INFORMATION RETRIEVAL OF NON - TEXT BASED DATA USING ANN

Thesis

submitted

In partial fulfillment for the award of the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Engineering

Submitted by

M.Amanullah

(Reg.No. M846900001)

VINAYAKA MISSIONS UNIVERSITY

SALEM - 636 308
BONAFIDE CERTIFICATE

Certified that this thesis entitled “INFORMATION RETRIEVAL OF NON-TEXT BASED DATA USING ANN” is bonafide work of Mr. M. AMANULLAH who carried out the research under my supervision. Certified further, that to the best of my Knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or other candidate.

Dr. V. KHANAA
Supervisor,
Dean-Information Centre,
Bharath University,
Chennai - 600 073.
DECLARATION

I do hereby declare that Thesis entitled “INFORMATION RETRIEVAL OF NON-TEXT BASED DATA USING ANN”, submitted to Vinayaka Missions University, for the award of the Degree of Doctor of Philosophy in Computer Science & Engineering is a record of original work done by me during 2008 – 2015 under the supervision and guidance of Dr.V.Khanaa, Dean-Information Centre, Bharath University, Chennai - 600 073. Further, it has not previously formed the basis for the award of any Degree, Diploma, and Associateship, Fellowship or other similar title to any University.

M.AMANULLAH
ACKNOWLEDGEMENTS

I would like to thank Respected chancellor, pro-chancellor, vice chancellor, directors, registrar and Dean R & D for giving me the opportunity to carryout research work at this esteemed organization.

I would like to thank Mr. S. K. JAGANATHAN, Director, Vinayaka Missions University for providing necessary support during my research work. I am highly thankful for his kind, co-operative nature, his constant inspiration and encouragement to my every work shall always be acknowledged.

Dr. V. KHANAA, Dean – Information Centre, Bharath University, Chennai 600 073, is my guide, has provided the continuous encouragement to complete this research project. He is indeed the driving force behind the research project; I sincerely thank him for his kind gesture.

I am grateful to god for providing me an excellent environment conducive to research.

I am highly thankful to my parents to support me in doing my post-graduation in spite of so many social barriers.

Besides this, a lot of people are there to make me stand at this level, without their motivation, I may never reach this goal and I owe this to all of them.

M. AMANULLAH
The area of soft computing to analyze weather forecasting has gained tremendous importance.

Our World is being threatened with global warming, greenhouse gas effect etc., The collective quantified data is fed into the forecasting system in science and technology. The flexibility and synergetic approach of soft computing easily lends itself to such problems.

Soft computing is an innovative approach to construct computationally intelligent systems that are supposed to possess humanlike expertise within a specific domain, adapt themselves and learn to do better in changing environments, and explain how they make decisions.

Soft computing methodologies spans and connects many areas like fuzzy logic, neuro-computing, evolutionary computing and probabilistic computing. After a brief overview of Soft Computing components, the scholar will analyze some of its synergistic combinations.

Further the scholar presents a comparative study of statistical and neuro-fuzzy network models for forecasting the weather of Visakhapatnam (Andhra Pradesh) an important coastal region of India. The Data recorded by the weather station at the prominent meteorological center of Vishakhapatnam, Andhra Pradesh have been used for the analysis and forecast applying ANFIS and ARIMA, they are
evaluated and compared. The scholar selected this topic for his research, as weather forecasting has been one of the most challenging problems around the world because of both its practical value in meteorology and popular sphere for scientific research.

For developing the models, six years data (2000-2006) comprising daily average temperature (dry-wet conditions), Adaptive Network Based Fuzzy Inference System (ANFIS) and Auto Regressive Moving Average (ARIMA) models have been applied. To ensure the effectiveness of ARIMA and ANFIS techniques, different models employing a different training and test data set have been tested. The criteria of performance evaluation are calculated for estimating and comparing the performances of ARIMA and ANFIS models. Hence, here the scholar explains briefly how neuro-fuzzy models can be formulated using different learning methods and then analyzes whether they can provide the required level of performance for a reliable model for practical weather forecasting. On the obtained the results the most suitable model and network structure are determined according to prediction performance, reliability and efficiency. The performance comparisons of ANFIS and ARIMA models due to $\text{MAE}$ (Moving Average Error), $\text{RMSE-R}^2$ (Root-Mean-Square Error) criteria, indicate that ANFIS yields better results.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>4</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>5</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>7</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>12</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>13</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>17</td>
</tr>
</tbody>
</table>

## CHAPTER 1: INTRODUCTION 18

1.1 INTRODUCTION 18

1.1.1 SOFT COMPUTING APPROACHES 19

1.1.2 SOFT COMPUTING COMPONENTS 20

& TAXONOMY

1.1.2.1 FUZZY SETS 21

1.1.2.1.1 FUZZY SETS AND MEMBERSHIP FUNCTIONS 23

1.1.2.1.2 MF TERMINOLOGY 24

1.1.2.1.3 FUZZY EXPERT SYSTEM 25

1.1.2.1.4 MODELING FUZZY EXPERT SYSTEMS 27

1.1.2.2 NEURO COMPUTING 28

1.1.2.2.1 THE BUILDING BLOCKS OF NEURAL NETWORKS 29

1.1.2.2.2 MODELS OF AN ARTIFICIAL NEURON 31

1.1.2.2.3 LEARNING RULES 35

1.1.2.2.4 BACK PROPAGATION 37

1.1.2.2.4.1 BACK PROPAGATION ALGORITHM 39

1.1.2.3 GENETIC ALGORITHMS 43
1.1.2.4 PROBABILISTIC REASONING 43
1.1.2.5 EVOLUTIONARY COMPUTING 44
1.1.3 BRIDGE BETWEEN FUZZY LOGIC AND WEATHER FORECASTING 45
1.1.4 ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)
   1.1.4.1 ANFIS ARCHITECTURE 46
   1.1.4.2 HYBRID LEARNING ALGORITHM 49
1.1.5 AUTO REGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)
   1.1.5.1 ARIMA PROCESSES 53
1.2 MATLAB TOOLBOX 55
   1.2.1 ANFIS EDITOR (ANFISEDIT) 55
   1.2.2 TRAINING ADAPTIVE NEURO FUZZY INFERENCE SYSTEM USING THE ANFIS EDITOR GUI
   1.2.3 LOADING, PLOTTING AND CLEARING THE DATA 57
   1.2.4 GENERATING OR LOADING THE INITIAL FIS STRUCTURE 58
   1.2.5 TRAINING THE FIS 59
   1.2.6 VALIDATING THE TRAINED FIS 60
1.3 TERMINOLOGIES USED IN WEATHER FORECASTING 61
   1.3.1 WEATHER 61
   1.3.2 METEOROLOGY 61
1.4 CONSTRAINTS IN WEATHER FORECASTING 62
   1.4.1 AREA CONSTRAINT 62
   1.4.2 TIME CONSTRAINT 62
   1.4.3 SUBJECTIVITY 63
   1.4.4 GLOBAL WARMING EFFECT 63
   1.4.5 CHANGE IN ATMOSPHERIC CONDITIONS 63
1.5 WEATHER ELEMENTS 64
1.6 IMPORTANCE OF WEATHER ELEMENTS 64
1.7 EFFECTS OF WEATHER ELEMENTS 65
   1.7.1 EFFECT OF TEMPERATURE 65
   1.7.2 EFFECT OF MOISTURE 66
1.7.3 EFFECT OF RAINFALL

1.8 WEATHER ELEMENTS USED IN THE RESEARCH: DEFINITION AND MEASUREMENTS

1.8.1 TEMPERATURE

1.8.1.1 INSTRUMENTS FOR THE MEASUREMENT OF AIR TEMPERATURE

1.8.2 ACTUAL VAPOUR PRESSURE

1.8.2.1 INSTRUMENTS FOR THE MEASUREMENT OF AIR PRESSURE

1.8.3 HUMIDITY AND EVAPORATION

1.9 WEATHER FORECASTING

1.9.1 EXPERIMENTAL WEATHER FORECAST AGENCIES

1.9.2 TYPES OF WEATHER FORECASTS

1.10 INTRODUCTION TO PROPOSED RESEARCH WORK

1.10.1 AIM OF RESEARCH

1.10.2 SIGNIFICANCE OF RESEARCH

1.10.3 PROBLEM DEFINITION

1.10.4 NEED FOR STUDY

1.11 ORGANIZATION OF THE THESIS

CHAPTER 2: LITERATURE SURVEY

2.1 HISTORICAL BACKGROUND OF WEATHER FORECASTING

2.2 APPROACHES FOR WEATHER FORECASTING

2.2.1. FUZZY LOGIC SYSTEMS

2.2.2. NEURO COMPUTING

2.2.3. NEURO FUZZY SYSTEMS

2.2.4. ARIMA MODEL

2.2.5. GAPS IN EXISTING RESEARCH

2.3 OBJECTIVES OF PROPOSED RESEARCH WORK

2.3.1 PROPOSED APPROACH

CHAPTER 3: DESIGN METHODOLOGIES

3.1 INTRODUCTION
3.2 STEPS OF DESIGN METHODOLOGY 96
3.3 DATA ANALYSIS 98

3.3.1. DATA COLLECTION 98
  3.3.1.1. INPUT AND OUTPUT VARIABLES 98
  3.3.1.2. PRIMARY SOURCE OF DATA 99
  3.3.1.3. RAW DATA COLLECTION 100

