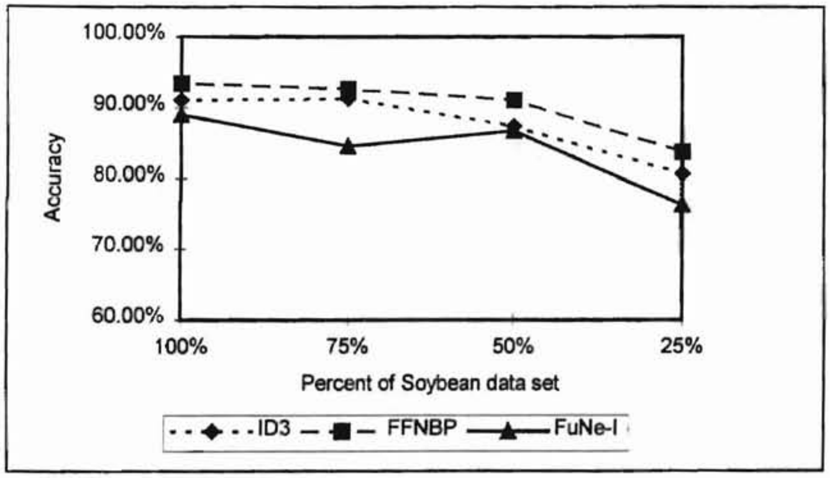
Graph B7: Classification performance as a function of percent of Monks-2 data set.



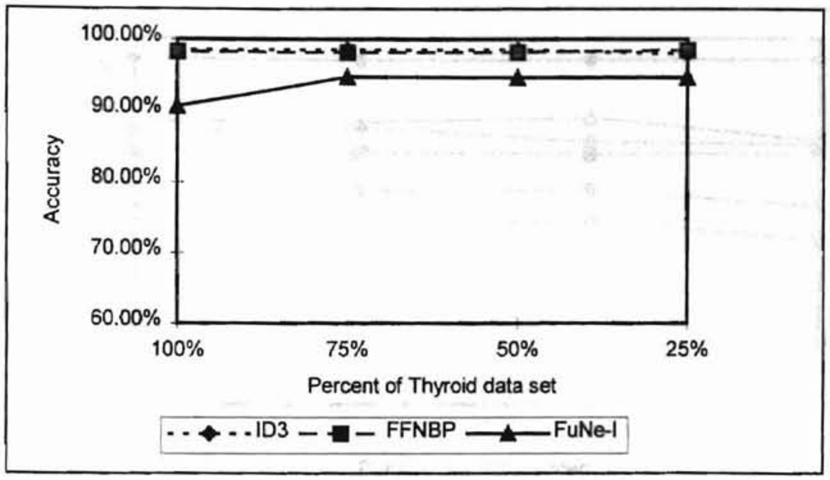
Graph B8: Classification performance as a function of percent of Mushroom data set.



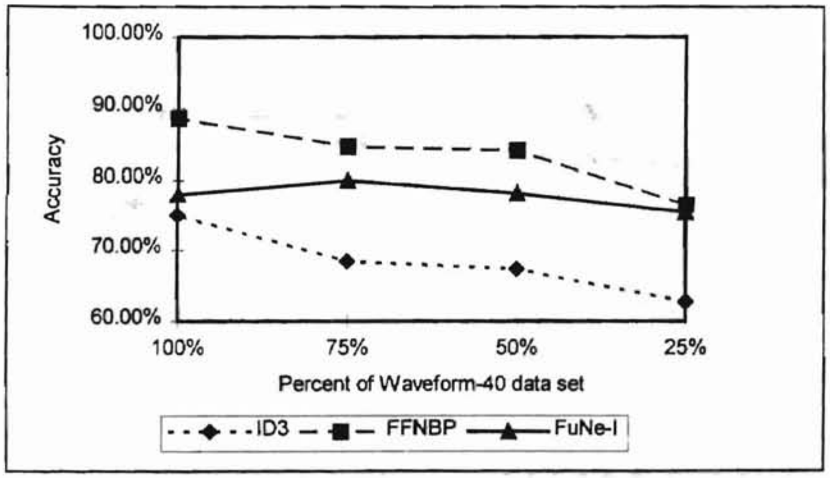
Graph B9: Classification performance as a function of percent of Promoter data set.



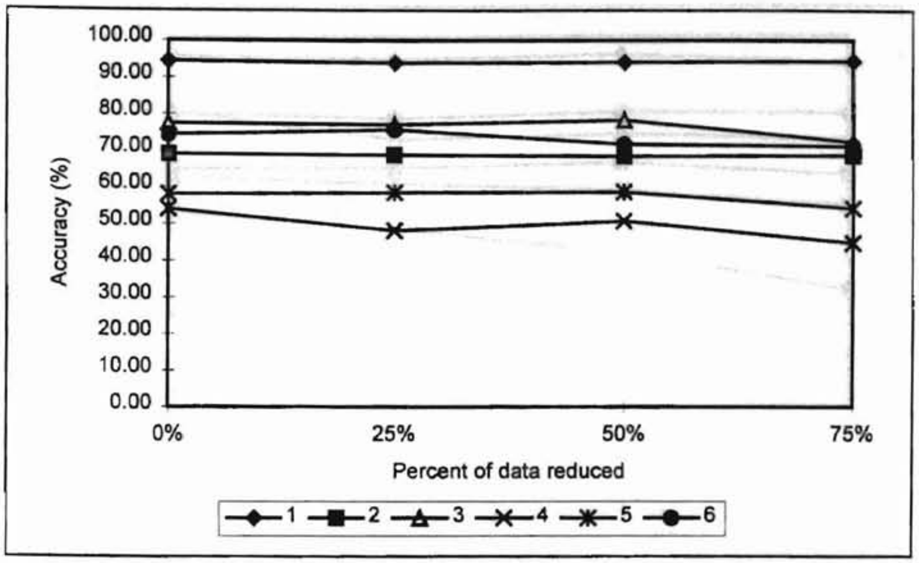
Graph B10: Classification performance as a function of percent of Soybean data set.



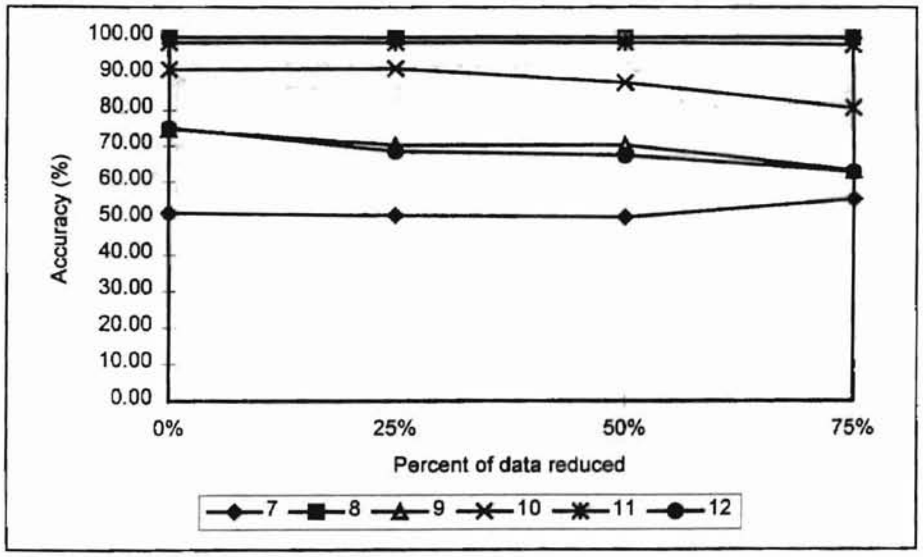
Graph B11: Classification performance as a function of percent of Thyroid data set.



