Tribhuvan University
Institute of Science and Technology

A Comparative Study of Rainfall Prediction Using Neural Network and Decision Tree

Dissertation
Submitted to

Central Department of Computer Science & Information Technology
Kirtipur, Kathmandu, Nepal

In partial fulfillment of the requirements
For the Master's Degree in Computer Science & Information Technology

By
Ramesh Shahi

Date: 2073-6-13

Supervisor
Asst. Prof. Sarbin Sayami

Tribhuvan University
Institute of Science and Technology

Central Department of Computer Science & Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

... ...
Mr. Ramesh Shahi

Date: 2073-6-13

Supervisor's Recommendation

I hereby recommend that this dissertation prepared under my supervision by Mr. Ramesh Shahi, entitled " A comparative study of Rainfall Prediction using Neural Network and Decision Tree" in partial fulfillment of the requirements for the degree of M.Sc. in Computer Science and Information Technology be processed for the evaluation.

... ...
Asst. Prof. Sarbin Sayami
Central Department of Computer Science and Information Technology,
Kirtipur, Nepal

Date: 2073-6-13

Tribhuvan University
Institute of Science and Technology

Central Department of Computer Science & Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Master's Degree in Computer Science and Information Technology.

Evaluation Committee

... ...
Asst.Prof Nawaraj Poudel
Central Department of Computer Science
And
Information Technology
Tribhuvan University
Kathmandu, Nepal
(Head)

....
Asst.Prof. Sarbin Sayami
Central Department of Computer Science
And
Information Technology
Tribhuvan University
Kathmandu, Nepal
(Supervisor)

... ...
(External Examiner)

. ...
(Internal Examiner)

Date: 2073-6-13

ACKNOWLEDGEMENTS

Above all, I thank God for his blessing and submit my graduate to my almighty for providing strength and confidence in me to complete this work. Secondly, I would like to extent my, gratitude and sincerest thanks to my respected Supervisor Asst.Prof.Sarbin Sayami, Central Department of Computer Science and Information Technology, for his impressive guidance, constructive criticism and intellectual support best owed for me sacrificing his invaluable time .

I would like to thank my respected teacher Asst. Prof. Nawaraj Poudel, Head of Central Department of Computer Science & IT. Kirtipur, TU (Kathmandu, Nepal) for his guidance and encouragement.

I would like to express my gratitude to respected teachers Prof. Dr. Shashidhar Ram Joshi, Prof.Dr. Subarna Shakya, Prof. Sudarshan Karanjeet, Mr. Min Bahadur Khati, Mr.Bishnu Gautam, Mr. Jagdish Bhatt, Mr. Bikash Balami, Mr. Dhiraj Pandey, Mr. Arjun Singh Saud , Mrs. Lalita Sthapit and others staffs of CDCSIT for granting me broad knowledge and inspirations within the time of period of two years.

Thanks to all my close friends for their supports. From the beginning to end in all count, I would like to thank my family members for their love, support and encouragement.

As we know that, there won't be 100% accuracy and efficiency in any work done by both machine any human, so there may be some errors in my project. But, I have done my best to complete this dissertation, so any suggestion regarding the mistakes of this work will be always welcomed.

Abstract

Weather forecasting is a vital application in meteorology and has been one of the most scientifically and technologically challenging problems around the world. In this paper, data mining technique was used for forecasting precipitation. This was carried out using Artificial Neural Network and Decision Tree algorithms and meteorological data collected between 2004 and 2008 from the city of Kathmandu, Nepal. A data model for the meteorological data was developed and this was used to train the two different algorithms. The performances of these algorithms were compared using standard performance metrics, and the algorithm which gave the best results used to generate classification rules for the mean weather variables. A predictive Neural Network model was also developed for the weather prediction program and the results compared with actual weather data for the predicted periods.

In order to build a model for Artificial Neural Network and Decision Tree. Among the total dataset , 80%,70 % and 75 % dataset, were used for training and 20%, 30 % and 35 % were used for the Testing.

Experimentation results show, feed-forward multilayer Perceptron based neural network classifier has lower error rate than Decision Tree. MLP classification system has the average system accuracy rate of 77.98%, system error rate of 22.02%, precision rate of 12.08%, and recall rate of 78.57%. Similarly, Decision Tree system has the average system accuracy rate of 73.74%, system error rate of 26.26%, and precision rate of 9.52% recall rate of 71.42%.

Keywords:

Preprocessing, Feature extraction, Artificial Neural Networks, Multilayer Perceptron, Decision Tree.

Table of Content

List of Figures

List of Tables

List of Abbreviation

ANN	Artificial Neural Network
KNN	K-Nearest Neighbor
NWP	Numerical Weather Prediction
AI	Artificial Intelligence
FFBP	Feed Forward Back Propagation
RMSE	Root Mean Square Error
DT	Decision Tree
GA	Genetic Algorithm
GRNN	Generalized Regression Neural Network
IDE	Integrated Development Environment

Chapter 1

1.1 Introduction

Rainfall prediction is estimation of future condition of rainfall. It is a state of atmosphere at given time in terms of weather variables like temp., humidity, wind speed, wind direction etc. It is a nonlinear and dynamic process. So; it varies day to day even minute to minute [2]. Scientists have tried to forecast meteorological characteristics using a number of methods, some of these methods being more accurate than others [5]. The important hydrological event rainfall is the quantity of water falling in drops from vapor condensed in the atmosphere. When water droplets in clouds become too heavy to stay in the air, they fall out towards the ground. Making reliable prediction about rainfall is very important in many areas of human activities. Rainfall supplies water for crops production. For example, crops plant use a huge amount of water and rains, Since Nepal is an agro-based country, accurate prediction of rainfall will be useful for proper planning of cultivation. That's why those who are involved in agriculture, they will be interested to know whether the next days (or months) will be rainy/non-rainy. Although water is vital to life, yet water can be extremely destructive. Thus, rainfall forecasting can warn of happening flood or drought so that peoples can save their lives and properties. Rainfall forecasting is also important for engineering applications, mainly for the design of hydroelectric power projects, because this system requires prior information about average rainfall, maximum/minimum rainfall for a year/each month. In urban areas, rainfall also has a strong influence on traffic control. Rainfall is one of the most important and challenging operational tasks carried out by the meteorological services all over the world. It is furthermore a complicated procedure that includes multiple specialized fields. The most widespread techniques used for rainfall prediction are the numerical and statistical methods [10]. Even thought researches in these fields are being conducted for a long time, successes of these models are rarely visible. Even though statistical model can predict accuracies in short term rainfall, it is difficult to predict the long term prediction of the rainfall due to nonlinear character of rainfall process. So, statistical method cannot generate good results [1].Although some of these models show notable accuracies in short term rainfall occurrence prediction, long term prediction and rainfall depth

prediction has proven to be somewhat difficult using traditional statistical methods. The reason for that is due to dependency of rainfall upon highly unpredictable physical parameters such as humidity, wind speed, wind direction, pressure, temperature, and cloud amount [1, 2]. There are many research work carried out for the rain fall prediction based on machine learning approaches and rule- based approaches. There are many learning approaches like linear regression, fuzzy logic, ANN, KNN for the rainfall prediction [16]. As climatic dataset is highly nonlinear, so; ANN can be used for rainfall prediction. ANN has matured to a great extent over a past few years. It provides methodology of solving highly non linear problems. As Inspired by brain, ANN is interconnection of highly non-linear neuron [2]. There have been a number of reported studies that have used ANNs, to solve problems in hydrology. For example, French et al. used an ANN to forecast rainfall with artificial inputs [20].

1.2 Motivation

In Data Mining data sets will be explored to yield hidden and unknown predictions which can be used in future for the efficient decision making. Now a day's companies use different techniques of Data Mining. Weather forecasting is one is the most of the challenging job. So using the data mining technique, it can predict the accurate result. As Nepal is the agricultural country and most of Nepali people depend on agriculture, weather forecasting plays vital role in agriculture. Research on weather forecasting may help directly or indirectly in agriculture.

1.3 Problem definition

Rainfall prediction has always been a challenging task. This is due to the fact that the data sets are highly nonlinear in nature. The accuracy of the prediction system is also highly dominated by the actual parameters chosen.

The inaccuracy of forecasting is due to the dynamic nature of the atmosphere, the computational power required to solve the equations that describe the atmosphere, the error involved in measuring the initial conditions, and an incomplete understanding of atmospheric processes. Therefore, forecasts become less accurate as the difference between current time and the time for which the forecast is being made increases. These

problems can only be addressed by the use of models based on Machine Learning which is also a challenging task.

1.4 Objective

The objectives of this research work are as follows:

1. To develop prediction model using Artificial Neural Network and Decision Tree.

2. To compare performance accuracy of these models using Precision, Recall and Average accuracy.

1.5 Outline of document

The remaining part of the document is organized as follows:

- Chapter 2 describes necessary background information and related work of rainfall forecasting.
- Chapter 3 describes detail system model and the theoretical approaches for rainfall prediction problem. It includes data normalization, Training and Testing Approaches.
- Chapter 4 describes the implementation details of the system. It includes description about the tools used and fundamental methods.
- Chapter 5 includes analysis and experimentation results about Performance Accuracy.
- Chapter 6 includes conclusion and future works.

Chapter 2

Literature review

Data mining was introduced in the 1990s and it is traced back along three categories i.e. classical statistics, artificial intelligence and machine learning. Data mining is the process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. It is also known as knowledge discovery i.e. detecting something new from large–scale or information processing [12]. Its objective is to extract knowledge or discovering of new information from large volumes of raw data for further use [24]. It is mainly related to database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post processing of discovered structures, visualization, and online updating. The data mining step might identify multiple groups in the data, which can then be used to obtain more accurate prediction results by a decision support system. A Decision Support System is a computer-based information system that supports business or organizational decision making activities [12]. It serves the management, operations, and planning levels of an organization and help to make management decisions, which may be rapidly changing and not easily specified in advance .Hence, the actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns such as groups of data records (cluster analysis i.e. grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups), unusual records (anomaly detection i.e. detection of outliers, noise, deviations or exceptions in large data sets) and dependencies (association rule mining i.e. detecting interesting relations between the variables in large databases).

2.1 Classification and prediction

Classification is the technique in which set of items are classified in the predefined category. Prediction is the process of predicting categorical class labels, constructing a model based on the training set.