3.4 GENERAL FRAMEWORK OF PROPOSED STUDY 101

3.4.1. DESIGN OF ANFIS 101
  3.4.1.1. SOFT COMPUTING MODEL USED IN THIS RESEARCH 103
  3.4.1.2. STRUCTURE OF ANFIS 104
  3.4.1.3. HYPOTHESIS OF THE MODEL USED 106
  3.4.1.4. MEMBERSHIP FUNCTION USED 107
  3.4.1.5. ANFIS SETUP 107
    3.4.1.5.1. SUGENO ANFIS SETUP 107
    3.4.1.5.2. TRAINING OF ANFIS MODEL 108

3.4.2. DESIGN OF ARIMA MODEL 109
  3.4.2.1. THE STEPS IN THE ARIMA MODEL BUILDING 109

3.5 MEANS OF ERROR PERFORMANCE 111

CHAPTER 4: RESULTS AND DISCUSSION 115

4.1 INTRODUCTION 115
4.2 DATA DESCRIPTION 116
  4.2.1 TEMPERATURE 117
  4.2.2 AVERAGE VAPOUR PRESSURE 118
  4.2.3 RELATIVE HUMIDITY 118

4.3 DATA ANALYSIS, PROCESSING AND NORMALIZATION 119
4.4 EXPERIMENTS ON FORECASTING USING STATISTICAL ARIMA MODEL 120
4.5 EXPERIMENTS ON FORECASTING USING ANFIS MODEL 123
4.6 RESULTS OF VARIOUS ANFIS MODELS 152
4.7 COMPARISON BETWEEN ANFIS AND ARIMA MODEL 154
LIST OF TABLES

Table 1.1- SOFT COMPUTING COMPONENTS 21
Table 1.2- COMPARISON BETWEEN FUZZY SYSTEMS AND NEURAL NETWORKS 42
Table 1.3- TWO PASSES IN THE HYBRID LEARNING ALGORITHM FOR ANFIS 50
Table 3.1- LIST OF INPUT AND OUTPUT VARIABLES 99
Table 3.2- SAMPLE RAW DATA (BEFORE PROCESSING) 101
Table 3.3- DESCRIPTIVE STATISTICS OF RESEARCH METHODOLOGY 113
Table 4.1- NORMALIZED DATA AFTER PROCESSING (SAMPLE) 120
Table 4.2- MEMBERSHIP FUNCTIONS AND LIST OF FUZZY VARIABLES 124
Table 4.3- COMPARISON OF GRID PARTITIONING METHOD AND SUBTRACTIVE CLUSTERING 130
Table 4.4- RESULTS OF VARIOUS ANFIS MODELS 152
Table 4.5- MAE COMPARISON BETWEEN ANFIS AND ARIMA FOR TRAINING AND TESTING DATA 155
Table 4.6- RMSE COMPARISON BETWEEN ANFIS AND ARIMA FOR TRAINING AND TESTING DATA 155
Table 4.7- R SQUARED COMPARISON BETWEEN ANFIS AND ARIMA FOR TRAINING AND TESTING DATA 156
LIST OF FIGURES

Figure 1.1- A NEURAL CHARACTER RECOGNIZER 20
Figure 1.2- MF FORMULATION & PARAMETERIZATION 25
Figure 1.3- BASIC ARCHITECTURE OF FUZZY EXPERT SYSTEM 27
Figure 1.4- THE BASIC NEURAL UNIT PROCESSES THE INPUT INFORMATION INTO THE OUTPUT INFORMATION 30
Figure 1.5- DENDRITIC REPRESENTATION OF NEURON 32
Figure 1.6- SIGNAL FLOW GRAPH 32
Figure 1.7- BLOCK-DIAGRAM REPRESENTATION 32
Figure 1.8- A SYNAPSE IN A DENDRITIC REPRESENTATION OF A NEURON 34
Figure 1.9- A SINGLE NEURON WITH A BIASING INPUT 35
Figure 1.10- BACK PROPAGATION 39
Figure 1.11- BACK PROPAGATION TRAINING SET 40
Figure 1.12- APPLYING A TRAINING PAIR TO A NETWORK 40
Figure 1.13- SINGLE CONNECTION LEARNING IN A BACK PROPAGATION NETWORK 41
Figure 1.14- A TWO-INPUT FIRST-ORDER SUGENO FUZZY MODEL WITH TWO RULES 46
Figure 1.15- EQUIVALENT ANFIS ARCHITECTURE 46
Figure 1.16- BASIC STRUCTURE OF ANFIS 47
Figure 1.17- INPUT DATA AS VARIABLE NAME VIZAG_TEST_DATA 57
Figure 1.18- TRAINING AND TESTING DATA GRAPH AS CIRCLES AND PLUSES 57
Figure 1.19- STRUCTURE OF ANFIS MODEL WITH THREE INPUTS AND ONE OUTPUT 59
Figure 1.20- TRAINING THE ANFIS MODEL WITH 100 EPOCHS 60
Figure 1.21- COMPARISON OF TESTING DATA AND OBSERVED DATA 61
Figure 1.22- TYPES OF WEATHER FORECAST 74
Figure 3.1- DESIGN OF RESEARCH METHODOLOGY 97
1 OUTPUT (3, -5, -5, -1) AND COMPARISON OF PREDICTED AND OBSERVED VALUE

Figure 4.23- RULE VIEWER WITH 75 RULES 136
Figure 4.24- SURFACE VIEWER WITH REFERENCE INPUT 136
[NAN NAN 0.62 0.16]
Figure 4.25- INPUT, OUTPUT DATA WITH ITS MFS(3,-3,-5,-1) 137
Figure 4.26- TRAINING OF ANFIS HYBRID MODEL WITH 40 EPOCHS 137
Figure 4.27- ANFIS ARCHITECTURE WITH ITS MF(3,-3,-5,-1) 138
Figure 4.28- ANFIS HYBRID MODEL (3,-3,-5,-1) 138
Figure 4.29- RULE VIEWER WITH 45 RULES 139
Figure 4.30- SURFACE VIEWER FOR ANFIS MODEL (3,-3,-5,-1) 139
Figure 4.31- COMPARISON OF PREDICTED AND OBSERVED VALUE (3,-3,-5,-1)
Figure 4.32- LOADING TRAINING DATA SET(OOOOO) AND TESTING DATA SET(*****)
Figure 4.33- COMPARISON OF PREDICTED AND OBSERVED VALUE (3,-3,-5,-1)
Figure 4.34- INPUT, OUTPUT DATA WITH ITS MF (5,-5,-5,-1) 142
Figure 4.35- ANFIS ARCHITECTURE (5, -5, -5, -1) 142
Figure 4.36- TRAINING OF ANFIS HYBRID MODEL WITH ITS MF (5,-5,-5,-1)
Figure 4.37- COMPARISON OF PREDICTED AND OBSERVED VALUE FOR MF (5,-5,-5,-1)
Figure 4.38- RULE VIEWER (5, -5, -5, -1) 144
Figure 4.39- SURFACE VIEWER FOR MF (5,-5,-5,-1) 144
Figure 4.40- INPUT, OUTPUT DATA WITH ITS MF (5,-3,-3,-1) 146
Figure 4.41- ANFIS ARCHITECTURE (5, -3, -3, -1) 146
Figure 4.42- COMPARISON OF PREDICTED AND OBSERVED VALUE WITH ITS MF (5,-3,-3,-1)
Figure 4.43- SURFACE VIEWER FOR MF (5,-3,-3,-1) 147
Figure 4.44- INPUT, OUTPUT DATA WITH ITS MF (5,-5,-3,-1) 149
Figure 4.45- COMPARISON OF PREDICTED AND OBSERVED VALUE WITH ITS MF (5,-5,-3,-1)
Figure 4.46- INPUT, OUTPUT DATA WITH ITS MF (5,-3,-5,-1) 150
Figure 4.47- ANFIS ARCHITECTURE WITH ITS MF (5,-3,-5,-1) 150
Figure 4.48- COMPARISON OF PREDICTED AND OBSERVED VALUE WITH ITS MF (5,-3,-5,-1) 151
Figure 4.49- RULE VIEWER MF (5,-3,-5,-1) 152
Figure 4.50- SURFACE VIEWER WITH ITS MF (5,-3,-5,-1) 152
LIST OF ABBREVIATIONS

AI    Artificial Intelligence
ANFIS Adaptive Network based Fuzzy Inference System
anfisedit Anfis Editor Toolbox
ANN    Artificial Neural Network
ARIMA  Auto Regressive Integrated Moving Average
AVP    Average Vapour Pressure
CANFIS Co-active Neuro Fuzzy Inference System
FIS    Fuzzy Inference System
GUI    Graphical User Interface
IMD    Indian Meteorology Department
MatLab Laboratory of Matrices
MAX TEMP Maximum Temperature
MF     Membership Function
MIN TEMP Minimum Temperature
MAE    Mean Absolute Error
NCMRWF National Centre For Medium Range Weather Forecasting
RH     Relative Humidity
RH_E   Relative Humidity at Evening
RH_M   Relative Humidity at Morning
RMSE   Root Mean Square Error
WP     Weather Profile
CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In this chapter, the scholar wants to explain the background of the research study under subject, concept of Soft Computing techniques for analysis of Weather forecasting using ANFIS and ARIMA models. In addition to it the research question and aim of the study has also included in this chapter.