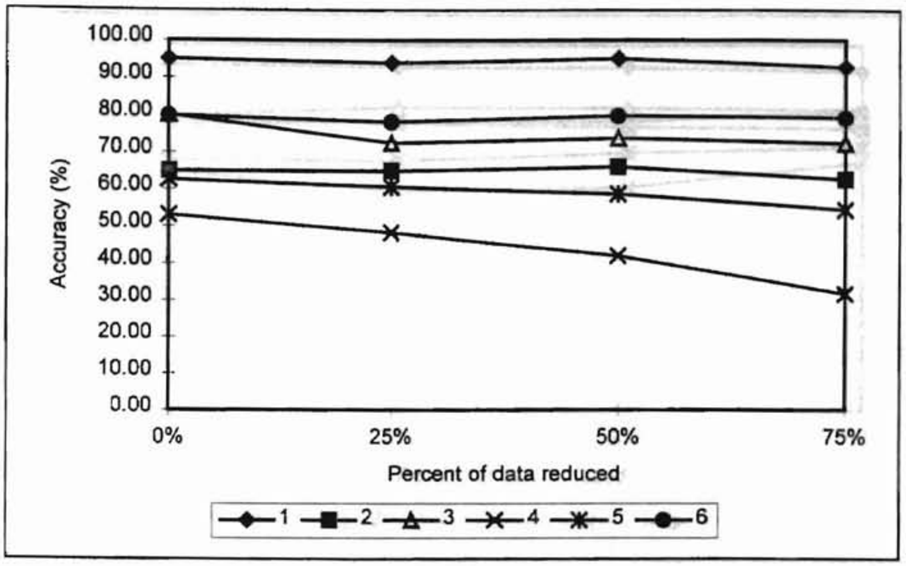
Graph B12: Classification performance as a function of percent of Waveform-40 data set.



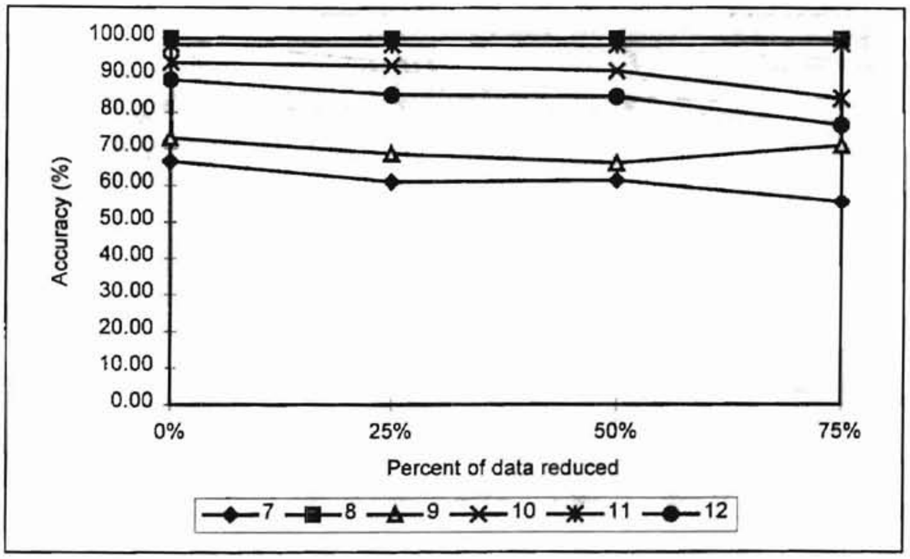
Graph B13a: Classification performance of ID3 as a function of percent of data reduced for data sets 1 to 6.



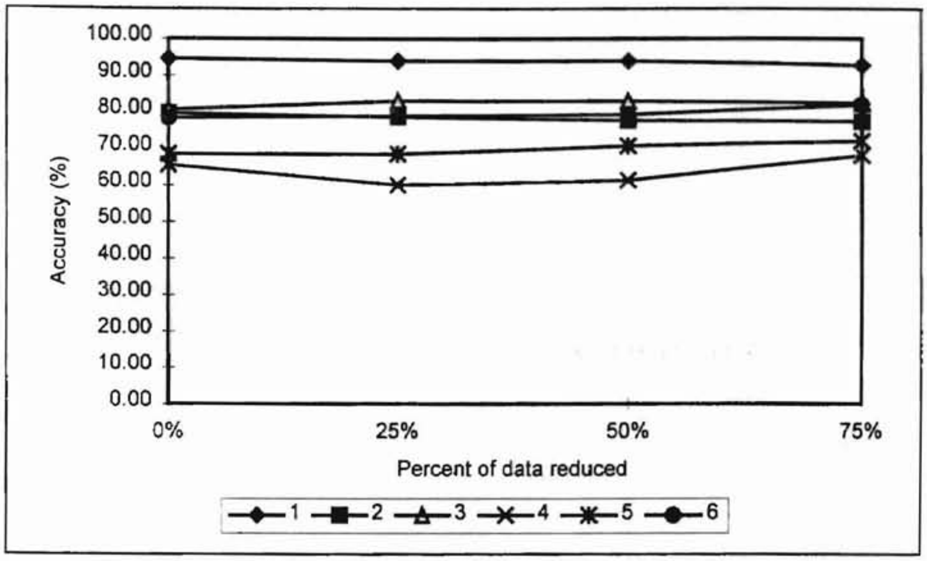
Graph B13b: Classification performance of ID3 as a function of percent of data reduced for data sets 7 to 12.



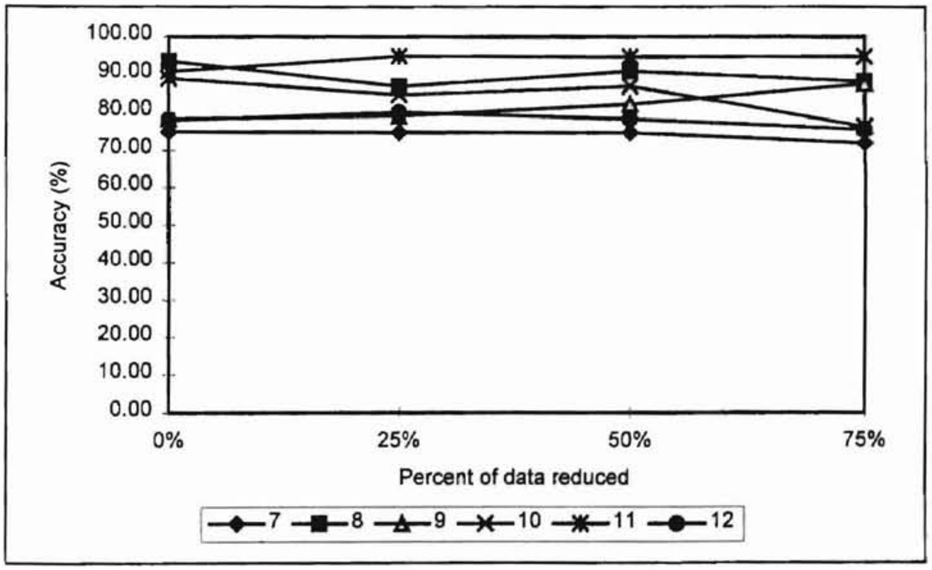
Graph B14a: Classification performance of FFNBP as a function of percent of data reduced for data sets 1 to 6.



Graph B14b: Classification performance of FFNBP as a function of percent of data reduced for data sets 7 to 12.



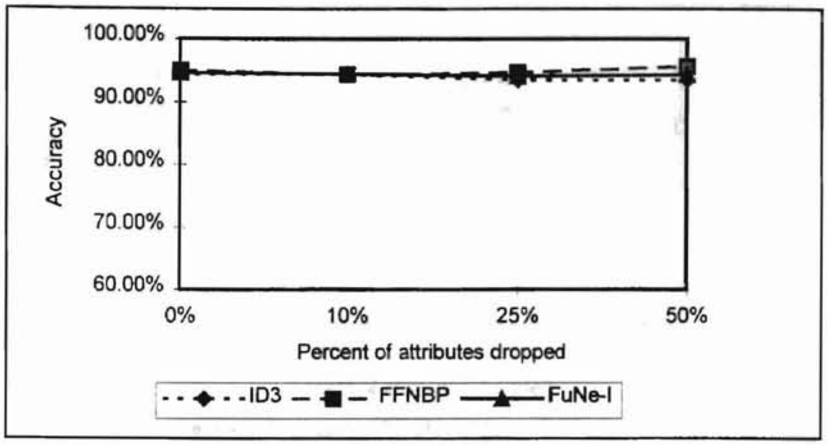
Graph B15a: Classification performance of FuNe-I as a function of percent of data reduced for data sets 1 to 6.