Classification techniques are supervised learning techniques that classify data item into predefined class label. It is one of the most useful techniques in data mining to build classification models from an input data set. The used classification techniques commonly build models that are used to predict future data trends. There are several algorithms for data classification such as decision tree and Naïve Bayes classifiers. There are two main kinds of models in data mining which are as follow:

2.1.1 Predictive model: In this model, known data results are used to develop a model and that can be used to explicitly predict values. The purpose of Predictive model is mainly to predict the future outcome than current behavior. The prediction output can be numeric value or in categorized form. The predictive models are the supervised learning functions which predict the target value.

2.1.2 Descriptive model: In this model, patterns are described from existing data and models are abstract representation of reality which can be reflected to understand business and suggest actions. The second approach for mining data from large datasets is known as Descriptive data mining. It is normally used to generate correlation, frequency, etc. This Descriptive method can be defined as to discover regularities in the data and to uncover patterns. This is also used to find interesting subgroups in the bulk of data

2.2 Clustering

Clustering is a technique used to discover appropriate groupings of the elements for a set of data. It is undirected knowledge discovery or unsupervised learning i.e. there is no target field and relationship among the data is identified by bottom-up approach. A cluster is a subset of objects which are “similar”. A subset of objects such that the distance between any two objects in the cluster is less than the distance between any object in the cluster and any object not located inside it. It is a process of partitioning a set of data (or objects) into a set of meaningful sub-classes, called clusters. it help users understand the natural grouping or structure in a data set. it is unsupervised classification that means there is no predefined classes.

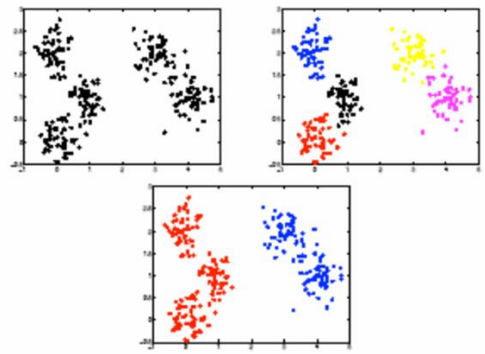

Fig: 2.1 Clustering of Objects

2.3 Association rule

In this technique, interesting association between attributes that are contained in a database is discovered which are based on the frequency counts of the number of items occur in the event (i.e. a combination of items), association rule tells if item X is a part of the event, then what is the percentage of item Y is also the part of event.

2.4 Supervised Learning

Supervised learning is the task of machine learning where the system can learn from the given available data. This type of learning is same as the human learning from the past experience to gain a new knowledge.

A data set used in the learning task consists of a set of data records, which are described by a set of attributes A = {A1, A2… $A_{|A|}$}, where |A| denotes the number of attributes or the size of the set A. The data set also has a special target attribute C, which is called the class attribute [13].

2.5 Unsupervised Learning

In unsupervised or undirected data mining however variable is singled out as the target as like the descriptive mining technique. The goal is rather to establish some relationship among all the variables in the data. The user asks the computer to identify patterns in the data that may be significant. Undirected modeling is used to explain those patters and relationships one they have been found.

2.6 Artificial neural networks

A neural network is a powerful data-modeling tool that is able to capture and represent complex input/output relationships [26]. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform intelligent tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following two ways:

- A neural network acquires knowledge through learning.
- A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

Physical nervous system is highly parallel, distributed information processing system having high degree of connectivity with capability of self learning. Human nervous system contains about 10 billion neurons with 60 trillions of interconnections. These connections are modified based on experience. Artificial neural network is non-linear, parallel, distributed, highly connected network having capability of adaptively, self-organization, fault tolerance etc which closely resembles with physical nervous system. Artificial neural networks are composed of interconnecting artificial neurons that mimic the properties of biological neurons which can be either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system. The real biological nervous system is highly complex. Artificial neural network algorithms attempt to abstract this complexity and focus on what may hypothetically matter most from an information processing point of view. Another incentive view is to reduce the amount of computation required to simulate artificial neural networks, so as to allow one to experiment with larger networks and train them on larger data sets.

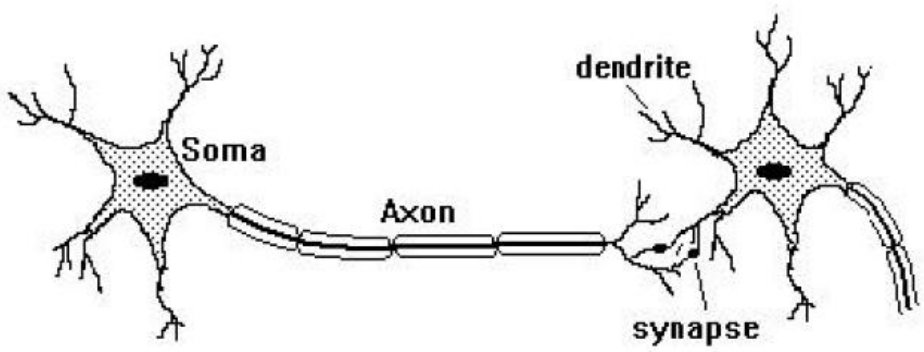

Fig: 2.2 Basic biological neuron.

Neural networks have seen an explosion of interest over the last few years and are being successfully applied in many problem domains like finance, medicine, engineering, geology and physics [31].

2.6.1 Feed forward network

Feed Forward neural network is such type of network where there is no directed cycle between the units. It is a simplest type of artificial neural network. The first layer is call input layer and information moves in only one direction and it is forwarded from input node to output node through hidden node if available. There are no cycles or loop in the network [4].

2.6.2 Multilayer perceptron

Multilayer perceptron is a feed forward neural network model which takes an input and gives the output. It consists of number layers of nodes which represent a perceptron and each layer is fully connected with another layer. Each node behaves as a neuron with non linear activation function. Multilayer perceptron use a supervised learning technique which is back-propagation algorithm for training the network [5].

2.6.3 Activation Function

Activation function is such type of function which is used for limiting the amplitude of the output of neuron. It is used to introduce the non-linearity in the hidden layer of the NN so that without this activation function NN would be similar to the single perceptions. In case of using linear function it wouldn't be a powerful as they are.

Activation function can be linear, threshold or sigmoid function. But here we use sigmoid function and is denoted by $F(x) = 1/1+e^{-x}$. [9].

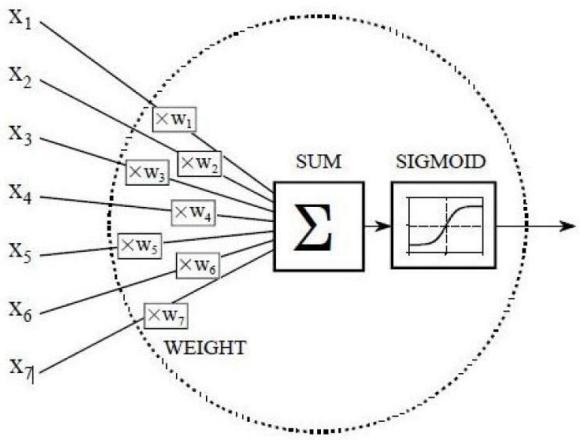

Fig: 2.3 Activation function

2.6.4 Back propagation algorithm

Back propagation is a form of supervised learning for multi-layer nets, also known as the generalized delta rule. It is a multilayer feed forward supervised network [7]. It provides an effective means of allowing a computer to examine data patterns that may be incomplete or noisy. In this learning algorithm, error data at the output layer is "back propagated" to earlier ones, allowing incoming weights to these layers to be updated. It is most often used as training algorithm in current neural network applications. The back propagation algorithm was developed by Paul Werbos in 1974 and rediscovered independently by Rumelhart and Parker [28]. Since its rediscovery, the back propagation algorithm has been widely used as a learning algorithm in feed forward multilayer neural networks. In general, the difficulty with multilayer Perceptron is calculating the weights of the hidden layers in an efficient way that resulting the least (or 27 zero) output error; it becomes more difficult if there are more hidden layers. To update the weights, one must calculate an error. At the output layer this error is measured; since error is the difference between the actual and desired (target) outputs. At the hidden layers, however, there is no direct observation of the error; hence, some other technique must be used. To calculate an error at the hidden layers that will cause minimization of the output error, as this is the ultimate goal [18]. The sum of all these numbers over all training examples is called the

total error of the network. If this number was zero, the network would be perfect, and the small the error, the better the network [2].

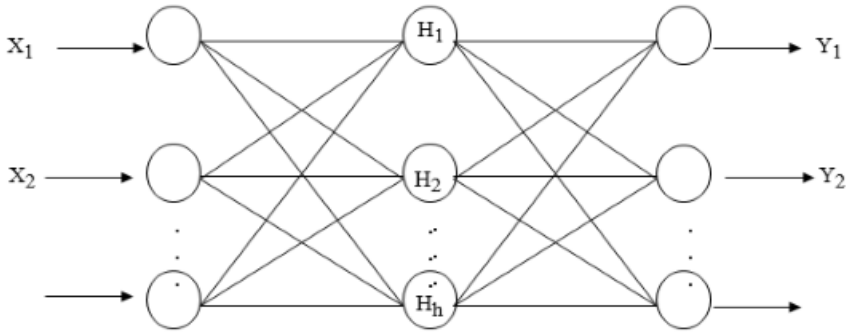

Fig: 2.4 Multilayer Back propagation Network

Let us assume, [X1, X2… Xn] be the input layer which can have more than one hidden layer. Let n1,n2, ……, nth derive unit for each neuron and target output H_1, H_2,. . . ,H_h, to be used as the input to derive the result for output layer and [Y_1,Y2, ……, Y_j] be the output layer and W_{ij} be weights. The nodes in the hidden layers organize themselves in a way that different nodes learn to recognize different features of the total input space. Initially, set up the network based on the problem domain and randomly generate weights Wij. Then feed a training set, [X1, X2 …Xn], into BPN in order to compute the weighted sum and apply the transfer function on each node in each layer. Feeding the transferred data to the next layer until the output layer is reached. The output pattern is compared to the desired output and an error is computed for each unit. Feedback error is back to each node in the hidden layer. Each unit in hidden layer receives only a portion of total errors and these errors then feedback to the input layer, until the error is very small.

2.6.5 Algorithm Details

2.6.5.1 Inputs

D: Dataset consisting of the training tuples and their associated target values.

l: learning rate

2.6.5.2Outputs

A trained neural network.