Intelligent system is a form of knowledge discovery which is essential for solving problems in domains involving large volume of data. The individual data sets may be gathered and studied collectively for purposes other than those for which they were originally created. New knowledge may also be obtained in this process, by eliminating the cost of additional data collection, which often exists in vast quantities over the internet in an unstructured format.

Typically, a real life data must not only be cleaned of error and redundancy, but must also be organized in a passion that makes sense to the problem. There can be existed imperfections in raw input data needed for knowledge questions mainly due to uncertainty, vagueness and incompleteness. While incompleteness arises due to missing or unknown data, uncertainty or vagueness can be caused by errors in physical measurements due to incorrect measuring devices or by a mixture of noisy and pure signals.

Soft Computing is a grouping of methodologies that works synergistically and provides in one form or another, flexible information processing capability for
handling real time ambiguous problems. Its aim is to exploit the tolerance for imprecision, uncertainty, appropriate reasoning and partial truth in order to achieve traceability, robustness, and low cost solutions. The guiding principle is to device methods of computation that lead to an acceptable solution at low cost by seeking an approximate solution to imprecisely or precisely formulated problems.

The main constituents of Soft Computing at this juncture, include fuzzy logic, neural networks, genetic algorithms, rough sets and signal processing tools such as wavelets. Each of them contributes a distinct methodology for addressing the problems in its domain. The result is more intelligent and robust system providing a human interpretable low cost, appropriate solution as compared to traditional techniques.

1.1.1 SOFT COMPUTING APPROACHES

Soft Computing is a new approach for constructing computationally an intelligent system. This technique came into market very recently. Now-a-days, all complex real world problems require an intelligent system which can combine knowledge, techniques, and methodologies from various sources. These systems are supposed to have humanlike expertise in any specific domain, adapt themselves to do better in changing environment and able to give feedback on adapting of any decision. In real world computing problems, it is beneficiary to use several computing techniques synergistically rather than exclusively constructing complementary hybrid intelligent systems.
The most essential concept which helps to design this type of expertise intelligent system is by Neuro-Fuzzy Computing: neural networks that recognize patterns and adapt themselves to cope with changing environment, fuzzy inference system that incorporate human knowledge and perform inference and decision making. The integration of these two different approaches, together with certain derivative free optimization techniques, results in a novel discipline called Neuro-Fuzzy and Soft Computing.

1.1.2 SOFT COMPUTING COMPONENTS AND TAXONOMY

Soft Computing is a modern approach in computing world which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision, Lofti A. Zadeh, Father of the Soft Computing.

Soft computing consists of several different computing paradigms, including neural networks, fuzzy set theory, probabilistic reasoning and derivative free optimization methods such as genetic algorithms. Each of these constituent methods has their own strength, as summarized in Table 1.1.

![Fig1.1: A neural character recognizer](image-url)
The collective blend of these methodologies forms the basis of soft computing. The synergism allows soft computing to incorporate human knowledge effectively, deal with imprecision and uncertainty and learns to adapt unknown or changing environment for better performance. For adaptation and learning, soft computing requires extensive computation. In this sense, soft computing shares the same characteristics as computational intelligence. Soft computing does not perform much symbolic manipulation so we can view it as a brand new discipline that complements the conventional artificial intelligence approaches and vice versa.

**Table 1.1: Soft computing Components**

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Strength</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy Set Theory</td>
<td>Computing with words, Knowledge representation using fuzzy if-then rules</td>
</tr>
<tr>
<td>Neural Network</td>
<td>Identification, Learning and adaptation</td>
</tr>
<tr>
<td>Genetic Algorithms</td>
<td>Systematic random search, optimization</td>
</tr>
<tr>
<td>Probabilistic Reasoning</td>
<td>Propagation of belief</td>
</tr>
<tr>
<td>Conventional AI</td>
<td>Symbolic manipulation</td>
</tr>
</tbody>
</table>

**1.1.2.1 FUZZY SETS**

Fuzzy sets were introduced in 1965 by Zadeh as a new way of representing vagueness in everyday life. This theory provides an approximate and an effective means for describing the characteristics of a system that is too
complex or ill-defined to admit precise mathematical analysis. The fuzzy approach is based on the premise that the key elements in human thinking are not just numbers but can be approximated to tables of fuzzy sets or in other words, classes of objects in which the transition from membership to non-membership is gradual rather than abrupt. Much of the logic behind human reasoning is not the traditional two valued or even multi valued logic, but logic with fuzzy truth, fuzzy connectives, and fuzzy rules of inference.

Fuzzy set theory is reputed to handle, to a reasonable extent, uncertainties in various applications particularly in decision making models under different kinds of risks, subjective judgment, vagueness and ambiguity. The deficiencies may result from various reasons, namely, incomplete, imprecise, not fully reliable, vague, or contradictory information depending on the problem. Since this theory is a generalization of the classical set theory, it has a greater flexibility to capture various aspects of incompleteness or imperfection in information about a situation.

The fuzzy inference system has a structured knowledge representation in the form of fuzzy if-then rules but it lacks the adaptability to deal with changing external environments. Thus, we incorporate neural network learning concept in fuzzy inference systems, resulting in a neuro-fuzzy modeling, a pivotal technique in soft computing.

A fuzzy expert system is simply an expert system that uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about
data (Schneider et al., 1996). The rules in a fuzzy expert system are usually of a form similar to the following:

\[
\text{If } A \text{ is low and } B \text{ is high then } X = \text{ medium}
\]

Where \( A \) and \( B \) are input variables, \( X \) is an output variable.

Here low, high, and medium are fuzzy sets defined on \( A, B, \) and \( X \) respectively. The antecedent describes to what degree the rule applies, while the rule’s consequent assigns a membership function to each of one or more output variables.

Let \( X \) be a space of objects and \( x \) be a generic element of \( X \). A classical set \( A \), a subset or equal to \( X \), is defined as a collection of elements or objects \( x \in X \), such that \( x \) can either belong or not belong to the set \( A \). By defining a characteristic function for each element \( x \) in \( X \), we can represent classical set \( A \) by a set of ordered pairs \((x,0)\) or \((x,1)\) which indicates \( x \) does not belong to \( A \) or \( x \) belongs to \( A \) respectively. Unlike the aforementioned conventional set, a fuzzy set expresses the degree to which an element belongs to a set. Hence the characteristic function of a fuzzy set is allowed to have values between 0 and 1, which denotes the degree of membership of an element in a given set.

1.1.2.1.1 FUZZY SETS AND MEMBERSHIP FUNCTIONS

If \( X \) is a collection of objects denoted generically by \( x \), then a Fuzzy set \( A \) in \( X \) is defined as a set of ordered pairs

\[
A = \{(x,\mu_A(x))|x \in X\}\ ...........................................................................................................(1.1)
\]
Where $\mu_A (x)$ is called the *membership function* (MF) for the fuzzy set $A$. The MF maps each element of $X$ to a membership grade between zero and one.

Before going to study of fuzzy expert system, here scholar present a detailed introduction to the theory and terminology of fuzzy disciplines including fuzzy sets, fuzzy rules, fuzzy reasoning and fuzzy inference systems.

As fuzzy set theory have a basic concept in mathematics, it is not misunderstood that it has a tremendous impact on various mathematical disciplines, such as logic, algebra, data analysis, statistics etc. But more importantly, fuzzy set theory has a vast numbers of applications that make use of these mathematical disciplines.

In fuzzy rule-based systems, knowledge is represented by if–then rules $P \Rightarrow Q$.

Fuzzy rules consist of two parts: an antecedent part $P$ stating conditions on the input variable(s); and a consequent part $Q$ describing the corresponding values of the output variable(s). Usually, the case of a single output variable is considered. In Mamdani type models, both antecedent and consequent parts consist of fuzzy statements concerning the value of the variables, where as in Sugeno type models, the consequent part expresses a non-linear relationship between the input variables and the output variable (Takagi and Sugeno, 1985).