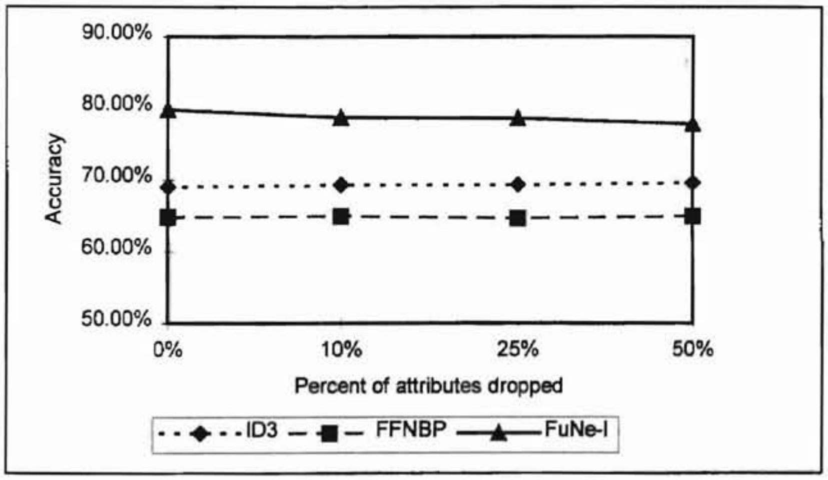
Graph B15b: Classification performance of FuNe-I as a function of percent of data reduced for data sets 7 to 12.

APPENDIX C:

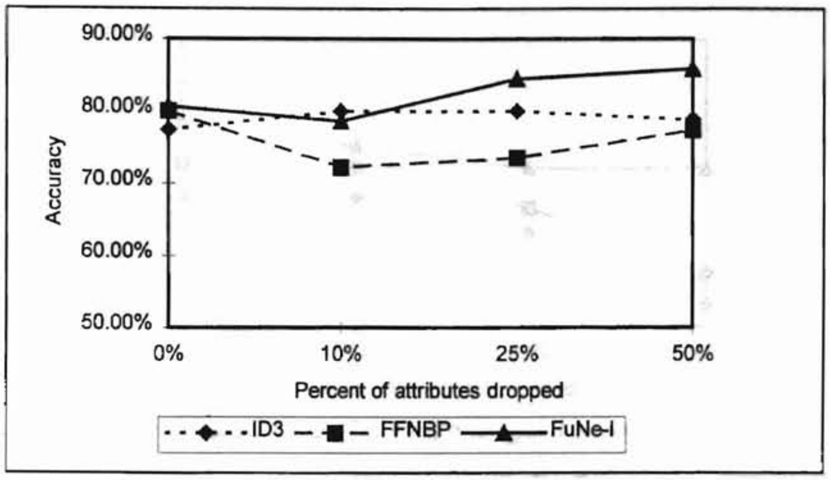
GRAPHS WITH DROPPING ATTRIBUTES



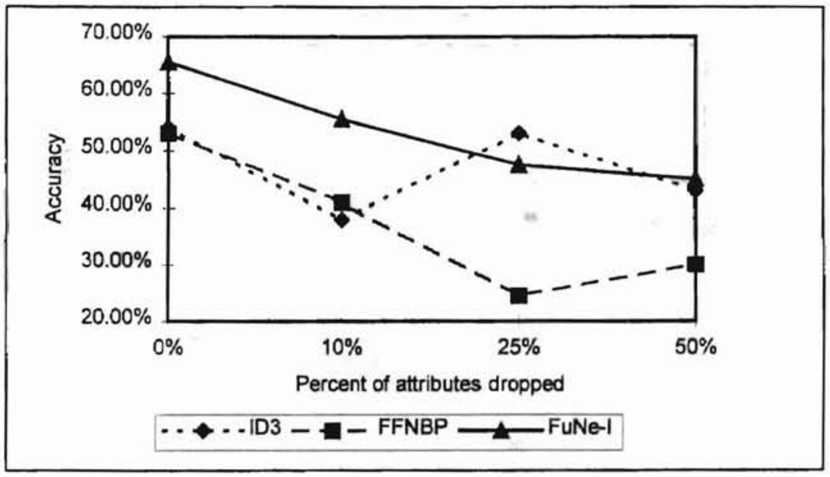
Graph C1: Classification performance as a function of percent of attributes dropped for Breast Cancer data set.



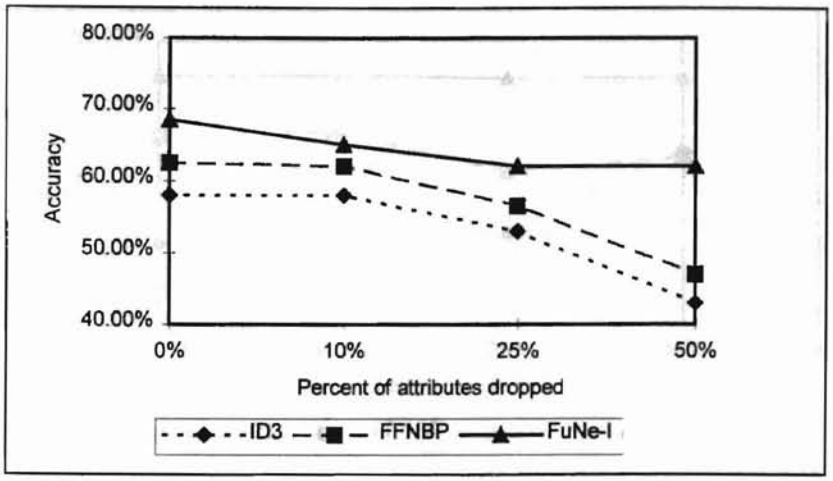
Graph C2: Classification performance as a function of percent of attributes dropped for Diabetes data set.



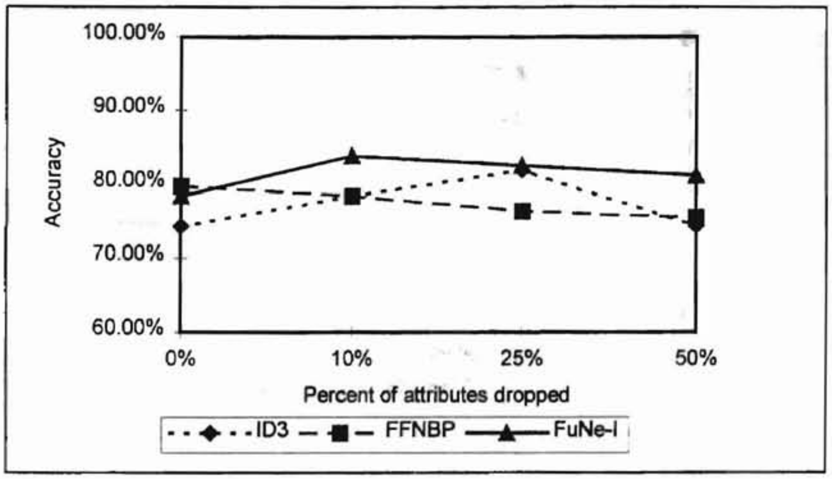
Graph C3: Classification performance as a function of percent of attributes dropped for Hepatitis data set.



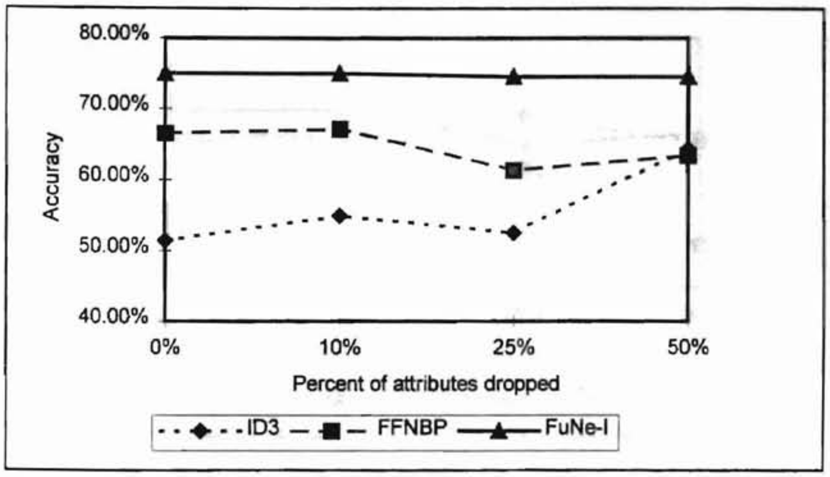
Graph C4: Classification performance as a function of percent of attributes dropped for LED-24 data set.