2.6.5.3 Methods

Initialize all weights in network.

while terminating condition is not satisfied {

for each training tuple X in D{

// Propagate the inputs forward.

for each input layer unit j {

$O_j = I_{j;}$ // output of an input unit is its actual input value.

for each hidden or output layer unit j {

$I_j = \sum_i w_{ij} O_i$ // compute the net input of unit j with respect to the previous layer, i

$O_j = 1/1+e^{-Ij}\}$ // compute the output of each unit j

// Backpropagate the errors

for each unit j in the output layer

$Err_{j =} O_j(1-O_j)\ (T_j – O_j)$ // compute the error

for each unit j in the hidden layers from the last to the first hidden layer

$Err_{j =} O_j(1-O_j) \sum_k Err_k w_{jk}$ // compute the error with respect to the next higher layer, k

for each weight w_{ij} in network {

$\Delta w_{ij} = (l)\ Err_j O_i$//weight increment

$w_{ij} = w_{ij} + \Delta w_{ij}\}$ //weight update

}

2.7 Decision Tree

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target parameter based on several input parameter. A tree can be made to learn by splitting the source data set into subsets based on an attribute value test [24]. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has all the same value of the target variable or when splitting no longer adds value to the predictions. Decision trees can be described as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data to facilitate the machine learning.

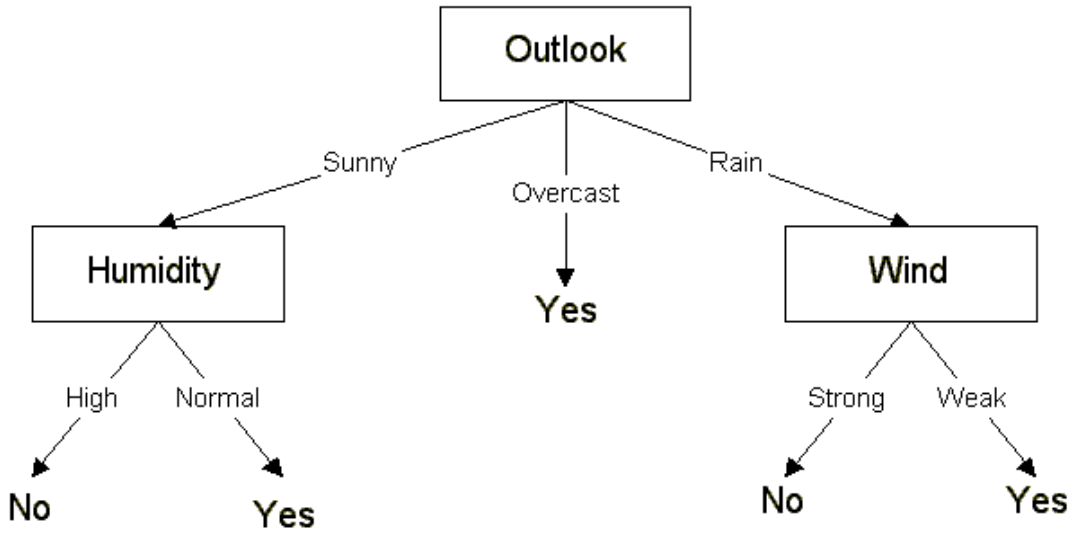

Fig: 2.5 Decision tree.

2.7.1 ID3 Algorithm

ID3 searches through the attributes of the training instances and extracts the attribute that best separates the given examples. If the attribute perfectly classifies the training sets then ID3 stops; otherwise it recursively operates on the n (where n = number of possible values of an attribute) partitioned subsets to get their "best" attribute. The algorithm uses a greedy search, that is, it picks the best attribute and never looks back to reconsider earlier choices. It derives its classes from a fixed set of training instances.

2.7.2 Attribute Selection

For the selection of attribute, it uses A statistical property, called information gain, is used. Gain measures how well a given attribute separates training examples into targeted classes. The one with the highest information (information being the most useful for classification) is selected. In order to define gain, we first borrow an idea from information theory called entropy. Entropy measures the amount of information in an attribute.

2.7.2.1 Entropy

It measures homogeneity of a node and it is denoted by formula

Entropy(S) = $\sum -p_i \log_2(p_i)$

Where p_i is the proportion of S belonging to class i. S is entire sample set.

2.7.2.2 Information gain

$$Gain(S, A) = Entropy(S) - \sum_{V \in value\ A} |S_v| / |S| * Entropy(S_v)$$

Where,

S is each value v of all possible values of attribute A

S_v = subset of S for which attribute A has value v

$|S_v|$ = number of elements in S_v

|S| = number of elements in S

These Entropy and information gain formula is used for the generation the rule for the rain fall prediction [22].

2.8. Previous work

True quantitative rainfall forecasting is generally difficult and also a challenging task for anyone Because of our complex atmospheric processes. Thus, rainfall is treated as one of the most complex and difficult events among other hydrological events.

In past there were number of models designed using NWP. In 2008, FRNN have proposed for solving forecasting problems [1]. In 2009, it has been compared common meteorological forecasting method with ANN and he found the performance of ANN with high accuracy [2]. Nekoosa, 2010, have used radial basis function neural network for financial time-series forecasting, and the result of their experiment shows the feasibility and effectiveness.[4]. Geetha and Selvaraj, 2011, have predicted Rainfall in Chennai using back propagation neural network model, by their research the mean monthly rainfall is predicted using ANN model. The model can perform well both in training and independent periods [5]. There are different techniques for weather forecasting namely statistical method and AI methods (GA method, fuzzy inference system , ANN , decision tree etc..) that statistical techniques have been developed for many years but have been proven to be incapable to handle the non-linear series. We believe rainfall data are multi-dimensional, non-linear and dynamic, therefore to search for an appropriate model, the powerful techniques ANN, decision tree and GA have been

chosen. Results obtained by these models are also compared to the linear multiple regression model to show selected AI models performances.

There are several researches on rainfall prediction using a neural network some of which are given below. Using the ANN for daily rainfall modeling Rajurkar have found that ANN with a multiple input and single output model predicted the rainfall value with high accuracy during the training and testing period [25].

Tsong Lin Lee 2004 has predicted long term rainfall prediction using back propagation neural network with efficient result. A contribution of Gwo-Fong Lin and Lu-Hsein Chen 2005 has used neural network with two hidden layer to forecast the rainfall and it has been observed that result was reasonable forecast [18].

Most of the researchers have been using ANN for various annual predictions like rainfall, tide, temperature etc. NiraveshSrikalra and ChularatTanpraserthave used ANN for daily rainfall prediction in Chao Phraya River with Online Data Collection, and they found that it is possible to predict rainfall on daily basis with acceptably accuracy using Artificial Neural Network [29].

A.D. Kumarasiri and D.U.J. Sonnadara, 2006, have applied an innovative technique for rainfall forecasting using Artificial Neural Networks based on feed-forward back-propagation architecture. Three Neural Network models were developed; a one-day-ahead model for predicting the rainfall occurrence of the next day, which was able to make predictions with a 74.25% accuracy, and two long term forecasting models for monthly and yearly rainfall depth predictions with 58.33% and 76.67% accuracies within a 5% uncertainty level [17].

In 2007, ANN is used in a new experiment of short term rainfall forecasting and he found that MLP network has the minimum forecasting error and can be considered as a good method to model the STRF systems [11].

Chattopadhyay, 2008, have worked out to find out best hidden layer size for three layered neural net in predicting monsoon rainfall in India, and they have found that eleven-hidden-nodes three-layered neural network has more reasonable result in forecasting task [3].

Mar, and Naing, 2008, have tested more over 100 cases by changing the number of input and hidden nodes from 1 to 10 nodes, respectively, and only one output node in an optimum artificial neural network architecture and they concluded that 3 inputs-10 hiddens-1 output architecture model gives the best prediction result for monthly precipitation prediction [32].

On the basis of humidity, dew point and pressure in India, Enireddy, 2010, have used the back propagation neural network model for predicting the rainfall. In the training they have obtained 99.79% of accuracy and 94.28% in testing. From these results they have concluded that rainfall can predicted in future using the same method [30].

In 2011 researcher, have put in a review report on Rainfall-Runoff modeling using ANN, in the same study they have reviewed three neural network methods, FFBP, RBF and Generalized Regression Neural Network and they have seen that GRNN flow estimation performances were close to those of the FFBP, RBF and MLP [14].

Rainfall forecasting in a mountainous region is a big task in itself in Iran. In 2011, Mekanik have tried to forecast rainfall using ANN model. And feed forward ANN rainfall model was developed to investigate its potentials in forecasting rainfall. A monthly feed forward multi layer perceptron [19].

In 2011, Rainfall was predicted in Chennai using back propagation neural network model, by their research the mean monthly rainfall is predicted using ANN model. The model can perform well both in training and independent periods [8].

Different researcher has research on the decision tree for rain fall prediction. Here it has mentioned the some research on the decision tree in the context of the classification of the rain fall.

Prasad proposed to employ Supervised Learning decision tree using Gini index for the prediction of the precipitation which resulted in an accuracy of 72.3% [21].
E. G. Petre [6] presented a small application of CART decision tree algorithm for weather prediction. The data collected is registered over Hong Kong. The data is recorded between 2002 and 2005. The data used for creating the dataset includes parameters year,

month, average pressure, relative humidity, clouds quantity, precipitation and average temperature. The decision tree, results and statistical information about the data are used to generate the decision model for prediction of weather.

F. Oliya and A. B. Adeyemo [15] investigated the use of data mining techniques in predicting maximum temperature, rainfall, evaporation and wind speed. C4.5, ID3 decision tree algorithms and artificial neural networks are used for prediction. The meteorological data is collected between 2000 and 2009 from the city Ibadan, Nigeria. A data model for the meteorological data is developed and is used to train the classifier algorithms. The performance of each algorithm is compared with the standard performance metrics and the algorithm with the best result is used to generate classification rules for the mean weather variables.

Data mining methods was implemented for guiding the path of the ships during sailing. Global Positioning System is used for identifying the area in which the ship is currently navigating. The attributes of weather data includes climate, humidity, temperature, stormy [23]. The weather report of the area traced is compared with the existing database. The analyzed dataset is provided to the decision tree algorithm, C4.5 and ID3. The decision obtained regarding the weather condition is instructed to the ship and the path is chosen accordingly.

Soo-YeonJi [28] predicted the hourly rainfall in any geographical regions. Rainfall, the hourly rainfall prediction is performed. CART and C4.5, ID3 are used to provide outcomes, which may provide hidden and important patterns with transparent reasons. Result obtained from both algorithms was satisfactory.

S. Kannan and S. Ghosh [27] contributed towards developing methodology for predicting state of rainfall at local or regional scale for a river basin from large scale climatological data. A model based on decision tree algorithm, CART, ID3, is used for the generation of rainfall states from large scale atmospheric variables in a river basin.