**1.1.2.1.2 MF TERMINOLOGY**

**MFs of one dimension:** Define several classes of parameterized MF of one dimension with a single input
Fig1.2: MF formulation and parameterization

**Triangular MFs:**

\[
trimf(x ; a, b , c) = \max\left(\min\left(\frac{x - a}{b - a}, \frac{c - x}{c - b}\right), 0\right)
\]

.................................................. (1.2)

**Trapezoidal MFs:**

\[
trapmf(x ; a, b, c, d) = \max\left(\min\left(\frac{x - a}{b - a}, \frac{d - x}{d - c}\right), 0\right)
\]

..................................................(1.3)

**Gaussian MFs:**

\[
gaussmf(x ; c , \sigma) = e^{-\frac{1}{2} \left(\frac{x - c}{\sigma}\right)^2}
\]

..................................................(1.4)

**Generalized bell MFs:**

\[
gbelimf(x ; a, b , c) = \frac{1}{1 + \left|\frac{x - c}{b}\right|^{2b}}
\]

..................................................(1.5)

### 1.1.2.1.3 FUZZY EXPERT SYSTEM

Figure 1.3 illustrates the basic architecture of a fuzzy expert system. The main components are fuzzification interface, a fuzzy rule base (knowledge base),
an inference engine (decision-making logic), and a defuzzification interface. The input variables are fuzzified whereby the membership functions defined on the input variables are applied to their actual values, to determine the degree of truth for each rule antecedent. Fuzzy *if-then* rules and fuzzy reasoning are the backbone of fuzzy expert systems, which are the most important modeling tools based on fuzzy set theory. The fuzzy rule base is characterized in the form of *if-then* rules in which the antecedents and consequents involve linguistic variables. The collection of these fuzzy rules forms the rule base for the fuzzy logic system. Using suitable inference procedure, the truth value for the antecedent of each rule is computed, and applied to the consequent part of each rule. This results in one fuzzy subset to be assigned to each output variable for each rule. Again, by using suitable composition procedure, all the fuzzy subsets assigned to each output variable are combined together to form a single fuzzy subset for each output variable. Finally, defuzzification is applied to convert the fuzzy output set to a crisp output.

The basic fuzzy inference system can take either fuzzy inputs or crisp inputs, but the outputs it produces are always fuzzy sets. The defuzzification task extracts the crisp output that best represents the fuzzy set. With crisp inputs and outputs, a fuzzy inference system implements a nonlinear mapping from its input space to output space through a number of fuzzy *if-then* rules.

In what follows, the two most popular fuzzy inference systems are introduced that have been widely deployed in various applications. The
differences between these two fuzzy inference systems lie in the consequents of their fuzzy rules, and thus their aggregation and defuzzification procedures differ accordingly.

Fig1.3: Basic architecture of fuzzy expert system

1.1.2.1.4 MODELING FUZZY EXPERT SYSTEMS

Fuzzy expert system modeling can be done using the following steps.

- Select relevant input and output variables. Determine the number of linguistic terms associated with each input/output variable. Also, choose the appropriate family of membership functions, fuzzy operators, reasoning mechanism, and so on.

- Choose a specific type of fuzzy inference system. In most cases, the inference of the fuzzy rules is carried out using the ‘min’ and ‘max’ operators for fuzzy intersection and union.

- Design a collection of fuzzy *if-then* rules. To formulate the initial rule base, the input space is divided into multidimensional partitions and then actions are assigned to each of the partitions. In most applications, the partitioning is achieved using one dimensional membership functions using fuzzy *if-then* rules.
The consequent parts of the rule represent the actions associated with each partition. It is evident that the MFs and the number of rules are tightly related to the partitioning.

1.1.2.2 NEURO COMPUTING

Neural networks are adaptive statistical models based on an analogy with the structure of the brain. They are adaptive because they can learn to estimate the parameters of some population using a small number of examples at a time. They do not differ essentially from standard statistical models. For example, one can find neural network architectures akin to discriminate analysis, principal component analysis, logistic regression, and other techniques. Neural networks are used as statistical tools in a variety of fields, including statistics, engineering, and even physics. In fact, the same mathematical tools can be used to analyze standard statistical models and neural networks. They are also used as models of cognitive processes by neuro and cognitive scientists.

Basically, neural networks are built from simple units, sometimes called neurons or cells by analogy with the real thing. These units are linked by a set of weighted connections. Learning is usually accomplished by modification of the connection weights. Each unit corresponds to a feature or a characteristic of a pattern that we want to analyze or that we want to use as a predictor. These networks usually organize their units into several layers. The first layer is called the input layer, the last one the output layer. The intermediate layers (if any) are called the hidden layers. The information to be analyzed is fed to the neurons of
the first layer and then propagated to the neurons of the second layer for further processing. The result of this processing is then propagated to the next layer and so on until the last layer. Each unit receives some information from other units (or from the external world through some devices) and processes this information, which will be converted into the output of the unit.

The aim of the network is to learn or to discover some association between input and output patterns, or to analyze, or to find the structure of the input patterns. The learning process is achieved through the modification of the connection weights between units. The learning process specifies the “algorithm” used to estimate the parameters.

1.1.2.2.1 THE BUILDING BLOCKS OF NEURAL NETWORKS

Neural Networks are made of basic units arranged in layers. A unit collects information provided by other units (or by the external world) to which it is connected with weighted connections called synapses. These weights, called synaptic weights, multiply the input information:

A positive weight is considered as excitatory, a negative weight as inhibitory. Each of these units is a simplified model of a neuron and transforms its input information into an output response. This transformation involves two steps: First, the activation of the neuron is computed as the weighted sum of inputs, and second this activation is transformed into a response by using a transfer function.
Fig 1.4: The basic neural unit processes the input information into the output information

Formally, if each input is denoted $x_i$, and each weight $w_i$, then the activation is equal to $a = \sum x_i w_i$, and the output denoted ‘o’ is obtained as $o = f(a)$. Any function whose domain is the real numbers can be used as a transfer function. The most popular ones are the linear function ($o \propto a$), the step function (activation values less than a given threshold are set to 0 or to −1 and the other values are set to +1), the logistic function $f(x) = \frac{1}{1 + \exp(-x)}$, which maps the real numbers into the interval $[-1, 1]$ and whose derivative, needed for learning, is easily computed $f'(x) = f(x) [1 - f(x)]$, and the normal or Gaussian function $o = (}$
neurons and the synaptic weights, completely specify the behavior of the network.

A typical artificial neuron and the modeling of a multilayered neural network are illustrated in Figure 3.3. The signal flow from inputs $x_1, \ldots, x_n$ is considered to be unidirectional, which are indicated by arrows, as is a neuron’s output signal flow ($O$). The neuron output signal $O$ is given by the following relationship:

$$O = f(\text{net}) = f \left( \sum_{j=1}^{n} w_j x_j \right)$$

Where $w_j$ is the weight vector, and the function $f(\text{net})$ is referred to as an activation (transfer) function. The variable $\text{net}$ is defined as a scalar product of the weight and input vectors,

$$\text{net} = w^T x = w_1 x_1 + \cdots + w_n x_n$$

Where $T$ is the transpose of a matrix, and, in the simplest case, the output value $O$ is computed as

$$O = f(\text{net}) = \begin{cases} 1 & \text{if } w^T x \geq \theta \\ 0 & \text{otherwise} \end{cases}$$

Where $\theta$ is called the threshold level; and this type of node is called a linear threshold unit.

1.1.2.2.2 Models of an Artificial Neuron

Artificial neural networks are nonlinear information processing devices which are built from interconnected elementary processing devices called
neurons. An artificial neuron is a single input output signal processing element which can be thought of as a simple model of a non-branching biological neuron.

From a dendritic representation of a single neuron we can identify synapses arranged along a linear dendrite which aggregates the synaptic activities, and a neuron body or axon-hillock generating an output signal.

Graphically, an artificial single p-input neuron is represented in one of the following forms:

Fig. 1.5: Dendritic representation of neuron

Fig 1.6: Signal flow graph

Fig. 1.7: Block-diagram representation

\[ v = w_1 \cdot x_1 + \cdots + w_p \cdot x_p = \mathbf{w} \cdot \mathbf{x} ; \quad y = \varphi(v) \]  
.................................(1.8)
The pre-synaptic activities are represented by a element column vector of input signals

\[ \mathbf{x} = [x_1 \ldots x_p]^T \]

.................................................................(1.9)

In other way, the space of input patterns is p dimensional. Synapses are characterized by adjustable parameters called weights or synaptic strength parameters. The weights are arranged in a p element row vector:

\[ \mathbf{w} = [w_1 \ldots w_p] \]

.................................................................(1.10)

In a signal flow representation of a neuron, synapses are arranged in a layer of input nodes. A dendrite is replaced by a single summing node. Weights are now attributed to branches(connections) between input nodes and the summing node. Passing through synapses and a dendrite (or a summing node), input signals are aggregated (combined) into the activation potential, which describes the total post-synaptic activity. The activation potential is formed as a linear combination of input signals and synaptic strength parameters, that is, as an inner product of the weight and input vectors:

Equation 1.10:

\[ v = \sum_{i=1}^{p} w_i x_i = \mathbf{w} \cdot \mathbf{x} = \begin{bmatrix} w_1 & w_2 & \cdots & w_p \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix} \]

...............(1.11)
Subsequently, the activation potential is passed through an activation function, \( \psi(.) \), which generates the output signal:

Equation 1.12:

\[
y = \psi(v)
\]  

The activation function is typically a saturating function which normalizes the total post-synaptic activity to the standard values of output signal.