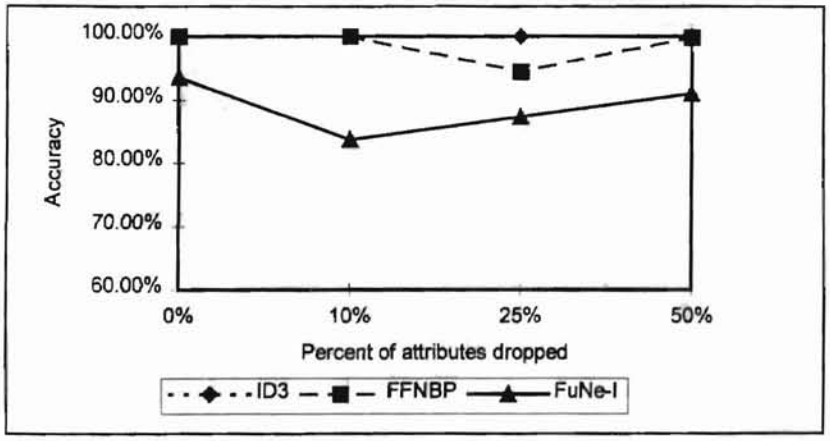
Graph C5: Classification performance as a function of percent of attributes dropped for LED-7 data set.



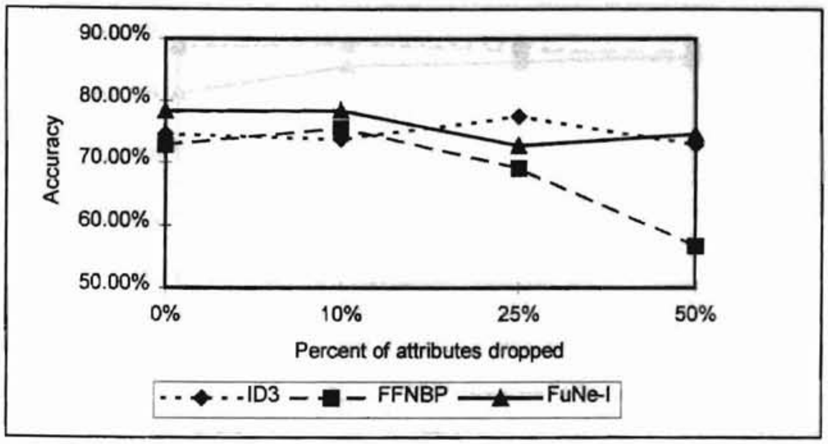
Graph C6: Classification performance as a function of percent of attributes dropped for Lymphography data set.



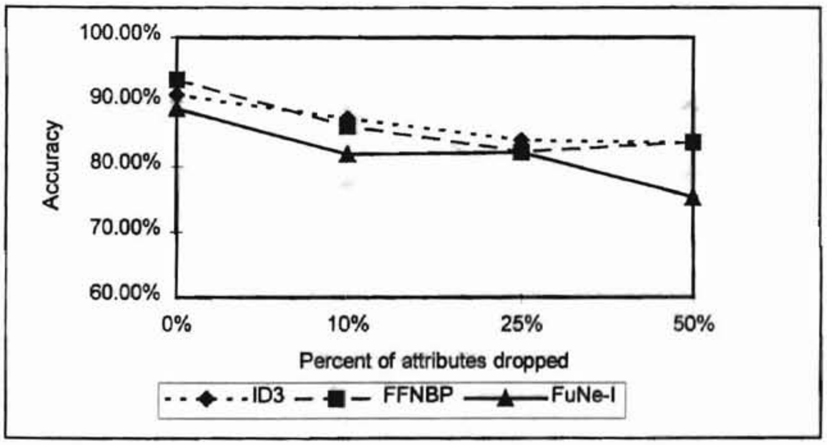
Graph C7: Classification performance as a function of percent of attributes dropped for Monks-2 data set.



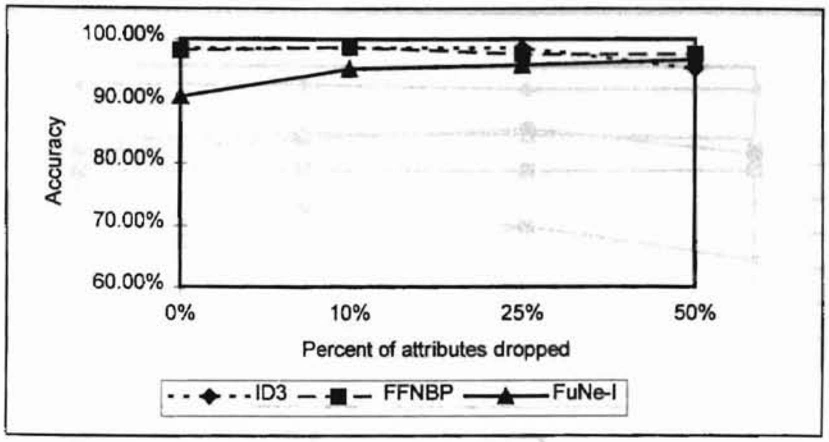
Graph C8: Classification performance as a function of percent of attributes dropped for Mushroom data set.



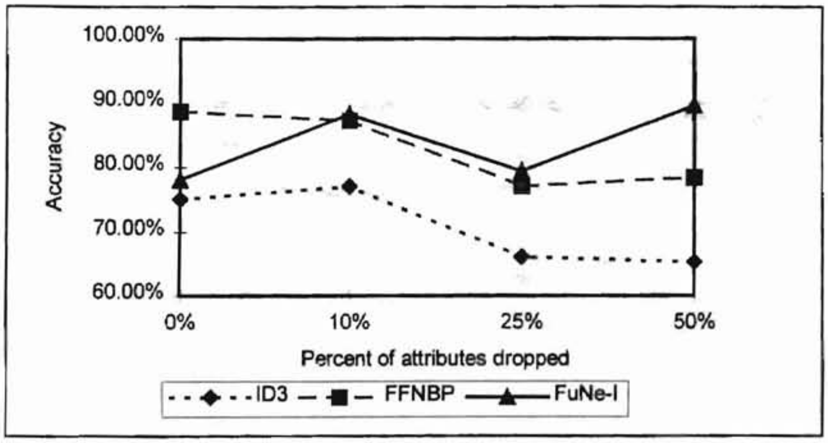
Graph C9: Classification performance as a function of percent of attributes dropped for Promoter data set.



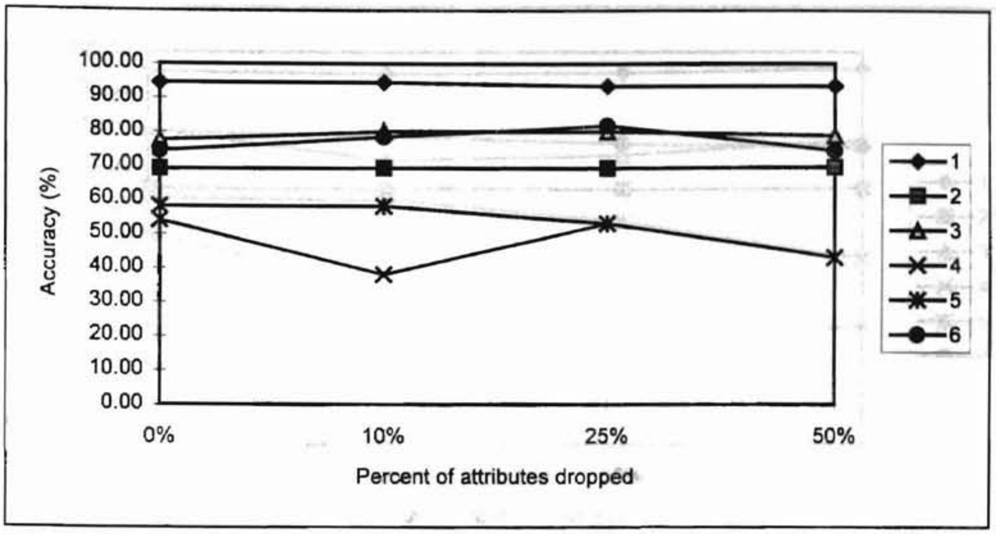
Graph C10: Classification performance as a function of percent of attributes dropped for Soybean data set.