Although there is a lots of AI model for prediction of rainfall but there is no comparative study of decision tree algorithm (ID3) and neural network algorithm (back propagation) for rainfall prediction. So in this thesis we are doing the comparative study of these two AI models for the rainfall prediction of Nepal at airport location.

Chapter 3

Research Methodology

The Top level rainfall prediction system as shown in Fig 3.1 is divided into four sub-systems, data acquisition, Data selection, data preprocessing and data transformation. Each stages of this theoretical model are briefly described in this section. Detail of each subsystem is given in later sections.

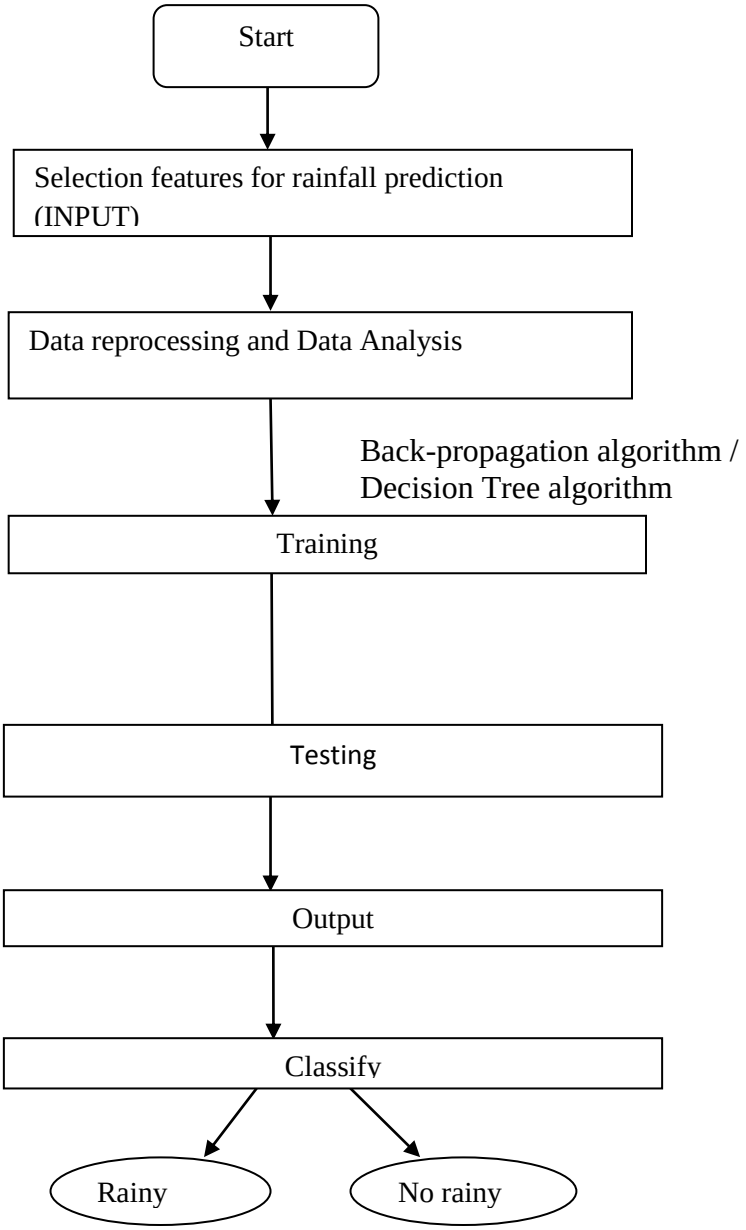

Fig: 3.1 Flowchart for rain fall prediction

In this thesis we are using two different algorithms like back-propagation and ID3 algorithm. So there is different process for prediction of rainfall using ANN and decision tree.

3.1 Process for Prediction

3.1.1 Data Collection

Dataset for rain fall prediction is collected from the Department of meteorology and Hydrology of Nepal. Different types of parameters were collected from the Kathmandu valley satiation" Tribhuvan International Airport "from the year 2004 to 2008 A.D. All the data is collected of daily base.

3.1.2Data Selection

At this stage, data relevant to the analysis was decided and retrieved from the dataset. The meteorological dataset has five attributes which are wind speed, humidity precipitation, minimum temperature, maximum temperature, their type and description is presented in Table 3.1.

Table: 3.1 sample data set with 5 parameters.

Wind Speed	Max. Temp	Min. Temp	Humidity	Precipitation
47.3	18.8	2.4	96.2	21.8
18.3	19	1.1	96.9	9.5
1.4	18.3	1.7	96.9	16.3
0.7	18.3	1	100	47.3
17.6	19.5	1.3	96.7	18.2
0.2	20.9	2	96.7	1.4
18.3	20	2	94.4	0.7
1.6	20.3	2	97	17.6

3.1.3 Data preprocessing

The obtained input and the output data have to be normalized because they are of different units and otherwise there will be no correlation between input and output values.

First the mean of all the data separately was taken for humidity, wind speed, rainfall, and temperature.

Let M be the mean.

M=sum of all entries/number of entries.

Then the standard deviations, SD, for each of these parameters were calculated individually. Now after having the values of mean and SD for every parameter, the values for each parameters were normalized by using

Normalized value = (x-M)/SD [16].

3.1.4 Data Transformation

This is also known as data consolidation. It is the stage in which the selected data is transformed into forms appropriate for data mining. The data file was saved in CVS file format

3.2 Formation of Decision Tree

To build a tree, it uses a humidity, minimum temperature, maximum temperature and wind speed as a attributes. As mentioned above decision tree generate some rules and based on the generated rules, it classifies the values. To build a tree, it selects the attribute as a root which has the highest value of information gain. The information gain is calculated on the basis of entropy. Here tree is generated having humidity as root, temperature and wind are the sub tree as shown in Fig 3.2.

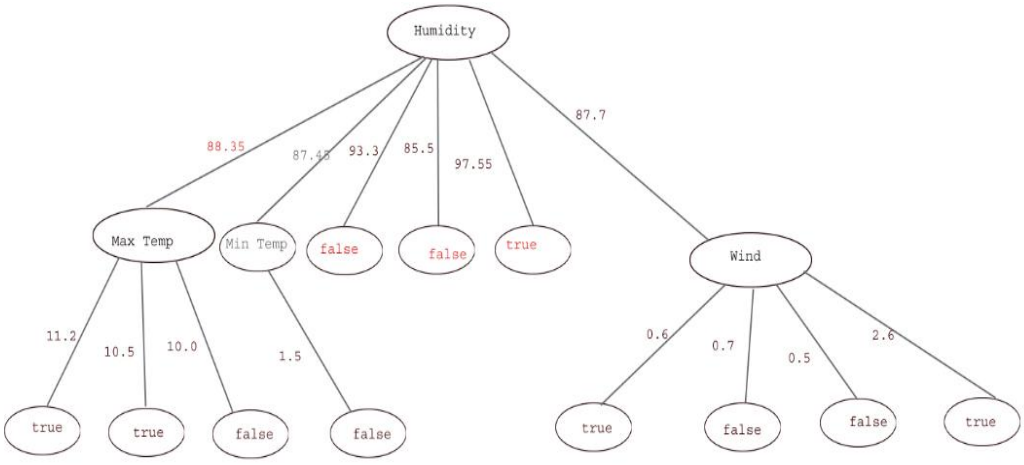

Fig: 3.2 Decision tree for rainfall prediction

3.3 Formation of NN architecture

For the neural network architecture, it uses the same parameters like humidity, minimum temperature, maximum temperature, wind speed and precipitation as in decision tree. It uses four parameter as input. These parameters are humidity, minimum temperature, maximum temperature and wind speed. But it is known to that entire hidden layer and number of node in hidden layer cannot be fixed. So here, we have found three hidden layer and each layer contains three, two and three node respectively. Four inputs are forwarded to the network and input values and weight is multiplied and sum is calculated. This calculated result is passed towards the sigmoid function and obtained result is further passed to the hidden layer. Finally it has one output layer and RMSE is set 20 that means our network only tolerate the 20% of the error. If the error obtained from the output layer is beyond the RMSE value, this obtained error is back-propagated to the network. This process is continued until final output is within the range of 20 of RMSE. To find the RMSE value it uses different iteration that is epoch. If error is not within the range of RMSE value and epoch is finished then again network is re-initialized with random weight and process is continued. Here final network structure is show given below

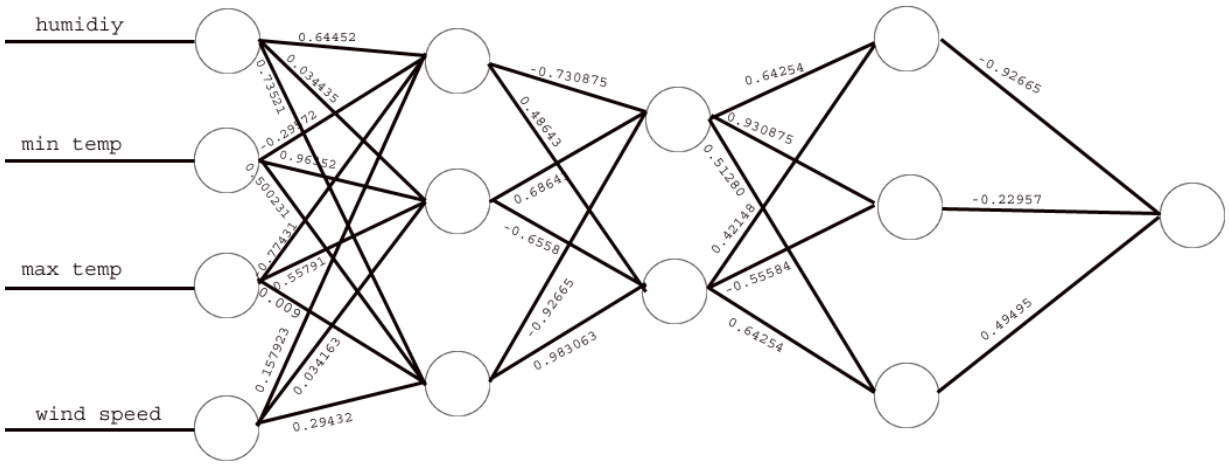

Fig: 3.3 Neural network structures

3.4 System Evaluation Measures

The correctness of a classification can be evaluated by computing the number of correctly recognized class examples (true positives), the number of correctly recognized examples that do not belong to the class (true negatives), examples that either were incorrectly assigned to the class (false positives) and examples that were not recognized as class examples (false negatives).

Measures for multi-class classification based on a generalization of the measures of binary classification for many classes C_i are given below. Where, tp_i represent true positive for class C_i, fp_i represent false positive for class C_i, fn_i represent false negative for class C_i, tn_i, representtrue negative for class C_i.