The block-diagram representation encapsulates basic operations of an artificial neuron, namely, aggregation of pre-synaptic activities, Equation 1.11, and generation of the output signal, Equation 1.12.

\[
v_i = v_{i-1} + w_i \cdot x_i \quad \text{(signal aggregation)}
\]  

A synapse is classified as excitatory, if a corresponding weight is positive, \( w_i > 0 \), and as inhibitory, if a corresponding weight is negative, \( w_i < 0 \). In the synapse model of Figure 1.8 we can identify: a storage for the synaptic weight, augmentation (multiplication) of the pre-synaptic signal with the weight parameter, and the dendritic aggregation of the post-synaptic activities.

A single synapse in a dendritic representation of a neuron can be represented by the following block-diagram:
It is sometimes convenient to add an additional parameter called threshold, $\theta$ or bias $b=\Theta$. It can be done by fixing one input signal to be constant. Than we have

$$x_p = +1, \quad \text{and} \quad w_p = b = -\theta$$

(1.14)

With this addition, the activation potential is calculated as:

$$\hat{v} = \sum_{i=1}^{p} w_i x_i = v - \theta, \quad v = \sum_{i=1}^{p-1} w_i x_i$$

(1.15)

where $\hat{v}$ is the augmented activation potential.

**Fig.1.9:** A single neuron with a biasing input

### 1.1.2.2.3 LEARNING RULES

Neural Networks are adaptive statistical devices which mean that they can modify iteratively the values of their parameters (i.e., the synaptic weights) as a function of their performance. These changes are made according to learning rules which can be characterized as supervised (when a desired output is known and used to compute an error signal) or unsupervised (when no such error signal is used).
The Widrow-Hoff rule is the most widely known supervised learning rule. It uses the difference between the actual input of the cell and the desired output as an error signal for units in the output layer. Units in the hidden layers cannot compute directly their error signal but estimate it as a function (e.g., a weighted average) of the error of the units in the following layer. This adaptation of the Widrow-Hoff learning rule is known as error back propagation.

The Hebbian rule is the most widely known unsupervised learning rule; it is based on the work done by the Canadian neuropsychologist Donald Hebb, who theorized that neural learning (i.e., synaptic change) is a local phenomenon expressible in terms of the temporal correlation between the activation values of neurons. Specifically, the synaptic change depends on both pre-synaptic and post-synaptic activities and states that the change in a synaptic weight is a function of the temporal correlation between the pre-synaptic and postsynaptic activities. The value of the synaptic weight between two neurons increases whenever they are in the same state and decreases when they are in different states.

The learning can be described either by differential equations (continuous-time)

\[
\dot{W}(t) = L(W(t), x(t), y(t), d(t))
\] ................................. (1.16)

or by the difference equations (discrete-time)

\[
W(n + 1) = L(W(n), x(n), y(n), d(n))
\] .................................(1.17)
Where d is an external teaching/supervising signal used in supervised learning. This signal is not present in networks employing unsupervised learning.

The discrete-time learning law is often used in a form of a weight update equation:

\[
W(n+1) = W(n) + \Delta W(n) \\
\Delta W(n) = L(W(n), x(n), y(n), d(n))
\]

\[...............(1.18)\]

1.1.2.2.4 BACK PROPAGATION

Back propagation is the most widely used supervised learning algorithm in neural computing. It is very easy to implement. A back propagation network includes one or more hidden layers. This type of network is considered feed forward because there are no interconnections between the output of a processing element and the input of a node in the same layer or in a preceding layer. Externally provided correct patterns are compared with the neural network’s output during (supervised) training, and feedback is used to adjust the weights until the network has categorized all the training patterns as correctly as possible.

Starting with the output layer, errors between the actual and desired outputs are used to correct the weights for the connections to the previous layer. For any output neuron \(j\), the error (delta) = \((Z_j - Y_j)(df/dx)\), where \(Z\) and \(Y\) are the desired and actual outputs, respectively. Using the sigmoid function, \(f = [1+ \exp(-\]

37
\( x^{[j]−1} \), where \( x \) is proportional to the sum of the weighted inputs to the neuron, is an effective way to compute the output of a neuron in practice. With this function, the derivative of the sigmoid function \( df/dx = f(1 - f) \) and the error is a simple function of the desired and actual outputs. The factor \( f(1 - f) \) is the logistic function, which used to keep the error correction well bounded. The weights of each input to the \( j \)th neuron are then changed in proportion to this calculated error. A more complicated expression can be derived to work backward in a same way from the output neurons through the hidden layers to calculate the corrections to the associated weights of the inner neurons. This complicated method is an iterative approach to solving a nonlinear optimization problem that is very similar in meaning to the one characterizing multiple linear regression.

The learning algorithm includes the following procedures:

1. Initialize weights with random values and set other parameters.
2. Read in the input vector and the desired output.
3. Compute the actual output via the calculations, working forward through the layers.
4. Compute the error.
5. Change the weights by working backward from the output layer through the hidden layers.

This procedure is repeated for the entire set of input vectors until the desired output and the actual output agree within some predetermined tolerance. Given the calculation requirements for one iteration, a large network can take a
very long time to train; therefore, in one variation, a set of cases are run forward and an aggregated error is fed backward to speed up learning. Sometimes, depending on the initial random weights and network parameters, the network does not converge to a satisfactory performance level. In those scenarios, new random weights must be generated, and the network parameters, or even its structure, may have to be modified before another attempt is made. Current research is aimed at developing algorithms and using parallel computers to improve this process.

Fig 1.10: Back Propagation

1.1.2.2.4.1 BACK PROPAGATION ALGORITHM

In most cases, people would consider the Back Propagation network to be the quintessential Neural Net. In reality, Back Propagation is the training or learning algorithm rather than the network itself. The network used is generally for Feed Forward Network or occasionally Multi-Layer Perceptrons.

A Back Propagation network learns by example. You give the algorithm examples of what you want the network to do and it changes the network’s
weights so that, when training is finished, it will give you the required output for a particular input. Back Propagation networks are ideal for simple Pattern Recognition and Mapping Tasks. As just mentioned, to train the network you need to give it examples of what you want the output you want (called the Target) for a particular input as shown in the following Figure 1.11.

**Fig1.11: Back Propagation training set**

So, if we put in the first pattern to the network, we would like the output to be 0 1 as shown in the following Figure 1.12 (a black pixel is represented by 1 and a white by 0 as in the previous examples). The input and its corresponding target are called a *Training Pair*.

**Fig.1.12: Applying a training pair to a network**
Once the network is trained, it will provide the desired output for any of the input patterns. Let’s now look at how the training works.

The network is first initialized by setting up all its weights to be small random numbers – say between –1 and +1. Next, the input pattern is applied and the output calculated. The calculation gives an output which is completely different to what you want (the Target), since all the weights are random. We then calculate the Error of each neuron, which is essentially: Target – Actual Output. This error is then used mathematically to change the weights in such a way that the error will get smaller. In other way, the Output of each neuron will get closer to its Target. The process is repeated again and again until the error is minimal.

Let's do an example with an actual network to see how the process works. We'll just look at one connection initially, between a neuron in the output layer and one in the hidden layer, consider the following figure

![Diagram](https://via.placeholder.com/150)

**Fig.1.13: Single connection learning in a Back Propagation network**
The connection we’re interested in is between neuron A (a hidden layer neuron) and neuron B (an output neuron) and has the weight $W_{AB}$. The diagram also shows another connection, between neuron A and C, but we’ll return to that later.

**VALIDATION**

From a statistical point of view, a Neural Networks represent a class of nonparametric adaptive models. In this framework, an important issue is to evaluate the performance of the model. This can be done by separating the data into two sets: the training set and the testing set. The parameters of the network are computed using the training set. Then learning is stopped and the network is evaluated with the data from the testing set.