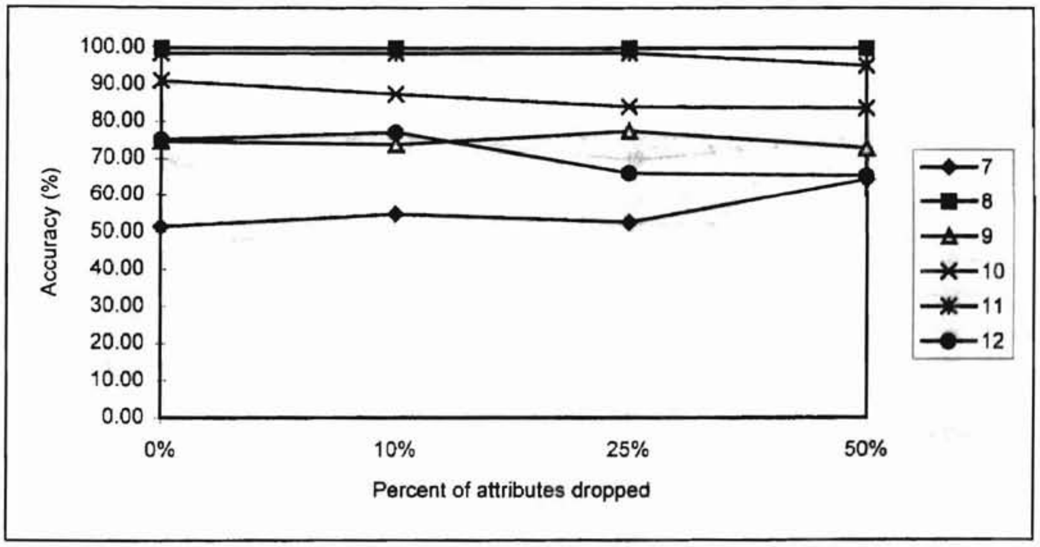
Graph C11: Classification performance as a function of percent of attributes dropped for Thyroid data set.



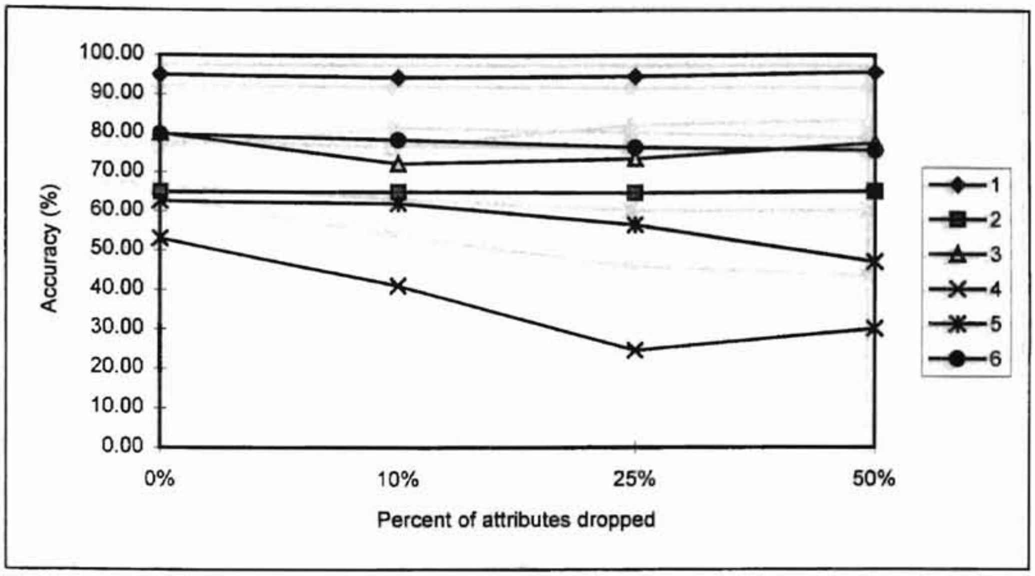
Graph C12: Classification performance as a function of percent of attributes dropped for Waveform-40 data set.



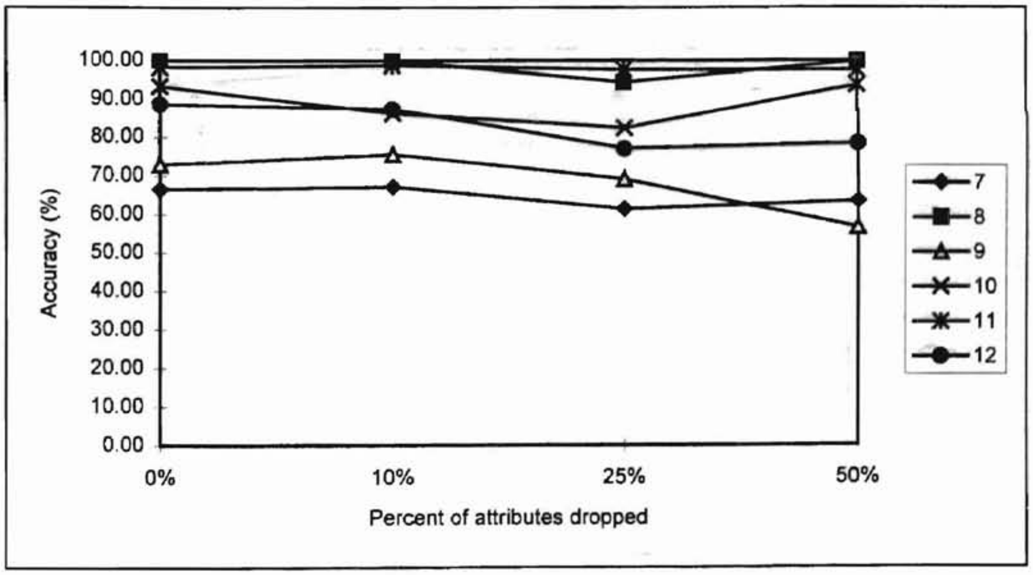
Graph C13a: Classification performance of ID3 as a function of percent of attributes dropped for data sets 1 to 6.



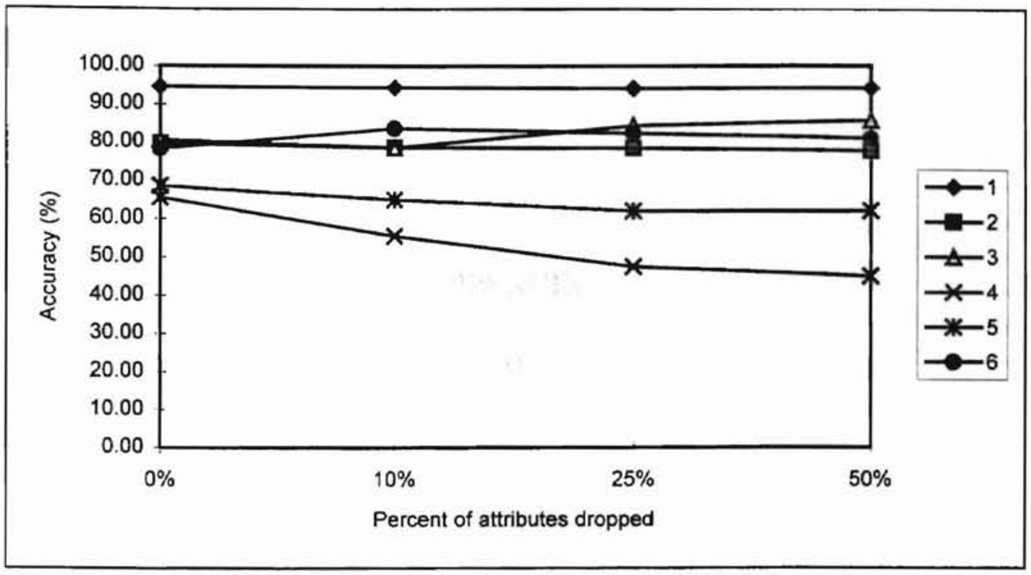
Graph C13b: Classification performance of ID3 as a function of percent of attributes dropped for data sets 7 to 12.



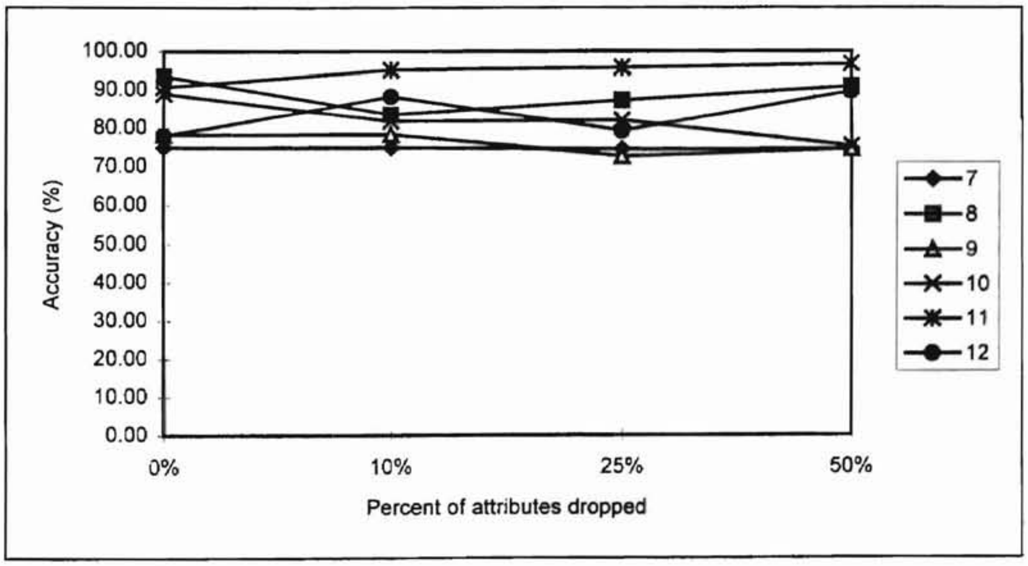
Graph C14a: Classification performance of FNNBP as a function of percent of attributes dropped for data sets 1 to 6.