3.4.1 Average System Accuracy

Average system accuracy evaluates the average per-class effectiveness of a classification system.

Average Accuracy=$\sum_{i=1}^{l} \frac{tpi+tni}{tpi+tni+fpi+fni}$

3.4.2 System Error

System error is the average per-class classification error of the system.

$$\text{Error rate} = \sum_{i=1}^{l} \frac{fpi+fni}{tpi+tni+fpi+fni}$$

3.4.3 Precision

Precision (also called positive predictive value) is the number of correctly classified positive examples divided by the number of examples labeled by the system as positive. Micro precision is the agreement of the data class labels with those of classifiers calculated from sums of per-test decisions.

$$\text{Precision} = \frac{\sum_{i=1}^{l} tpi}{\sum_{i=1}^{l} tpi+fpi}$$

3.4.4 Recall

Recall is the effectiveness of a classifier to identify class labels if calculated from sums of per-test decisions.

$$\text{Recall} = \frac{\sum_{i=1}^{l} tpi}{\sum_{i=1}^{l} tpi + fni}$$

Chapter 4

Implementation tools and techniques

All the algorithms of purposed prediction system are implemented in Microsoft Visual Studio 10 with .net framework 3.0 versions. Visual studio is installed on an Intel(R) Core(TM) i5 CPU M 480 @ 2.67 GHz, 2.67 GHz processor. The Computer has total main memory of 4 Gigabyte and 64-bit Operating system, x64-based processor and Microsoft Windows7 ultimate operating system installed in it.

4.1 Dot net framework

The .NET Framework is a technology that supports building and running the next generation of applications and XML Web services. The .NET Framework consists of the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. We can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, while also enforcing strict type safety and other forms of code accuracy that promote robustness. In fact, the concept of code management is a fundamental principle of the runtime. The class library is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

4.2 Programming Language and IDE

4.2.1 C# programming Language

C# is type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. We can use C# to create Windows client applications, web application and other various applications. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time.

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure, an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, when the C# program is executed, the assembly is loaded into the CLR, Then, and the CLR performs Just In Time compilation to convert the IL code to native machine instructions.

4.2.2 Visual Studio IDE

Microsoft Visual Studio is an IDE from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services

Visual Studio includes a code editor supporting IntelliSense (the code completion component). It support integrated debugger and Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhances the functionality at almost every level—including adding support for source-control systems (like Subversion). Visual Studio supports different programming languages and allows the code editor and debugger It support visual C#, VB.net (visual basic.net),F#,J#. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. The Visual Studio 2010 IDE was redesigned which, according to Microsoft, clears the UI organization and "reduces clutter and complexity. The new IDE better supports multiple document windows The IDE shell has been rewritten using the Windows Presentation

Foundation .The new multi-paradigm ML-variant F# forms part of Visual Studio 2010. Visual Studio 2010 comes with .NET Framework 4 and support developing applications targeting Windows. It supports IBM DB2 and Oracle databases, in addition to Microsoft SQL Server. It has integrated support for developing Microsoft Silverlight applications; including an interactive designer. Visual Studio 2010 offers several tools to make parallel programming simpler: Visual Studio 2010 includes tools for debugging. The new tools allow the visualization of parallel Tasks and their runtime stacks. Tools for profiling parallel applications can be used for visualization of thread wait-times and thread migrations across processor cores.

The Visual Studio 2010 code editor now highlights references; whenever a symbol is selected; all other usages of the symbol are highlighted. It also offers a Quick Search feature to incrementally search across all symbols in C++, C# and VB.NET projects. Quick Search supports substring matches and camel Case searches. The Call Hierarchy feature allows the developer to see all the methods that are called from a current method as well as the methods that call the current one. Visual Studio supports a consume-first mode which developers can opt into. In this mode, IntelliSense does not auto-complete identifiers; this allows the developer to use undefined identifiers (like variable or method names) and define those later. Visual Studio 2010 can also help in this by automatically defining them, if it can infer their types from usage. Current versions of Visual Studio have a known bug which makes IntelliSense unusable for projects using pure C (not C++).

4.2.3 Methods Used in Implementation

4.2.3.1 Random Number generator

For the training of neural network it uses random weight assignment. For this Random class is used to generate random number between 0 and 1. This random function is

```
Random rand = new Random ();
rand.NextDouble ()
```

4.2.3.2 Training Function for NN

In the implementation of NN, it uses Train ( ) function for the training of the NN and error is calculated within this function.

```
void Train( )
{
}
```

4.2.3.3 Adjusting Weights Function for NN Training.

Within Train ( ) function another function AdjustWeights ( ) is revoked. This function is used to adjust the weights between the network layers when expected error is not obtained as output.

```
void AdjustWeights (double delta)
{       }
```

4.2.3.4 Testing Function for NN

After completion of training, model has to test with testing data. In order to implementation for testing model, Test ( ) function is used for testing the system as given below.

```
void Test ( )
{
}
```

4.2.3.5 Building Records for DT

This is a function where it is used to read data from file and create a table for the calculation of entropy and information gain.

```
List<Record> buildRecords (int num)
{

}
```

4.2.3.6 Entropy calculation Function for DT

This function is used for the calculation entropy in order to the calculation of Information Gain.

When entropy is zero at that time decision is made either it is rain or no rain.

```
double calculateEntropy (List<Record> data)
{
        //statement goes here.
}
```

4.2.3.7 Information Gain Function for DT

This function is used for the finding the Information gain based on the entropy calculated.

Attribute having the highest information gain is selected as a root node of the tree.

```
double calculateGain (double rootEntropy, List<Double> subEntropies, List<int>
setSizes, int data)
{

}
```

The detail definition of methods is included in Appendix B.

Chapter 5

Experiments and Results

Rainfall prediction system is experimented by creating one training data set and one testing dataset. This chapter describes the datasets used in experiment and empirical results. Training and testing dataset are described below.

5.1 Cross fold Validation

During the training of NN and DT different datasets were chosen for the training and testing. It has chosen 70% for training and 30% for testing, 80 % for training and 20% for testing and 75% for training and 35 % for testing. Testing result for both NN and Decision tree is given below.

5.2Training Dataset for NN

As shown in Table 4.1 dataset comprises of attributes like Minimum Temperature, Maximum Temperature, Humidity, Wind Speed and Precipitation. This data set is used for training ANN and DT models respectively. Further dataset are given in appendix A.

Table: 5.1 Sample dataset for training

Wind Speed	Max. Temp	Min. Temp	Humidity	Precipitation
47.3	18.8	2.4	96.2	21.8
18.3	19	1.1	96.9	9.5
1.4	18.3	1.7	96.9	16.3
0.7	18.3	1	100	47.3
17.6	19.5	1.3	96.7	18.2
0.2	20.9	2	96.7	1.4
18.3	20	2	94.4	0.7
1.6	20.3	2	97	17.6

5.3 Training of ANN

We have collected dataset from the department of meteorology and Hydrology of Nepal. Here we use different input parameters like wind speed, minimum temperature, maximum temperature, humidity and precipitation of the Kathmandu valley satiation "Tribhuvan International Airport" from the year 2004 to 2008 A.D. All the data is collected of daily base. 80%,70 % and 75 % of total data set were used for the training of neural network, all the parameter is feed to the structure and it is found that network structure has 3 hidden layers where first hidden layer contains 3 nodes, second hidden layer contain 2 nodes and third hidden layer contains 3 nodes. The epoch is set 2500 and RMSE is set 20.when the epoch is greater than 2500 then network again reinitialize. The sum of product of input values and weight is calculated as described by algorithm. If error produce by network structure is satisfied with RMSE then training is stopped.

We have set weight from range 0 to 1 before training as shown in figure 5.1. Here It is shown how the back propagation algorithm work. Calculation for iteration first is given below to calculate the error of output layer.

First hidden layer

18.8 * 0.001 + 2.4*0.004 + 96.2*0.007+ 21.8*0.10 = 2.8818

18.8 * 0.002 + 2.4*0.005 + 96.2*0.008+ 21.8*0.11 = 3.2172

18.8 * 0.003 + 2.4*0.006 + 96.2*0.009+ 21.8*0.12=3.5526

Second hidden layer

2.8818 * 0.13 + 3.2172*0.15 + 3.5526 * 0.17 =1.461156

2.8818 * 0.14 + 3.2172*0.16 + 3.5526 * 0.18 = 1.557672

Third hidden layer

1.461156 * 0.19 + 1.557672*0.22 =0.62030748

1.461156 * 0.20 + 1.557672*0.23 =0.65049576

1.461156 * 0.21 + 1.557672*0.24 =0.68068404

Output Layer

0.62030748*0.25 + 0.65049576 * 0.26 + 0.68068404 * 0.27 =0.5079904584

Using sigmoid function on output

We have sigmoid function

$F(x) = 1/1 + e^{-x}$

$F($0.5079904584**$) = 1/1+e^{-0.5079904584} =$ 0.62433527**

here our target error is 0.2 i. e 20 % error tolerate but in the output layer it is found that 0.62433527 error . So this error is propagated backward to adjust the weight between each layer so that we can get minimum error which meet out target error. In this same, this calculation process is repeated for other iteration and it will stop train when it meet targeted error defined

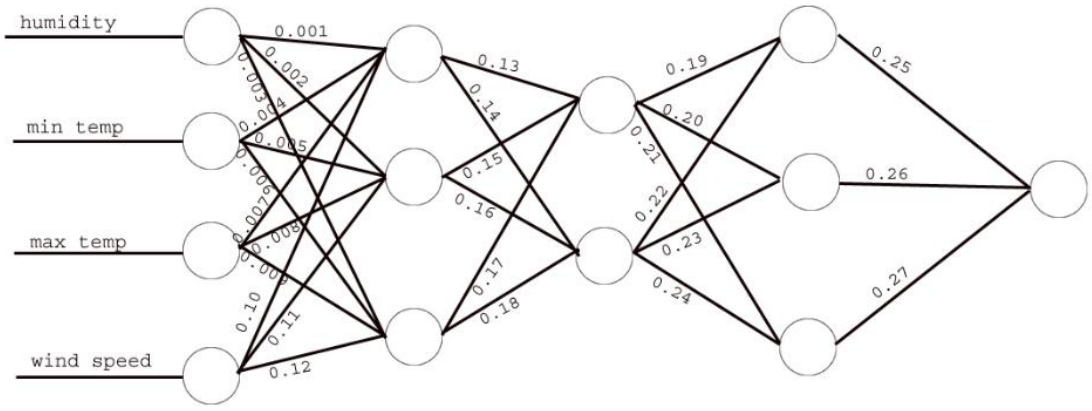

Fig: 5.1 Initialization of weight set before training.