**Table 1.2: Comparison between Fuzzy Systems and Neural Networks**

<table>
<thead>
<tr>
<th>Property</th>
<th>Skills</th>
<th>Fuzzy System</th>
<th>Neural Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge Acquisition</td>
<td>Input Tools</td>
<td>Human Experts Interaction</td>
<td>Sample Sets Algorithms</td>
</tr>
<tr>
<td>Uncertainty</td>
<td>Information Cognition</td>
<td>Quantitative and Qualitative Decision Making</td>
<td>Quantitative Perception</td>
</tr>
<tr>
<td>Reasoning</td>
<td>Mechanism Speed</td>
<td>Heuristic Search Low</td>
<td>Parallel Computing High</td>
</tr>
<tr>
<td>Adaption</td>
<td>Fault-Tolerance Learning</td>
<td>Low Induction</td>
<td>Very High Adjusting Synaptic Weights</td>
</tr>
</tbody>
</table>
1.1.2.3 GENETIC ALGORITHMS

The first work on genetic algorithm was begun in the 60’s by John Holland in Michigan University. The first breakthrough in the genetic algorithms was the *Adaptation in Natural and Artificial System* in 1975.

Genetic algorithms are adaptive and robust computational search procedure, modeled on the mechanics of natural genetic systems. They act as a biological metaphor and try to evaluate some of the processes observed in natural evaluation. While evaluation operates on encodings of biological entities in the form of a collection of genesis called a chromosome, GAs operate on string representation of possible solutions in terms of individuals or chromosomes containing the features. The feature value, the string structure and the string structure decoded value in case of GA correspond to the allele, genotype, and phenotype in natural evolution.

1.1.2.4 PROBABILISTIC REASONING

Instead of going back to the history of probability, one can concentrate on the development and improvement of probabilistic computing (PC) and illustrate the way it complements fuzzy computing. Probabilistic computing can be divided into two classes: single-values and interval-values systems.
Bayesian Belief Networks (BBNs), based on the original work of Bayes, are typical example of single-valued probabilistic reasoning systems. They started with approximate methods used in first-generation expert systems, such as MYCIN’s confirmation theory and PROSPECTOR’s modified Bayesian rule and evolved into formal methods for propagating probability values over networks. In general, probabilistic reasoning systems have exponential complexity, before the invention of BBNs, it was customary to avoid such computational problems by making unrealistic, global assumptions of conditional independence. By using BBNs, one can reduce this complexity by encoding domain knowledge as structural information: the presence or lack of conditional dependency between two variables is indicated by the presence or lack of a link connecting the nodes representing such variables in the network topology.

Probabilistic computing provides a way to evaluate the outcome of systems affected by randomness. PC’s basic inferential mechanism-conditioning-allows to modify previous estimates of the system’s outcome based on new evidence.

1.1.2.5 EVOLUTIONARY COMPUTING

Natural intelligence is the product of millions of years of biological evolution. Simulating complex biological evolutionary processes may lead to discover how evolution propels living systems towards higher-level intelligence. Greater focus is thus being paid to evolutionary computing techniques such as Genetic Algorithms (GAs), which are based on the evolutionary principle of natural selection. Immune modeling and Artificial Life are similar disciplines and
are based on the assumption that chemical and physical laws may be able to explain living intelligence.

Most of the AI applications, search techniques are employed heuristically. If a search space is too large for an exhaustive search and it is difficult to identify knowledge that can be applied to reduce the search space, there is no choice but to use other, more efficient search techniques to find less than optimum solutions. The GA is a candidate technique for this purpose; it offers the capacity for population based systematic random searches.

1.1.3 BRIDGE BETWEEN FUZZY LOGIC AND WEATHER FORECASTING

The Neuro Fuzzy model that is used in this research work is ANFIS. Neuro fuzzy systems that implement Takagi-Sugeno type fuzzy inference systems get more accurate results than the approaches that implement neuro fuzzy inference systems of Mamdani type, although it has a higher bigger computational complexity. As a guide line for implementing highly efficient neuro-fuzzy systems they should have the following characteristics such as fast learning, on-line adaptability, self-adjusting with the aim of obtaining the small global error possible and small computational complexity. This will help in improving the accuracy of weather forecasting (Yen wee Khun, 2006).

1.1.4 ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)

The scholar proposes a class of adaptive networks that are functionally equivalent to fuzzy inference systems. The proposed architecture is referred to
as ANFIS, which stands for adaptive network-based fuzzy inference system or semantically equivalently, adaptive neuro fuzzy inference system.

![Fig.1.14: A two-input first-order Sugeno fuzzy model with two rules](image)

**1.1.4.1 ANFIS ARCHITECTURE**

In ANFIS, Takagi-Sugeno type fuzzy inference system is used. The output of each rule can be a linear combination of input variables plus a constant term or can be only a constant term. The final output is the weighted average of each rule’s output. Basic ANFIS architecture that has two inputs $x$ and $y$ and one output $z$ is shown in the following figure.

![Fig.1.15: Equivalent ANFIS architecture](image)
The rule base contains two Takagi-Sugeno if-then rules as follows:

\[ f_1 = p_1 x + q_1 y + r_1 \]

\[ f_2 = p_2 x + q_2 y + r_2 \]

\[ \text{Rule 1: If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \text{ then } f_1 = p_1 x + q_1 y + r_1 \]

\[ \text{Rule 2: If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \text{ then } f_2 = p_2 x + q_2 y + r_2 \]

\[ \text{Figure 1.16: Basic structure of ANFIS} \]

Layer 1:

- \( O_{l,i} \) is the output of the \( i \)th node of the layer \( l \).
- Every node \( i \) in this layer is an adaptive node with a node function
  - \( O_{1,i} = \mu_{A_i}(x) \) for \( i = 1, 2 \), or
  - \( O_{1,i} = \mu_{B_{i-2}}(x) \) for \( i = 3, 4 \)
- \( x \) (or \( y \)) is the input node \( i \) and \( A_i \) (or \( B_{i-2} \)) is a linguistic label associated with this node
- Therefore \( O_{1,i} \) is the membership grade of a fuzzy set \( (A_1, A_2, B_1, B_2) \).
- Typical membership function
• \( a_i, b_i, c_i \) is the parameter set.

• Parameters are referred to as **premise parameters**

**Layer 2:**

• Every node in this layer is a fixed node labeled **Prod**. 

• The output is the product of all the incoming signals.

• \( O_{2,i} = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y), i = 1, 2 \)

• Each node represents the fire strength of the rule

• Any other T-norm operator that perform the **AND** operator can be used

**Layer 3:**

• Every node in this layer is a fixed node labeled **Norm**.

• The ith node calculates the ratio of the ith rule’s firing strength to the sum of all rule’s firing strengths.

\[
O_{3,i} = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2
\]

……………………………..…….(1.21)

• Outputs are called normalized firing strengths.

**Layer 4:**

• Every node i in this layer is an adaptive node with a node function:

\[
O_{4,1} = \overline{w_i}f_i = \overline{w_i}(p_x + q_iy + r_i)
\]

……………………………..…….(1.22)
• $w_i$ is the normalized firing strength from layer 3.
• $\{p_i, q_i, r_i\}$ is the parameter set of this node.
• These are referred to as consequent parameters

**Layer 5:**

• The single node in this layer is a fixed node labeled sum, which computes the overall output as the summation of all incoming signals:

$$overall\ output = O_{5,1} = \sum_i w_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$  

..........(1.23)

**1.1.4.2 HYBRID LEARNING ALGORITHM**

From the proposed ANFIS architecture in the above Figure the output $f$ can be defined as:

..........(1.24, 1.25, 1.26)

Where $p_1, q_1, r_1, p_2, q_2$ and $r_2$ are the linear consequent parameters.

• The ANFIS can be trained by a hybrid learning algorithm.
• In the forward pass the algorithm uses least-squares method to identify the consequent parameters on the layer 4.
In the backward pass the errors are propagated backward and the premise parameters are updated by gradient descent.

**Table 1.3: Two passes in the hybrid learning algorithm for ANFIS**

<table>
<thead>
<tr>
<th></th>
<th>Forward Pass</th>
<th>Backward Pass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Premise Parameters</td>
<td>Fixed</td>
<td>Gradient Descent</td>
</tr>
<tr>
<td>Consequent Parameters</td>
<td>Least-squares estimator</td>
<td>Fixed</td>
</tr>
<tr>
<td>Signals</td>
<td>Node outputs</td>
<td>Error signals</td>
</tr>
</tbody>
</table>

Two passes in the hybrid learning algorithm for ANFIS.

### 1.1.5 AUTO REGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)

Briefly, ARIMA econometric modeling takes into account historical data and decomposes it into an Autoregressive (AR) process, where there is a memory of past events; an Integrated (I) process, which accounts for stabilizing or making the data stationary and ergodic, making it easier to forecast; and a Moving Average (MA) of the forecast errors, such that the longer the historical data, the more accurate the forecasts will be, as it learns over time. ARIMA models therefore have three model parameters, one for the AR(p) process, one for the I(d) process and one for the MA(q) process, all combined and interacting among each other and recomposed into the ARIMA(p,d,q) model.