Graph C14b: Classification performance of FFNBP as a function of percent of attributes dropped for data sets 7 to 12.



Graph C15a: Classification performance of FuNe-I as a function of percent of attributes dropped for data sets 1 to 6.



Graph C15b: Classification performance of FuNe-I as a function of percent of attributes dropped for data sets 7 to 12.

APPENDIX D

TABLES

Table D1: Classification accuracy of ID3, FFNBP, and FuNe-1 using 4-fold cross-validation.

Data Set #	Data Set	ID3	FFNBP	FuNe-I
1	Breast Cancer	94.42	***94.99***	94.56
2	Diabetes	69.01	64.89	***79.69***
3	Hepatitis	77.41	79.99	***80.65***
4	LED-24	54.00	53.00	***65.50***
5	LED-7	58.00	62.50	***68.50***
6	Lymphography	74.32	***79.73***	78.38
7	Monks-2	51.39	66.44	***75.00***
8	Mushroom	***100.00***	***100.00***	93.50
9	Promoter	74.61	72.80	***78.30***
10	Soybean	91.07	***93.41***	88.97
11	Thyroid	***98.51***	98.23	90.76
12	Waveform-40	75.00	***88.67***	78.00
	Total	917.74	954.65	971.81
	Average	76.48	79.55	***80.98***

Table D2: Classification accuracy of ID3 using 4-fold cross-validation.

Data Set #	Data Set	Total # of Data	Time: User (mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	699	00:04.16	94.42
2	Diabetes	768	00:09.55	69.01
3	Hepatitis	155	00:01.78	77.41
4	LED-24	200	00:02.66	54.00
5	LED-7	200	00:01.08	58.00
6	Lymphography	148	00:01.20	74.32
7	Monks-2	432	00:02.64	51.39
8	Mushroom	8124	00:29.00	100.00
9	Promoter	106	00:01.35	74.61
10	Soybean	683	00:06.62	91.07
11	Thyroid	3163	00:25.60	98.51
12	Waveform-40	300	00:12.51	75.00
	Total		01:38.15	917.74
	Average		00:08.18	76.48

Table D3: Classification accuracy of FFNBP using 4-fold cross-validation.

Data Set #	Data Set	Network	Momentum	Learning Rate	Total # of Data	Time: User (h:mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	9-9-2	0.9	0.002	699	0:04:20.00	94.99
2	Diabetes	8-8-2	0.9	0.002	768	0:04:02.93	64.89
3	Hepatitis	19-19-2	0.9	0.0005	155	0:03:09.96	79.99
4	LED-24	24-24-10	0.9	0.001	200	0:02:57.75	53.00
5	LED-7	7-7-10	0.9	0.001	200	0:01:49.08	62.50
6	Lymphography	18-18-4	0.9	0.005	148	0:00:37.78	79.73
7	Monks-2	6-6-2	0.9	0.003	432	0:01:39.06	66.44
8	Mushroom	22-22-2	0.9	0.0001	8124	1:12:42.97	100.00
9	Promoter	57-57-2	0.9	0.005	106	0:00:06.71	72.80
10	Soybean	35-35-19	0.9	0.001	683	1:04:06.22	93.41
11	Thyroid	25-25-2	0.9	0.0001	3163	1:44:50.69	98.23
12	Waveform-40	40-40-3	0.9	0.004	300	0:00:42.40	88.67
	Total					4:21:05.55	954.65
	Average					0:21:45.46	79.55

Table D4: Classification accuracy of FuNe-I using 4-fold cross-validation.

Data Set #	Data Set	Momentum	Learning Rate	Total # of Data	Time: User (h:mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	0.90	0.002	699	0:44:48.93	94.56
2	Diabetes	0.90	0.002	768	0:47:24.11	79.69
3	Hepatitis	0.90	0.0001	155	0:36:53.96	80.65
4	LED-24	0.90	0.0001	200	0:56:36.41	65.50
5	LED-7	0.90	0.0001	200	0:18:47.26	68.50
6	Lymphography	0.90	0.002	148	0:28:24.94	78.38
7	Monks-2	0.90	0.005	432	0:17:13.91	75.00
8	Mushroom	0.90	0.0001	8124	16:39:33.85	93.50
9	Promoter	0.90	0.005	106	1:10:42.53	78.30
10	Soybean	0.90	0.001	683	5:40:14.87	88.97
11	Thyroid	0.90	0.0001	3163	8:02:45.58	90.76
12	Waveform-40	0.90	0.004	300	1:17:07.78	78.00
	Total				13:00:34.13	971.81
	Average				3:05:02.84	80.98

Table D5: Classification accuracy of ID3, FFNBP, and FuNe-1 using 10-fold cross-validation.

Data Set #	Data Set	ID3	FFNBP	FuNe-I
1	Breast cancer	94.13	***95.01***	93.71
2	Diabetes	70.57	66.26	***80.61***
3	Hepatitis	77.33	73.00	***83.83***
4	LED-24	50.50	60.50	***61.00***
5	LED-7	60.00	63.50	***70.00***
6	Lymphography	78.38	76.56	***81.49***
7	Monks-2	49.52	67.15	***74.09***
8	Mushroom	***100.00***	***100.00***	84.06
9	Promoter	***83.27***	72.13	79.04
10	Soybean	***93.84***	91.99	81.68
11	Thyroid	***98.61***	98.13	92.69
12	Waveform-40	68.33	***83.33***	76.33
	Total	924.48	947.56	958.53
	Average	77.04	78.96	79.88

Table D6: Classification accuracy of ID3 using 10-fold cross-validation.

Data Set #	Data Set	Total # of Data	Time: User (mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	699	00:11.68	94.13
2	Diabetes	768	00:26.71	70.57
3	Hepatitis	155	00:04.63	77.33
4	LED-24	200	00:07.00	50.50
5	LED-7	200	00:02.66	60.00
6	Lymphography	148	00:02.84	78.38
7	Monks-2	432	00:07.17	49.52
8	Mushroom	8124	00:56.02	100.00
9	Promoter	106	00:03.00	83.27
10	Soybean	683	00:16.10	93.84
11	Thyroid	3163	01:06.60	98.61
12	Waveform-40	300	00:37.98	68.33
	Total		04:02.39	924.48
	Average		00:20.20	77.04

Table D7: Classification accuracy of FFNBP using 10-fold cross-validation

Data Set #	Data Set	Network	Momentum	Learning Rate	Total # of Data	Time:User (h:mm:ss.00)	Testing Accuracy (%)
1	Bcancer	9-9-2	0.90	0.002	699	0:05:24.62	95.01
2	Diabetes	8-8-2	0.90	0.002	768	0:04:58.38	66.26
3	Hepatitis	19-19-2	0.90	0.0001	155	0:03:36.04	73.00
4	LED-24	24-24-10	0.90	0.0001	200	0:09:36.16	60.50
5	LED-7	7-7-10	0.90	0.0001	200	0:02:14.77	63.50
6	Lymphography	18-18-4	0.90	0.002	148	0:00:41.62	76.56
7	Monks-2	6-6-2	0.90	0.005	432	0:02:06.67	67.15
8	Mushroom	22-22-2	0.90	0.0001	8124	0:59:55.45	100.00
9	Promoter	57-57-2	0.90	0.005	106	0:00:10.16	72.13
10	Soybean	35-35-19	0.90	0.001	683	1:17:30.01	91.99
11	Thyroid	25-25-2	0.90	0.0001	3163	2:08:23.50	98.13
12	Waveform-40	40-40-3	0.90	0.004	300	0:01:21.62	83.33
	Total					4:55:59.00	947.56
	Average					0:24:39.92	78.96

Table D8: Classification accuracies of FuNe-I using 10-fold cross-validation.