5.4 Training of Decision tree

We have collected dataset from the department of meteorology and Hydrology of Nepal. Here we use different input parameters like wind speed, min temperature, max temperature, humidity and precipitation of the Kathmandu valley satiation "Tribhuvan International Airport" from the year 2004 to 2008 A.D. All the data is collected of daily base. 80%, 70 % and 75 % of total data set is used for the training of decision tree. In

order to train in decision tree, we calculate the entropy and information gain. Information gain is used to select the root node.

Here it is shown for calculation of entropy and information gain for selection of root node.

Gain(S, Temperature) = Entropy(S) - 5/1450*Entropy (11.2) -1/1450*Entropy (10.05) - 2/1450*Entropy (10) -

2/1450*Entropy(9.65) - 2/1450*Entropy(9.55) - 4/1450*Entropy(10.75) - 5/1450*Entropy(11.45) - 4/1450*Entropy(11) -3/1450*Entropy(10.65) -2/1450*Entropy(10.9) -2/1450*Entropy(10.5) - 3/1450*Entropy(11.4) -5/1450*Entropy(10.95) -6/1450*Entropy(12.75) -2/1450*Entropy(11.65) -

Gain(S, Temperature) = 0.36101517026802227

Gain(S, Humidity) = Entropy(S) - 2/1450*Entropy(88.35) -4/1450*Entropy(87.7) - 1/1450*Entropy(93.3) -5/1450*Entropy(89.65) -5/1450*Entropy(91.8) - 5/1450*Entropy(80.05) -3/1450*Entropy(84.75) -7/1450*Entropy(79.65) - 6/1450*Entropy(80.75) -7/1450*Entropy(77.15) -5/1450*Entropy(84.1) - 3/1450*Entropy(81) -1/1450*Entropy(85.5) -5/1450*Entropy(82.2) - 2/1450*Entropy(82.85) -3/1450*Entropy(86.45) -5/1450*Entropy(83.1) - 1/1450*Entropy(84.9) -2/1450*Entropy(84.65) -2/1450*Entropy(93.75) - 1/1450*Entropy(88.95)

Gain(S, Humidity) = 0.46861828366535757

Gain(S, Wind) = Entropy(S) - 270/1450*Entropy(0.6) -139/1450*Entropy(0.7) - 117/1450*Entropy(0.5) -110/1450*Entropy(0.8) -102/1450*Entropy(0.9) - 92/1450*Entropy(0.4) 91/1450*Entropy(1) -32/1450*Entropy(0.1) - 54/1450*Entropy(1.3) -59/1450*Entropy(1.2) -89/1450*Entropy(1.1) - 27/1450*Entropy(1.7) -9/1450*Entropy(1.9) -36/1450*Entropy(1.5) - 39/1450*Entropy(1.4) -1/1450*Entropy(3.3) -67/1450*Entropy(0.3) - 41/1450*Entropy(0.2) -18/1450*Entropy(1.6) -7/1450*Entropy(2.2) - 15/1450*Entropy(1.8) -9/1450*Entropy(2) -10/1450*Entropy(2.1) -1/1450*Entropy(2.5) -2/1450*Entropy(2.4) -10/1450*Entropy(0) -1/1450*Entropy(2.6) -1/1450*Entropy(4.1) -1/1450*Entropy(2.8)

Gain(S, Wind) = 0.09551649719733106

Here we have humidity has highest value of information gain so we choose the humidity as root node. Further we calculate the information gain with the help of entropy for

different values of humidity. And we select temperature as second root node as shown in figure. In this similar way the entire tree is created.

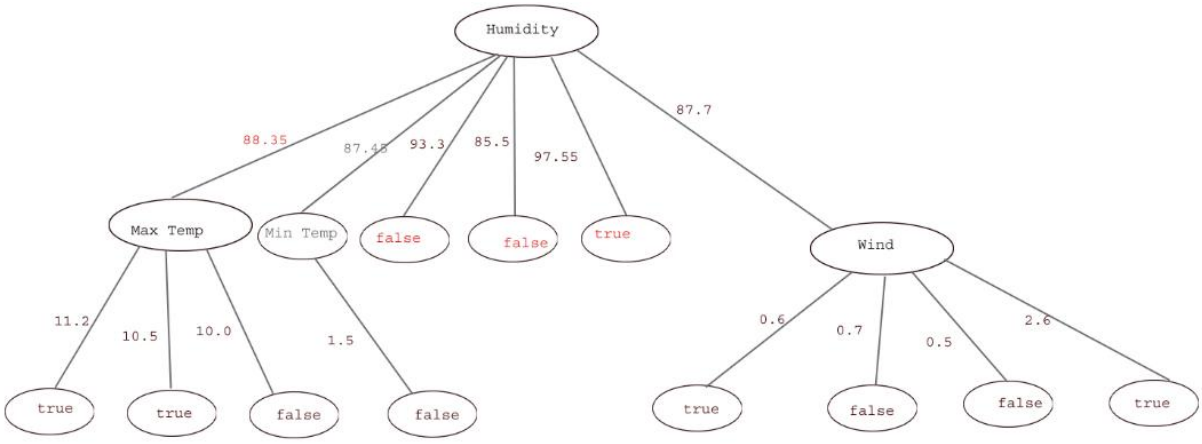

Fig: 5.2 Final decision tree.

5.5 Sample of Rule Generated by Decision tree

After the training for decision tree using the above sample dataset, it generates a certain rule as shown in Table 5.1.

Table: 5.2 Rule generated by DT

Humidity	**Min Temp**	**Max Temp**	**Wind**	**Rain fall**
88.35		11.2		True
88.35		10.5		True
88.35		10.0		False
93.3				False
85.5				False
97.55				True
87.7			0.6	True
87.7			0.7	False
87.7			0.5	False
87.6			2.6	True
87.45	1.5			False

5.6 Testing Dataset

In order to test decision tree model, 20%, 30% and 35% dataset of the total datasets were separated. Table 5.2 shows the sample dataset for testing the decision tree.

Table: 5.3 Sample dataset for testing.

Wind Speed	Max. Temp	Min. Temp	Humidity	Precipitation
47.3	18.8	2.4	98	21.8
18.3	19	1.1	96.9	9.5
1.4	18.3	1.7	96.9	16.3
0.7	18.3	1	100	47.3
17.6	19.5	1.3	96.7	18.2
0.2	20.9	2	96.7	1.4
18.3	20	2	94.4	0.7
1.6	20.3	2	97	17.6

5.7 Testing of Neural Network

For the testing the system, we use the trained structure. In the trained structure, final weight is assigned as shown below.

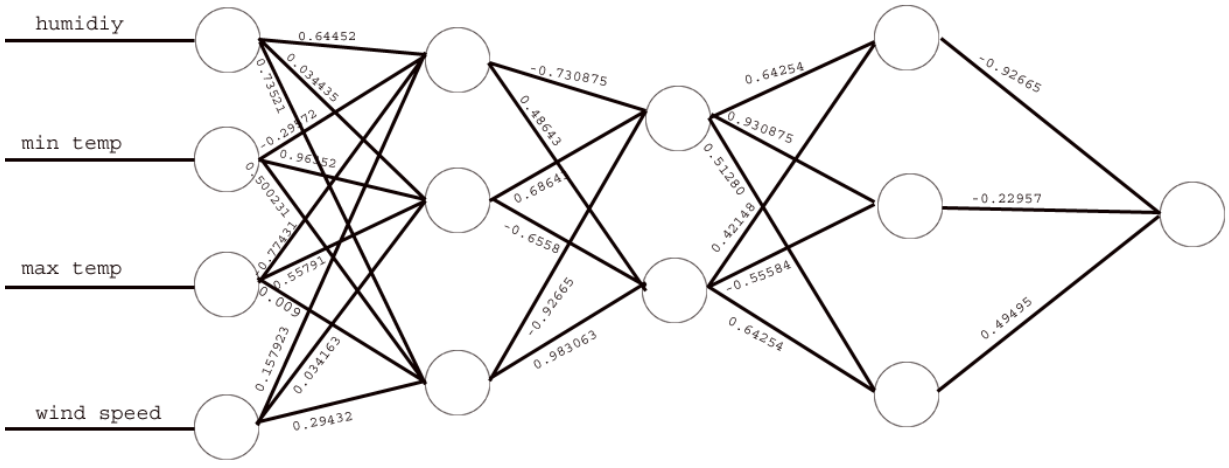

Fig: 5.3 Final NN architecture

In order to test the system we use 20 %, 30 % and 35 % data set of entire data set. i. e 365 tuples is used. Different parameters like humidity, temperature and wind were fed to the trained structure. These input parameters are multiplied with correspond weight and sum is calculated this sum is again passed to the hidden layer and finally we get in output layer and error is calculated. Here it is given sample data set for testing the structure.

5.8 Testing of Decision Tree

The decision tree was tested on the basis of data set shown in Table 5.1. The data were fed to the trained Decision tree. The predicted output was verified against the actual output.

5.9 Result Analysis

The analysis was carried out on the basis of precision, recall and average accuracy. The parameters required for carrying out analysis like True Positive, True Negative, False Positive and False Negative were calculated as shown in Table 5.3 and Table 5.4 for ANN and Decision tree respectively.

Table: 5.4 Analysis Parameters for Decision Tree

TP	10
TN	268
FP	95
FN	4

Table: 5.5 Analysis Parameters for ANN

TP	11
TN	283
FP	80
FN	3

After the running both program for 5 times , it found the following result by for precision, recall, , average system accuracy and system error for decision tree and ANN is calculated as given Table 5.5.

Table: 5.6 Experimentation Results

Algorithm	Precision (%)	Recall (%)	Average System Accuracy (%)	Error (%)
ANN	12.08	78.57	77.98	22.02
Decision Tree	9.52	71.42	73.74	26.26

As shown in Fig 5.4, precision, recall, average system accuracy and error for both decision tree and ANN is represented in graphical way for the comparison.