There are many reasons why an ARIMA model is superior to common time-series analysis and multivariate regressions. The common finding in time series analysis and multivariate regression is that the error residuals are correlated with
their own lagged values. This serial correlation violates the standard assumption of regression theory that disturbances are not correlated with other disturbances. The primary problems associated with serial correlation are:

- Regression analysis and basic time-series analysis are no longer efficient among the different linear estimators. However, as the error residuals can help to predict current error residuals, we can take advantage of this information to form a better prediction of the dependent variable using ARIMA.

- Standard errors computed using the regression and time-series formula are not correct and are generally understand. If there are lagged dependent variables set as the regressors, regression estimates are biased and inconsistent but can be fixed using ARIMA.

Autoregressive Integrated Moving Average or ARIMA (p,d,q) models are the extension of the AR model that uses three components for modeling the serial correlation in the time series data. The first component is the autoregressive (AR) term. The AR (p) model uses the p lags of the time series in the equation. An AR (p) model has the form: \( y_t = a_1 y_{t-1} + \ldots + a_p y_{t-p} + e_t \). The second component is the integration (d) order term. Each integration order corresponds to differencing the time series. \( I(1) \) means differencing the data once. \( I(d) \) means differencing the data d times. The third component is the moving average (MA) term. The MA (q) model uses the q lags of the forecast errors to improve the forecast. An MA (q)
model has the form: \( y_t = e_t + b_1 e_{t-1} + \ldots + b_q e_{t-q} \). Finally, an ARIMA \((p, q)\) model has the combined form: \( y_t = a_t y_{t-1} + \ldots + a_p y_{t-p} + e_t + b_1 e_{t-1} + \ldots + b_q e_{t-q} \).

In interpreting the results of an ARIMA model, most of the significations are identical to the multi-variant regression analysis. However, there are several additional sets of result specific to the ARIMA analysis. The first is the addition of Akaike Information Criterion (AIC) and Schwarz Criterion (SC), which are often used in ARIMA model selection and identification. That is, AIC and SC are used to determine if a particular model with a specific set of \(p, d\) and \(q\) parameters is a good statistical fit. SC imposes a greater penalty for additional coefficients than the AIC but generally, the model with the lowest AIC and SC values should be chosen. Finally, an additional set of results called the autocorrelation (AC) and partial autocorrelation (PAC) statistics are provided in the ARIMA report.

For instance, if autocorrelation AC (1) is nonzero, it means that the series is first order serially correlated. If AC dies off more or less geometrically with increasing lags, it implies that the series follows a low order autoregressive process. If AC drops to zero after a small number of lags, it implies that the series follows a low-order moving average process. In contrast, PAC measures the correlation of values that are \(k\) periods apart after removing the correlation from the intervening lags. If the pattern of autocorrelation can be captured by an auto regression of order less than \(k\), then the partial auto correlation at lag \(k\) will be close to zero. The Ljung-Box Q-statistics and their p-values at lag \(k\) are also provided, where the null hypothesis being tested is such that there is no
autocorrelation up to order \( k \). The dotted lines in the plots of the autocorrelations are the approximate two standard error bounds. If the autocorrelation is within these bounds, it is not significantly different from zero at approximately the 5% significance level.

Finding the right ARIMA model takes practice and experience. These AC, PAC, SC, and AIC are highly useful diagnostic tools to help identify the correct model specification. Finally, the ARIMA parameter results are obtained using sophisticated optimization and iterative algorithms, which means that although the functional forms look like those of a multivariate regression, they are not the same. ARIMA is a much more computationally intensive and advanced economic approach.

**1.1.5.1 ARIMA PROCESSES**

ARIMA processes are mathematical models used for forecasting. ARIMA is an acronym for **AutoRegressive**, **Integrated**, **MovingAverage**. Each of these phrases describes a different part of the mathematical model. ARIMA processes have been studied extensively and are a major part of time series analysis. They were popularized by George Box and Gwilym Jenkins in the early 1970s; as a result, ARIMA processes are sometimes known as Box-Jenkins models. Box and Jenkins (1970) effectively put together in a comprehensive manner the relevant information required to understand and use univariate ARIMA processes. The ARIMA approach to forecasting is based on the following ideas:

1. The forecasts are linear functions of the sample observations;
2. The aim is to find models that provide an adequate description of the observed data with as few parameters as possible. This is sometimes known as the principle of parsimony.

One of the simplest ARIMA models is a first-order autoregressive model. Let \( Y_t \) represent the observed value at time \( t \), and suppose we have observations at times \( 1, \ldots, n \). Then the first-order autoregressive model, or AR(1) model, is

\[
Y_t = c + \phi_1 Y_{t-1} + e_t
\]

where \( c \) and \( \phi \) are both constants, and \( e_t \) represents random error at time \( t \). Thus, the next value in the series is equal to a constant \( c \) plus a multiple of the last value in the series plus some random error. Other observed values of the series can be included in the right-hand side of the equation to give higher-order autoregressive (or AR) processes:

\[
Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + e_t.
\]

Moving average (or MA) processes arise when past errors rather than past observations appear on the right hand side of the equation. For example,

\[
Y_t = c + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q}.
\]

Thus, each observation is considered a linear function of past errors. An autoregressive moving average (or ARMA) process occurs when a mixture of past errors and past observations occur on the right hand side of the equation. An autoregressive integrated moving average (or ARIMA) process occurs when
we study the difference between consecutive observations rather than the observations themselves.

There are also ARIMA processes designed to handle seasonal time series, and vector ARIMA processes designed to model multivariate time series. Other variations allow the inclusion of explanatory variables. ARIMA processes have been a popular method of forecasting because they have a well-developed mathematical structure from which it is possible to calculate various model features including prediction intervals, the autocorrelation function, and the spectral density function. Prediction intervals, in particular, are a very important feature of forecasting as they enable forecast uncertainty to be quantified.

1.2 MATLAB TOOLBOX

In this research work two MatLab toolboxes (anfisedit) are used which are explained in brief in further subsections.

1.2.1 ANFIS EDITOR (ANFISEDIT)

In this section, ANFIS Editor GUI in the MatLab toolbox is discussed whose syntax is $\text{anfisedit}$

In fuzzy inference GUIs, it is required to manually choose the shape of the membership functions depending upon the parameters. By changing these parameters there will change in the shape of the membership function. In some modeling situations, there is no way to decide the shape of membership functions simply from looking at data.
Rather than choosing the parameters associated with a given membership function arbitrarily, these parameters could be chosen from the input/output data. In such cases, the Fuzzy Logic Toolbox *neuro-adaptive* learning techniques incorporated in the *anfis* command may be used.

**ANFIS** derives its name from *Adaptive Neuro-Fuzzy Inference System*. Using a given input/output data set, the toolbox function *anfis* constructs a Fuzzy Inference System (FIS) whose membership function parameters are adjusted using either a back propagation algorithm alone or in combination with a least squares type of method called hybridization. The *anfis* function can be accessed either from the command line or through the ANFIS Editor GUI. However the functionality of the command line function *anfis* and the ANFIS Editor GUI is similar, they are used somewhat interchangeably in this discussion.

### 1.2.2 TRAINING ADAPTIVE NEURO FUZZY INFERENCE SYSTEMS USING THE ANFIS EDITOR GUI

To start the GUI, type the following command at the MATLAB prompt:

```
$ anfisedit
```

The ANFIS Editor GUI window will open and have to perform the following tasks:

A. Loading, Plotting, and Clearing the Data

B. Generating or Loading the Initial FIS Structure

C. Training the FIS

D. Validating the Trained FIS
1.2.3 LOADING, PLOTTING, AND CLEARING THE DATA

In order to load an initial input data set, the following steps are used:

• First of all, the data Type is specified.

• Then the input data is selected from a file saved in the computer or already loaded in the MatLab workspace (as shown in figure 1.17)

![Input Data as variable name vizag_test_data](image1)

**Fig. 1.17: Input Data as variable name vizag_test_data**

Then by clicking Load Data, data is loaded as input in anfis editor. After loading the data, it will be displayed in the plot. The training and testing data are shown in blue as *circles and dots* respectively as shown in figure 1.18

![Training and Testing data Graph as circles and pluses](image2)

**Fig. 1.18: Training and Testing data Graph as circles and pluses**
To clear a specific data set from the GUI:

- In the Load data area, the data Type of which we want to clear the data, is selected.

- Then by clicking ‘Clear Data’ the data, training, testing or checking data will be cleared from anfis editor. However it will remain in workspace.

This action also removes the corresponding data from the plot.

1.2.4 GENERATING OR LOADING THE INITIAL FIS STRUCTURE

After loading the data, there is need to describe an initial FIS model structure. It can be done by either of the following ways:

- Load a previously saved Sugeno-type FIS structure from a file on disk

- Or load an FIS from MatLab workspace.

Or generate the initial FIS model by choosing one of the following partitioning techniques:

1. Grid partition— It generates a single-output Sugeno-type FIS by using grid Partitioning on the data.

2. Subtractive Clustering — it generates an initial model for ANFIS training by first applying subtractive clustering on the data.