Data Set #	Data set	Momentum	Learning Rate	Total # of Data	Time: User (h:mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	0.90	0.002	699	0:49:38.58	93.71
2	Diabetes	0.90	0.002	768	0:52:21.31	80.61
3	Hepatitis	0.90	0.0001	155	0:41:03.52	83.83
4	LED-24	0.90	0.0001	200	0:52:19.19	61.00
5	LED-7	0.90	0.0001	200	0:23:51.11	70.00
6	Lymphography	0.90	0.002	148	0:40:18.02	81.49
7	Monks-2	0.90	0.005	432	0:18:56.34	74.09
8	Mushroom	0.90	0.0001	8124	18:15:48.19	84.06
9	Promoter	0.90	0.005	106	1:20:16.99	79.04
10	Soybean	0.90	0.001	683	6:39:49.54	81.68
11	Thyroid	0.90	0.0001	3163	9:32:33.34	92.69
12	Waveform-40	0.90	0.004	300	1:47:18.94	76.33
	Total				18:14:15.07	958.53
	Average				3:31:11.26	79.88

Table D9: Classification performance of ID3 with reducing data.

Data Set #	Data Set	% of Data Reduced	Total # of Data	Time: User (mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	0%	699	00:04.16	94.42
		25%	522	00:02.98	93.87
		50%	348	00:01.77	94.08
		75%	174	00:00.75	94.26
2	Diabetes	0%	768	00:09.55	69.01
		25%	576	00:06.67	68.69
		50%	384	00:04.36	68.40
		75%	192	00:01.84	68.58
3	Hepatitis	0%	155	00:01.78	77.41
		25%	114	00:01.30	77.04
		50%	76	00:00.86	78.28
		75%	38	00:00.45	72.22
4	LED-24	0%	200	00:02.66	54.00
		25%	150	00:02.03	48.02
		50%	100	00:01.36	50.67
		75%	50	00:00.75	44.61
5	LED-7	0%	200	00:01.08	58.00
		25%	150	00:00.86	58.44
		50%	100	00:00.68	58.67
		75%	50	00:00.41	54.06
6	Lymphography	0%	148	00:01.20	74.32
		25%	111	00:00.87	75.65
		50%	74	00:00.59	71.69
		75%	37	00:00.29	71.11
7	Monks-2	0%	432	00:02.64	51.39
		25%	324	00:01.95	50.82
		50%	216	00:01.31	50.31
		75%	108	00:00.63	55.25
8	Mushroom	0%	8124	00:29.00	100.00
		25%	6093	00:18.25	100.00
		50%	4062	00:11.94	99.98
		75%	2031	00:05.96	99.95
9	Promoter	0%	106	00:01.35	74.61
		25%	78	00:00.88	70.18
		50%	52	00:00.74	70.19

		75%	26	00:00.40	63.10
10	Soybean	0%	683	00:06.62	91.07
		25%	510	00:05.17	91.28
		50%	340	00:03.44	87.39
		75%	170	00:02.03	80.63
11	Thyroid	0%	3163	00:25.60	98.51
		25%	2370	00:18.12	98.58
		50%	1580	00:11.34	98.59
		75%	790	00:05.44	97.93
12	Waveform-40	0%	300	00:12.51	75.00
		25%	225	00:08.96	68.44
		50%	150	00:05.48	67.34
		75%	75	00:02.39	62.77

Table D10: Classification performance of FFNBP with reducing data.

Data Set #	Data Set	% of Data Reduced	Total # of Data	Time: User (h:mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	0%	699	0:04:20.00	94.99
		25%	522	0:02:25.39	93.99
		50%	348	0:00:55.80	95.11
		75%	174	0:00:11.79	92.73
2	Diabetes	0%	768	0:04:02.93	64.89
		25%	576	0:03:10.56	64.82
		50%	384	0:01:53.95	65.97
		75%	192	0:01:00.67	62.50
3	Hepatitis	0%	155	0:03:09.96	79.99
		25%	114	0:02:24.96	72.36
		50%	76	0:01:35.24	73.68
		75%	38	0:00:46.65	72.22
4	LED-24	0%	200	0:02:57.75	53.00
		25%	150	0:01:47.13	48.13
		50%	100	0:01:04.06	42.00
		75%	50	0:00:27.70	31.62
5	LED-7	0%	200	0:01:49.08	62.50
		25%	150	0:01:25.69	60.53
		50%	100	0:00:58.51	58.67
		75%	50	0:00:28.94	54.27
6	Lymphography	0%	148	0:00:37.78	79.73
		25%	111	0:00:17.55	78.09
		50%	74	0:00:06.67	79.72
		75%	37	0:00:03.21	79.08
7	Monks-2	0%	432	0:01:39.06	66.44
		25%	324	0:01:19.28	60.91
		50%	216	0:00:50.24	61.27
		75%	108	0:00:23.42	55.55
8	Mushroom	0%	8124	1:12:42.97	100.00
		25%	6093	0:56:32.78	99.99
		50%	4062	0:37:26.68	99.98
		75%	2031	0:38:26.54	99.84
9	Promoter	0%	106	0:00:06.71	72.80
		25%	78	0:00:06.19	68.58
		50%	52	0:00:03.38	66.03

		75%	26	0:00:01.68	70.83
10	Soybean	0%	683	1:04:06.22	93.41
		25%	510	0:40:05.37	92.61
		50%	340	0:19:15.27	91.07
		75%	170	0:05:42.58	83.72
11	Thyroid	0%	3163	1:44:50.69	98.23
		25%	2370	1:19:36.10	98.19
		50%	1580	0:52:24.39	98.16
		75%	790	0:26:43.95	98.48
12	Waveform-40	0%	300	0:00:42.40	88.67
		25%	225	0:00:29.56	84.73
		50%	150	0:00:11.69	84.21
		75%	75	0:00:05.47	76.59

Table D11: Classification performance of FuNe-I with reducing data.

Data Set #	Data Set	% of Data Reduced	Total # of Data	Time: User (h:mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	0%	699	0:44:48.93	94.56
		25%	522	0:32:06.18	93.93
		50%	348	0:22:04.14	93.97
		75%	174	0:11:43.94	92.70
2	Diabetes	0%	768	0:47:24.11	79.69
		25%	576	0:35:52.25	78.65
		50%	384	0:21:05.25	77.60
		75%	192	0:13:37.02	77.26
3	Hepatitis	0%	155	0:36:53.96	80.65
		25%	114	0:24:02.26	83.02
		50%	76	0:17:17.79	82.89
		75%	38	0:08:11.74	82.32
4	LED-24	0%	200	0:56:36.41	65.50
		25%	150	0:33:52.86	60.01
		50%	100	0:22:57.70	61.33
		75%	50	0:12:53.22	68.06
5	LED-7	0%	200	0:18:47.26	68.50
		25%	150	0:15:28.44	68.45
		50%	100	0:10:01.40	70.67
		75%	50	0:04:41.48	71.92
6	Lymphography	0%	148	0:28:24.94	78.38
		25%	111	0:23:15.54	78.95
		50%	74	0:16:15.64	79.35
		75%	37	0:06:54.36	81.85
7	Monks-2	0%	432	0:17:13.91	75.00
		25%	324	0:11:25.04	74.79
		50%	216	0:08:57.21	74.69
		75%	108	0:04:47.43	71.91
8	Mushroom	0%	8124	16:39:33.85	93.50
		25%	6093	12:29:38.97	86.77
		50%	4062	8:24:44.29	90.83
		75%	2031	4:16:45.08	88.10
9	Promoter	0%	106	1:10:42.53	78.30
		25%	78	0:49:52.88	79.05
		50%	52	0:33:24.05	82.05

		75%	26	0:17:45.77	87.50
10	Soybean	0%	683	5:40:14.87	88.97
		25%	510	4:16:19.24	84.57
		50%	340	2:54:09.05	86.76
		75%	170	1:25:42.65	76.23
11	Thyroid	0%	3163	8:02:45.58	90.76
		25%	2370	6:00:45.39	94.77
		50%	1580	3:55:26.75	94.62
		75%	790	2:01:54.37	94.64
12	Waveform-40	0%	300	1:17:07.78	78.00
		25%	225	1:16:53.41	80.00
		50%	150	0:51:14.19	78.19
		75%	75	0:27:30.22	75.46

Table D12: Classification performance of ID3 with dropping attributes.