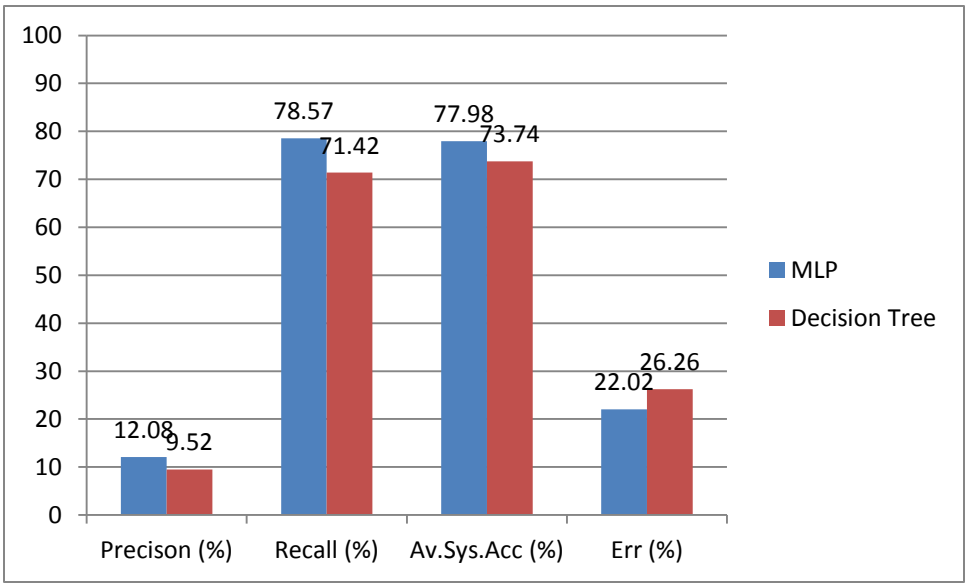

Fig: 5.4 Graph of Experimentation

In the above result shows that neural network has slight less error than a decision tree algorithm.

Result of system is influence by the number of training and testing data and extracted features. Number of parameters also plays important roles for better learning the system. so far- so good, results are promising and can be enhanced further

Chapter 6

6.1 Conclusion

Rainfall prediction problem is addressed in this dissertation work. Here it is used two algorithms for the precipitation prediction. These two algorithms are NN and DT. Data is collected from the department of Hydrology and meteorology and all the dataset is used of airport satiation of Kathmandu valley. For both algorithms same dataset is used and dataset is used of 5 years. Before training the system, data is preprocessed. In the given dataset we have used 4 input parameters namely min temp, max temp, humidity and wind speed. Both NN and DT use same input parameters.

Empirical results shows, Neural Network based classifier (MLP) performs better than Decision tree. MLP classification system has the average system accuracy rate of 77.98%, system error rate of 22.02%, and precision rate of 12.08% recall rate of 78.57%. Similarly, Decision Tree classification system has the average system accuracy rate of 73.74%, system error rate of 26.26%, and precision rate of 9.52% recall rate of 71.42%.

6.2 Future work

- The accuracy of the system can be increased by using other parameters like sea level, pressure and dew point. These parameters could not be incorporated in the current thesis work due to unavailability of data.
- Thesis doesn't work on the calculation of intensity of rainfall.

References

[1] A.D. Kumarasiri and D.U.J. Sonnadara "Rainfall Forecasting: An Artificial Neural Network Approach" Proceedings of the Technical Sessions, 22 (2006) 1-16 Institute of Physics –Sri Lanka

[2] Arti R. Naik, Prof. S.K.Pathan "Weather Classification and Forecasting using Back Propagation Feed-forward Neural Network" International Journal of Scientific and Research Publications, Volume 2, Issue 12, December 2012 1 ISSN 2250-3153

[3] ChattopadhyaySurajit and ChattopadhyayGoutami, 2008, "Identification of the best hidden layer size for three layered neural net in predicting monsoon rainfall in India", Journal of Hydroinformatics, vol. 10(2), pp. 181-188.

[4] Daniel Svozil, Vladimir KvasniEka, JiEPospichal"Introduction to multi-layer feed- forward neural networks" Received 15 October 1996; revised 25 February 1997; accepted 6 June 1997

[5] Edward R. Jones, Ph.D., "An Introduction to Neural Networks" A White PaperPublishing History: December 2004

[6] Elia Georgiana Petre "A Decision Tree for Weather Prediction", Buletinul, Vol. LXI No. 1, 77-82, 2009.

[7] Enireddy. Vamsidhar, K.V.S.R.P.Varma, P.SankaraRao, Ravikanthsatapati"Prediction of Rainfall Using Backpropagation Neural Network Model" (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1119-1121

[8] Geetha G.,.Selvaraj R Samuel, 2011, "Prediction of monthly rainfall in Chennai using back propagation neural network model", International Journal of Engineering Science and Technology, Vol. 3 No. 1, pp. 211-213.

[9] Guoqiang Zhang, B. Eddy Patuwo, Michael Y. Hu "Forecasting with artificial neural networks" Graduate School of Management, Kent State University, Kent, Ohio 44242-0001, USA, Accepted 31 July 1997

[10] GyaneshShrivastava, SanjeevKarmakar, Manoj Kumar Kowar "Application of Artificial Neural Networks in Weather Forecasting" International Journal of Computer Applications (0975 – 8887) Volume 51– No.18, August 2012

[11] Hayati Mohsen, and Mohebi Zahra, 2007, "Application of Artificial Neural Networks for Rainfall Forecasting", World Academy of Science, Engineering and Technology, vol 28, pp. 275-279.

[12] J. Han and M. Kamber, Data Mining: Concepts and Techniques.

[13] Jiawei Han | MichelineKamber | jian Pei "Data Mining Concepts and Techniques" Third Edition Chapter 8: Classification: Basic Concepts

[14] Joshi Jignesh, Patel Vinod M., 2011, "Rainfall-Runoff Modeling Using Artificial Neural Network (A Literature Review)", National Conference on Recent Trends in Engineering & Technology.

[15] Kaya, E.; Barutçu, B.; Menteş, S. "A method based on the van der Hoven spectrum for performance evaluation in prediction of wind speed". Turk. J. Earth Science, 22, 1–9, 2013.

[16] Kumar Abhishek1, Abhay Kumar2, Rajeev Ranjan, Sarthak Kumar "A Rainfall Prediction Model using Artificial Neural Network" 2012 IEEE Control and System Graduate Research Colloquium (ICSGRC 2012)

[17] KumarasiriA.D.andSonnadara D.U.J., 2006, "Rainfall Forecasting: An Artificial Neural Network Approach", Proceedings of the Technical Sessions, vol 22, pp. 1-13.

[18] Lin Gwo-Fong * and Chen Lu-Hsien, 2005, "Application of an artificial neural network to typhoon rainfall forecasting", Hydrological Processes, 19, pp. 1825–1837.

[19] Mekanik F., Lee T.S. and Imteaz M. A., 2011, "Rainfall modeling using Artificial Neural Network for a mountainous region in West Iran".

[20] N.A.Charaniya, S.V.Dudul "Design of Neural Network Models for Daily Rainfall Prediction" International Journal of Computer Applications (0975 – 8887) Volume 61– No.14, January 2013

[21] Narasimha Prasad, Prudhvi Kumar Reddy, Naidu MM, "An Approach to Prediction of Precipitation Using Gini Index in SLIQ Decision Tree", 4th International Conference on Intelligent Systems, Modeling & Simulation, Bangkok, pp.56-60, 2013.

[22] NishantMathur, Sumit Kumar, Santosh Kumar, and Rajni Jindal "The Base Strategy for ID3 Algorithm of Data Mining Using Havrda and Charvat Entropy Based

on Decision Tree" International Journal of Information and Electronics Engineering, Vol. 2, No. 2, March 2012

[23] P.Hemalatha, "Implementation of Data Mining Techniques for Weather Report Guidance for Ships Using Global Positioning System", International Journal Of Computational Engineering Research Vol. 3 Issue. 3 , march 2013.

[24] Rajesh Kumar, Ph.D "Decision Tree for the Weather Forecasting" International Journal of Computer Applications (0975 – 8887) Volume 76– No.2, August 2013 31

[25] Rajurkar M. P., Kothyari U. C., Chaube U. C., 2002, "Artificial neural networks for daily rainfall-runoff modelling", Hydrologkal Sciences-Journals, 47(6), pp. 865-877.

[26] S. Haykin, Neural Networks: A Comprehensive Introduction. Prentice Hall, 1999.

[27] S. Kannan , SubimalGhosh, "Prediction of daily rainfall state in a river basin using statistical downscaling from GCM output", Springer-Verlag, July- 2010.

[28] Soo-YeonJi, Sharad Sharma, Byunggu Yu, Dong Hyun Jeong, "Designing a Rule-Based Hourly Rainfall Prediction Model", IEEE IRI 2012, August – 2012.

[29] SrikalraNiravesh and TanprasertChularat, 2006, "Rainfall Prediction for Chao Phraya River using Neural Networks with Online Data Collection", Malaysia, pp. 13-15.

[30] VamsidharEnireddyVarmaK.V.S.R.P..SankaraRao P satapatiRavikanth, 2010, "Prediction of Rainfall Using Backpropagation Neural Network Model", International Journal on Computer Science and Engineering, Vol. 02, No. 04, pp. 1119-1121.

[31] VidushiSharma ,SachinRai, AnuragDev "A Comprehensive Study of Artificial Neural Networks" Volume 2, Issue 10, October 2012 ISSN: 2277 128X, International Journal of Advanced Research in Computer Science and Software Engineering

[32] Win Khaing Mar and Thu NaingThinn, 2008, "Optimum Neural Network Architecture for Precipitation Prediction of Myanmar", World Academy of Science, Engineering and Technology, vol. 48, pp. 130-134.