Data Set #	Data Set	Total # of Attributes	% of Attributes Dropped	# of Attributes Dropped	Time: User (mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	9	0%	0	00:04.16	94.42
			10%	1	00:03.83	94.42
			25%	2	00:03.63	93.42
			50%	4	00:02.96	93.42
2	Diabetes	8	0%	0	00:09.55	69.01
			10%	1	00:09.23	69.27
			25%	2	00:08.08	69.27
			50%	4	00:07.10	69.53
3	Hepatitis	19	0%	0	00:01.78	77.41
			10%	2	00:01.59	80.01
			25%	5	00:01.48	80.01
			50%	10	00:01.10	78.74
4	LED-24	24	0%	0	00:02.66	54.00
			10%	2	00:02.76	38.00
			25%	4	00:02.79	53.00
			50%	9	00:02.33	43.00
5	LED-7	7	0%	0	00:01.08	58.00
			10%	1	00:00.91	58.00
			25%	2	00:00.81	53.00
			50%	4	00:00.47	43.00
6	Lymphography	18	0%	0	00:01.20	74.32
			10%	2	00:01.09	78.38
			25%	6	00:01.03	81.76
			50%	12	00:01.09	74.32
7	Monks-2	6	0%	0	00:02.64	51.39
			10%	1	00:01.80	54.86
			25%	2	00:01.40	52.55
			50%	3	00:00.84	64.32
8	Mushroom	22	0%	0	00:29.00	100.00
			10%	2	00:22.51	100.00
			25%	6	00:23.66	100.00
			50%	11	00:19.27	100.00
9	Promoter	57	0%	0	00:01.35	74.61
			10%	6	00:01.62	73.75
			25%	14	00:01.19	77.42

			50%	28	00:00.93	72.79
10	Soybean	35	0%	0	00:06.62	91.07
			10%	4	00:06.51	87.41
			25%	9	00:06.58	84.04
			50%	18	00:04.91	83.60
11	Thyroid	25	0%	0	00:25.60	98.51
			10%	2	00:21.80	98.64
			25%	6	00:20.30	98.67
			50%	12	00:15.54	95.23
12	Waveform-40	40	0%	0	00:12.51	75.00
			10%	4	00:11.41	77.00
			25%	10	00:10.06	66.00
			50%	20	00:07.27	65.23

Table D13: Classification performance of FFNBP with dropping attributes.

Data Set #	Data Set	Total # of Attributes	% of Attributes Dropped	# of Attributes Dropped	Time: User (h:mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	9	0%	0	0:04:20.00	94.99
			10%	1	0:04:03.87	94.42
			25%	2	0:03:25.68	94.70
			50%	4	0:02:22.20	95.69
2	Diabetes	8	0%	0	0:04:02.93	64.89
			10%	1	0:03:47.25	64.98
			25%	2	0:02:42.02	64.72
			50%	4	0:02:18.52	64.98
3	Hepatitis	19	0%	0	0:03:09.96	79.99
			10%	2	0:02:40.73	72.15
			25%	5	0:01:44.84	73.51
			50%	10	0:00:56.92	77.36
4	LED-24	24	0%	0	0:02:57.75	53.00
			10%	2	0:03:29.94	41.00
			25%	4	0:05:39.35	24.50
			50%	9	0:03:14.44	30.00
5	LED-7	7	0%	0	0:01:49.08	62.50
			10%	1	0:01:37.07	62.00
			25%	2	0:01:25.82	56.50
			50%	4	0:01:05.57	47.00
6	Lymphography	18	0%	0	0:00:37.78	79.73
			10%	2	0:00:37.48	78.38
			25%	6	0:01:35.62	76.35
			50%	12	0:01:09.80	75.47
7	Monks-2	6	0%	0	0:01:39.06	66.44
			10%	1	0:01:23.42	67.13
			25%	2	0:01:08.45	61.34
			50%	3	0:00:55.50	63.43
8	Mushroom	22	0%	0	1:12:42.97	100.00
			10%	2	1:31:05.76	100.00
			25%	6	1:57:10.26	94.37
			50%	11	1:08:34.08	99.80
9	Promoter	57	0%	0	0:00:06.71	72.80
			10%	6	0:00:06.37	75.55
			25%	14	0:00:09.22	69.02

			50%	28	0:00:09.74	56.66
10	Soybean	35	0%	0	1:04:06.22	93.41
			10%	4	0:54:07.49	86.24
			25%	9	0:43:14.40	82.28
			50%	18	0:24:40.71	83.73
11	Thyroid	25	0%	0	1:44:50.69	98.23
			10%	2	1:37:20.46	98.71
			25%	6	1:04:47.55	97.60
			50%	12	1:05:05.38	97.60
12	Waveform-40	40	0%	0	0:00:42.40	88.67
			10%	4	0:00:36.13	87.33
			25%	10	0:01:13.24	77.00
			50%	20	0:00:55.40	78.33

Table D14: Classification performance of FuNe-I with dropping attributes.

Data Set #	Data Set	Total # of Attributes	% of Attributes Dropped	# of Attributes Dropped	Time: User (h:mm:ss.00)	Testing Accuracy (%)
1	Breast cancer	9	0%	0	0:44:48.93	94.56
			10%	1	0:38:41.02	94.43
			25%	2	0:34:37.33	94.14
			50%	4	0:21:04.21	94.28
2	Diabetes	8	0%	0	0:47:24.11	79.69
			10%	1	0:40:47.76	78.65
			25%	2	0:29:43.03	78.52
			50%	4	0:16:08.60	77.73
3	Hepatitis	19	0%	0	0:36:53.96	80.65
			10%	2	0:28:46.67	78.63
			25%	5	0:16:44.45	84.45
			50%	10	0:10:17.82	85.77
4	LED-24	24	0%	0	0:56:36.41	65.50
			10%	2	0:39:52.15	55.50
			25%	4	0:29:19.47	47.50
			50%	9	0:17:12.21	45.00
5	LED-7	7	0%	0	0:18:47.26	68.50
			10%	1	0:15:56.64	65.00
			25%	2	0:12:31.16	62.00
			50%	4	0:06:30.07	62.00
6	Lymphography	18	0%	0	0:28:24.94	78.38
			10%	2	0:28:33.29	83.78
			25%	6	0:20:42.89	82.43
			50%	12	0:15:41.34	81.08
7	Monks-2	6	0%	0	0:17:13.91	75.00
			10%	1	0:13:59.87	75.00
			25%	2	0:09:27.16	74.54
			50%	3	0:06:39.63	74.54
8	Mushroom	22	0%	0	16:39:33.85	93.50
			10%	2	14:24:04.46	83.65
			25%	6	10:33:33.80	87.30
			50%	11	6:26:51.94	90.95
9	Promoter	57	0%	0	1:10:42.53	78.30
			10%	6	0:56:22.94	78.37
			25%	14	0:40:15.45	72.66

			50%	28	0:21:14.82	74.59
10	Soybean	35	0%	0	5:40:14.87	88.97
			10%	4	4:42:42.62	81.83
			25%	9	3:37:22.04	82.14
			50%	18	2:13:27.62	75.28
11	Thyroid	25	0%	0	8:02:45.58	90.76
			10%	2	7:17:44.37	95.23
			25%	6	5:24:08.13	95.83
			50%	12	3:29:23.46	96.71
12	Waveform-40	40	0%	0	1:17:07.78	78.00
			10%	4	1:29:41.62	88.33
			25%	10	1:05:54.21	79.33
			50%	20	0:34:04.32	89.67

APPENDIX E

TRADEMARK INFORMATION

Sparc Sparc is a registered trademark of Sun Microsystems

SunOS SunOS is a registered trademark of Sun Microsystems.

UNIX UNIX is a registered trademark of AT&T.

2

VITA

Khian Thong Lim

Candidate for Degree of

Master of Science

Thesis: MACHINE LEARNING ALGORITHMS AND FUZZY NEURAL NETWORKS: AN EXPERIMENTAL COMPARISION

Major Field: Computer Science

Biographical:

Personal Data: Born in Georgetown, Penang, Malaysia, On October 30, 1970, the youngest son of Mr. Lim Boon Kooi and Mrs. Teh Ah Hiang.

Education: Received Bachelor of Science in Computing and Information Science from Oklahoma State University, Stillwater, Oklahoma in May 1993. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University in July 1996.

Professional Experience: Employed by Department of Computing and Information Services at Oklahoma State University as a part time Support Specialist, from January 1994 to May 1996.