Appendix A

Wind	Humidity	Max Temp	Min Temp	Precipitation
0.6	8:45	20	2.4	0
0.7	95	19	1.1	0
0.6	96.9	18.3	1.7	0
0.6	100	18.3	1	0
0.6	96.7	17.8	1.3	0
0.7	96.7	19.5	2	0
0.7	94.4	20.9	2	0
0.7	97	20	2	0
0.6	96.9	20.3	1	0
0.7	96.9	20.5	1.3	0
0.7	93.6	19.5	1.5	0
0.6	96.9	20.3	2.5	0
0.6	97	19.5	2.4	0
0.5	94	19.5	6	0
0.8	95.9	19.5	2.9	0
0.7	94.1	16.6	6.7	0
0.6	92.2	18.6	3.5	0
0.7	97.1	20	2.3	0
0.8	96.9	20.5	3	2.5
0.9	95.7	19.2	2.8	0
0.4	97	18.2	6.5	0
1	97.3	17.6	8.6	6.4
0.4	97.5	11.5	8	5
0.1	97.5	11	7	6.4
0.1	97.5	15.1	3	9.1
0.8	98.4	16.6	0.6	0
0.4	96.7	17.1	0.6	0
0.8	96.9	17.5	0.8	0
0.9	96.8	17.7	3.6	0
0.6	98.5	19.2	3.3	0
1	95.6	17.3	4.3	0
0.6	94.5	19.4	1.5	0
1	100	19.5	1	0
1.3	93.9	19	1	0
1	96.8	17.6	0.6	0
1.2	93.8	18.2	5	0
1.1	86.5	17	2.7	0
0.8	95.7	18.5	2.6	0
0.9	88.6	19.6	1.8	0
0.8	94	22.4	2	0
0.9	91.2	23.5	3.5	0
1	97.1	22.5	3.3	0

1.1	94.4	22.1	3.4	0
0.9	89	23.2	5.1	0
0.9	84.8	23.5	5	0
1.3	92	20.3	4.8	0
0.7	94.7	22.2	4.3	2.5
-99.9	94.5	23.2	4.4	0
-99.9	94.7	24	6.5	0
1.2	94.8	22.9	6.3	0
1.1	94.9	23	5.7	0
1.3	80.6	22.5	6.1	0
1	95	21.6	7.2	0
1	95.1	23.5	7.3	0
1.1	96.3	23.5	11.6	0
0.9	89.2	21.4	12	0
0.6	89.2	25	9.8	0
1.1	91.1	25	10.9	0
1	95.6	25.8	8.4	0
1.7	95.2	27.3	7.7	0
1.7	89.4	28	7.2	0
1.9	96.4	27.8	7	0
1.5	86	27	7.3	0
1.1	87.4	26	10.3	0
1.4	83.2	27.8	9.5	0
3.3	94.2	25.8	7.5	0
0.3	94.2	22.3	6.3	0
0.2	79.9	24.8	5.5	0
1.6	70.9	27.6	6.5	0
1.1	77.7	22.8	7.6	0
0.7	79.7	24.2	11	2.7
-99.9	91.4	23.5	9.7	0
-99.9	87	25.1	12.4	0.4
0.9	95.7	24.4	11.3	0
0.6	97.8	26.1	11.3	0
1.2	96.7	26.1	12	29.2
0.8	97.8	27.4	12.8	2.5
1	97.9	28	15.3	0
1.2	91.4	27.8	15.6	0
1.2	86.7	29.1	16.4	0
1.5	86.4	28.4	18	0
1.5	85.9	28.4	15.4	0
1.1	80.5	29.6	14.2	0
2.2	86	30.6	13	0
1.7	72.1	30.7	11.4	0
1.8	67.4	30.5	9.7	0
1.8	63	28.6	10	0
0.9	68.4	30	9.5	0

1.3	61.2	30.2	10	0
1.1	56.2	29	9	0
1.4	70	27.5	9	0
1.3	69.1	27.6	8.6	0
1.4	65.9	25.6	10.9	0
-99.9	78	26.5	10.9	26.5
-99.9	80.3	28	13.5	2.5
1.7	75.8	27.1	14.5	0
1.3	78.1	29.2	14	0
2	75.5	23.5	13.5	0
0.7	73	27.2	10	2.5
1.5	80.1	27.3	13.6	0.2
1.2	74.7	29.4	13.3	2.4
1.3	86.8	30.6	15	0
1.5	83.7	30.5	13.6	0
2.1	81.4	29.5	13	0.3
1.9	74.2	30.6	13.5	0.2
1.9	73.1	31.3	13.5	0
2.1	63.1	29.8	12.8	7.2
1	92.1	30	14.6	0
1.4	73.8	30	5.7	13.5
1.4	94.2	25.2	13.5	72.4
0.4	90.3	26.7	17	2.5
0.8	90.5	29.3	15.5	0
1.4	83.7	29.5	17	2.5
2	78.5	26.8	13.4	10.4
1.5	77.1	26.8	14.7	0.4
1	68.7	26.2	11.4	1
1.3	83.8	28.5	12.6	0
1.1	81.1	26.4	14.4	0.4
1	73.1	19	14.5	10.6
0.7	83.4	26.6	12.7	18.6
1.6	85.9	25.5	15.4	0
1.8	78.9	26.1	13.5	8.6
1	65	28.5	12.2	6.1
1.2	75.1	28.6	12.7	0
1.5	73.3	28.2	12.6	0
1.1	66.4	28.3	14	0
2	70.3	30	13.6	0
0.9	68	30.8	15.3	0
0.8	70.6	31.3	14.9	0
1.8	65.3	30.9	16	0
1.7	71.1	31.7	18.4	0
1.3	72.8	32.4	16.6	0
1.3	67.7	32.5	16.7	0
1	69	32.4	18.1	0

Appendix B

```
private void Train(int value)
   {
      double error;

      do
      {
         error = 0;

         foreach (Pattern pattern in _patterns)
         {
            double delta = pattern.Output - Activate(pattern,value);
            AdjustWeights(delta);
            error +=  Math.Pow(delta, 2);

         }
         error = Math.Sqrt(error);
         error = error / 1450;

         Console.WriteLine("Iteration {0}\tError {1:0.000}", _iteration, error);
         _iteration++;
         if (_iteration > _restartAfter) Initialise();

      } while (error >= 0.012);
   }

   private void Test(int j)
   {
      if (File.Exists("I:\\virus\\output\\finalWeight.txt"))
      {
         File.Delete("I:\\virus\\output\\finalWeight.txt");
```

```
        }
        Console.WriteLine("\nBegin network testing\nPress Ctrl C to exit\n");
        //while (1 == 1)
        //{
           try
           {

              StreamWriter writer = File.CreateText("I:\\Virus\\output\\neuroOutput0.txt");
             // StreamWriter swriter = File.CreateText("E:\\output\\neuroOutput1.txt");
              string values = string.Empty;
              double val = 0.0;
              //Console.Write("Input x, y: ");
              //string values = Console.ReadLine() + ",0";
             LoadPatterns(1);

              int count=0;
              StreamWriter w = File.CreateText("I:\\Virus\\output\\value.txt");
              foreach (Pattern pat in _patterns)
              {

                 //for (int i = 0; i < pat.Inputs.Length; i++)
                 //{
                    values = pat.Inputs[0].ToString() + "," + pat.Inputs[1].ToString()+"," +
pat.Inputs[2]+","+ pat.Inputs[3] + ",0";
                    count++;

                 //}
                 val = Activate(new Pattern(values, _inputDims),j);

                 w.WriteLine(val);
```

```
                Console.WriteLine("{0:0}\n", val);
                if (val <= 0.41)
                {
                    writer.WriteLine(0);
                }
                else
                {
                    writer.WriteLine(1);
                }
            }
            writer.Close();
            w.Close();

        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }

        Console.ReadLine();
      // }
   }
  private void AdjustWeights(double delta)
   {
      _output.AdjustWeights(delta);
      foreach (Neuron neuron in _hidden)
      {
         neuron.AdjustWeights(_output.ErrorFeedback(neuron));
      }
   }
public static List<Record> buildRecords(int num)
```

```
        {
            List<Record> records = new List<Record>();
            string filePath = string.Empty;
            if (num == 0)
            {
                filePath = "I:\\virus\\decision_data\\train\\final.csv";
            }
            else
            {
                filePath = "I:\\virus\\decision_data\\train\\final.csv"; //for training accuracy
                //filePath= "I:\\virus\\decision_data\\test\\decisionTest1.csv"; //for testing
accuracy
            }


            string line;
            List<DiscreteAttribute> attributes;
            StreamReader file = null;
                try
                {
                    file = new StreamReader(filePath);
                    Record r = null;

                        while ((line = file.ReadLine()) != null)
                        {

                            attributes = new List<DiscreteAttribute>();
                            r = new Record();
                            string[] splitValue = line.Split(',');
                            if (splitValue[0].Equals("temperature"))
                            {
```

```
                continue;
            }
            if(Hw1.NUM_ATTRS != splitValue.Length)
            {
                throw new Exception("unknown error of attributes");
            }
            //double outlook = Convert.ToInt32(splitValue[0]);
            double temperature =Convert.ToDouble(splitValue[0]);
            double humidity =Convert.ToDouble(splitValue[1]);
            double wind =Convert.ToDouble(splitValue[2]);
            string rainfall = splitValue[3];

                //attributes.Add(new DiscreteAttribute("Outlook", outlook));

                //attributes.Add(new DiscreteAttribute("Outlook",
DiscreteAttribute.Sunny));

                //attributes.Add(new DiscreteAttribute("Outlook",
DiscreteAttribute.Rain));

            //attributes.Add(new DiscreteAttribute("Outlook", outlook));

                attributes.Add(new DiscreteAttribute("Temperature", temperature));

                attributes.Add(new DiscreteAttribute("Humidity", humidity));

                attributes.Add(new DiscreteAttribute("Wind", wind));

                attributes.Add(new DiscreteAttribute("RainFall", rainfall));
```

```
                r.setAttributes(attributes);
                records.Add(r);
            }
        }
        catch(IOException e)
        {
          Console.WriteLine("Uh oh, got an IOException error: " + e.Message);

        }
        catch (Exception e)
        {
          Console.WriteLine("Uh oh, got an Exception error: " + e.Message);
        }

        finally
        {
        if (file != null)
        {
          try
          {
            file.Close();
          }
          catch (IOException ioe)
          {
            Console.WriteLine("IOException error trying to close the file: " +
ioe.Message);
          }
        }
      }
      return records;
```

```
            }
public static double calculateEntropy(List<Record> data)
      {
         double entropy = 0;

         if (data.Count == 0)
         {
            // nothing to do
            return 0;
         }

         for (int i = 0; i < Hw1.setSize("PlayTennis"); i++)
         {
            int count = 0;
            for (int j = 0; j < data.Count; j++)//here first record is title like overcast so value
of J start with 1
            {
               Record record = data[j];

               //string val = record.getAttributes().ElementAt(4).getValue().ToString();
               if (i == 0)
               {
                  if (record.getAttributes().ElementAt(3).getBoolValue() == "TRUE")
                  {
                     count++;
                  }
               }
               else
               {
                  if (record.getAttributes().ElementAt(3).getBoolValue() == "FALSE")
```

```
                {
                    count++;
                }
            }
        }

        double probability = count / (double)data.Count;
        if (count > 0)
        {
            entropy += -probability * (Math.Log(probability) / Math.Log(2));
        }
    }

    return entropy;
}

public static double calculateGain(double rootEntropy, List<Double> subEntropies,
List<int> setSizes, int data)
{
    double gain = rootEntropy;

    for (int i = 0; i < subEntropies.Count; i++)
    {
        gain += -((setSizes.ElementAt(i) / (double)data) * subEntropies.ElementAt(i));
    }

    return gain;
}
```