By clicking the ‘Structure’ a graphical representation of the initial FIS model structure can be seen which is shown in figure 1.19.
Fig. 1.19: Structure of ANFIS Model with three inputs and one output

1.2.5 TRAINING THE FIS

After loading the training data and generating the initial FIS structure, training of FIS can be started. The following steps show how to train the FIS:

- In Optim. Method, choose hybrid or back propagation as the optimization method.

The optimization methods train the membership function parameters to emulate the training data. The hybrid optimization method is a combination of least-squares and back propagation gradient descent method.

Enter the number of training Epochs and the training Error Tolerance to set the stopping criteria for training. The training process stops whenever the maximum epoch number is reached or the training error goal is achieved.
• By clicking train now to train the FIS. This action adjusts the membership function parameters and displays the error plots shown in figure 1.20.

1.2.6 VALIDATING THE TRAINED FIS

After the FIS is trained, validate the model using a testing that differs from the one used to train the FIS. To validate the trained FIS:

• Select the validation data set and click Load Data.

• Click Test Now. This action plots the test data against the FIS output (shown by red dots in figure 1.21).

Fig. 1.20: Training the ANFIS model with 100 epochs - example
1.3 TERMINOLOGIES USED IN WEATHER FORECASTING

1.3.1 WEATHER

It refers to the physical state of atmosphere (envelop of gasses surrounding the earth) at a given time over a place. Weather may be defined as instantaneous condition of atmosphere over a place. It is highly variable and constantly changing from hour to hour or day to day. The entity that determines the state of weather is called as parameter/element of weather such as Temperature, Relative Humidity, Evaptranspiration, Rainfall etc.. These weather elements are not independent rather they are correlated to each other.

1.3.2 METEOROLOGY

Meteorology is the branch of science, which study day-to-day atmospheric conditions. It may be defined as the physics of atmosphere or the science, which
studies weather. Meteorology is the science, which study the physical processes occurring in the atmosphere, those produce weather.

1.4 CONSTRAINTS IN WEATHER FORECASTING

Over the last decade, weather forecast has improved greatly in accuracy because of the availability of satellite data, numerical models and real-time computer system. These forecast tools are highly impressive, but have the following limitations.

1.4.1 AREA CONSTRAINT

Till now, weather forecast is issued for relatively large area covering countries and major cities. However local weather can vary considerably within the area to which weather is being forecasted. Local area which is very flat reduces wind disturbance and absorb rainwater deeply into the soil. Depending on wind direction and season, local atmosphere may be hydrated and moderated. Because every region is exceptional to some degree, it may be believed that computer-assisted forecasting could be improved if, as a last step, the process were refined by an expert system that could take better account of local climatic modifiers. For this purpose soft computing approach mainly Neuro-Fuzzy may be used in order to discard such type of problems.

1.4.2 TIME CONSTRAINT

Available forecasts are based on observations that may be eight hours old and dataset for most of Automatic Weather Station (AWS) may be Because of this lag time, it may be concluded that data obsolescence will continue to be a
compromising factor for the foreseeable future and this lag will continue till the validation of observed and predicted value resulting a big difference in measurement of errors.

In this situation, it is required that available public forecasts could become more reliable if, they will be adjusted by an intelligent system that could take better account of current and changing local conditions.

1.4.3 SUBJECTIVITY

Discovering and understanding the dynamic phenomena of weather, to accurately predict different weather events, has been an integral component of our study. The weather data, being inherently fuzzy in nature, requires highly complex processing based on human observations, satellite photography, followed by computer simulations. This is further combined with an understanding of the principles of global and local weather dynamics.

1.4.4 GLOBAL WARMING EFFECT

Weather forecasting has been a very important issue through last two decades. As we know due to global warming and human interference in nature has made the predication of climate, a totally unpredictable. We need to have a new approach which can predicate about the trend of weather- rain, Min temperature, Max temperature, Bright Sun Shine Hours, Humidity etc.

1.4.5 CHANGE IN ATMOSPHERIC CONDITIONS

Atmospheric conditions change rapidly. Hence, even with sophisticated computer models, satellite images and monitoring stations across most of the planet,
forecasts may not be exact. That is, sometimes the weather is more predictable than at other times.

1.5 WEATHER ELEMENTS

Any entity that controls the weather is called as weather element or weather parameter. Given below are various parameters which individually (or in combination) control the weather:

- Solar radiation
- Temperature
- Air Pressure
- Wind
- Sunshine
- Humidity
- Cloudiness
- Precipitation etc.

All these elements are highly variable and constitute the weather. Day to day or hour to hour changes in weather is mainly the result of variations in the amount, intensity and distribution over earth of weather elements listed above.

1.6 IMPORTANCE OF WEATHER ELEMENTS

The weather elements constitute a physical environment in which an organism or a plant grows. Success or failure of a bio-system depends upon the weather conditions. The rate of growth and development of a bio-system largely depends
upon its ambient temperature and moisture. The management and execution of agricultural farm practices is also weather dependent. The maximum efficiency of farm practices can be achieved with the knowledge of forecasting the weather parameters in anticipation.

1.7 EFFECTS OF WEATHER ELEMENTS

1.7.1 EFFECT OF TEMPERATURE

Increase/decrease in temperature are the major concern of changing the climate. Our climate is changing, both naturally and due to human exploitation. There is already undeniable evidence that animals, birds and plants are being affected by climate change and global warming in both their distribution and behavior. Unless greenhouse gas emissions are severely reduced, climate change could cause a quarter of land animals, birdlife and plants to become extinct.

Climate variability and change affects birdlife and animals in a number of ways; birds lay eggs earlier in the year than usual, plants bloom earlier and mammals are come out of hibernation sooner. Distribution of animals is also affected; with many species moving closer to the poles as a response to the rise in global temperatures. Birds are migrating and arriving at their nesting grounds earlier, and the nesting grounds that they are moving to are not as far away as they used to be and in some countries the birds don’t even leave anymore, as the climate is
suitable all year round. A sea level rise of only 50cm could cause sea turtles to lose their nesting beaches.

Humans have already destroyed many of the natural migrations of animals. There are wide spread of many diseases among humans due to the change in temperature.

Changing rainfall patterns are causing dams to be erected in some areas of our planet. Severe droughts in many countries which are now facing reduced crop production and major drinking water shortages.

1.7.2 EFFECT OF MOISTURE

The presence of moisture in atmosphere is termed as humidity. Its presence in the atmosphere plays a significant role. An abundance of moisture results in a rich natural flora and makes possible a wide choice of crops. Deficiency of moisture, on the other hand, permits only narrow range of potential crops and is accompanied by hazards to efficient crop production. Increasing relative humidity decreases the rate of transpiration. Reduction in transpiration reduces the translocation of food material and uptake of nutrients. Humidity also affects the water requirement through evapotranspiration. High humidity with high temperature favors the outbreak of pest and disease. Too much water may be harmful to plants just as too little.
1.7.3 EFFECT OF RAINFALL

The water source such as rivers, tanks and wells, which supply water for irrigation also depend on the rain. Rain occurring during the flowering and grain filling period is very harmful. After the spray it reduces the efficiency of agrochemical. Heavy rain induces soil erosion and leaching of nutrient. Deficient rain causes drought and limits the crop growth and production.

1.8 WEATHER ELEMENTS USED IN THE RESEARCH: DEFINITION AND MEASUREMENTS

1.8.1 TEMPERATURE

Temperature is a measure of the heat concept of a body. It shows the intensity of heat energy or degree of hotness or coldness. The temperature may be defined as the intensity aspect of heat energy. The temperature measures the average kinetic energy of the molecules. It is the measures of sensible heat energy of a system. Most widely used scales for the measurement of temperature is:

**Celsius scale:** This scale is named for the Swedish astronomer Anders Celsius. It is internationally accepted for measurement of temperature. The boiling point of water is 100 °C and freezing point of water (Triple point) is 0 °C.

**Fahrenheit scale:** The Fahrenheit scale is used for temperature as evident from historical records of temperatures in English speaking countries. The freezing point (triple point) of water is 0°F and boiling point of water is 212 °F.

*In this research data used for temperature is in °C.*
1.8.1.1 INSTRUMENTS FOR THE MEASUREMENT OF AIR TEMPERATURE

- **Thermometer:** The most common instrument for measuring temperature is mercury in glass thermometer. Alcohol-in-glass thermometer is also used for measuring temperature.

- **Minimum thermometer:** It is an alcohol in glass thermometer with a dark dumb bell shaped index placed in the bore below top of the alcohol column. It is used to measure minimum temperature.

- **Maximum thermometer:** It is mercury in glass thermometer with a constriction in bore near bulb. It is used for measuring maximum temperature.

1.8.2 ACTUAL VAPOUR PRESSURE

**Pressure** of air at a given place is the force exerted against a surface by continuous collision of gas molecules. Pressure may be defined as the force per unit surface area.

\[ P = \frac{F}{A} \] (1.30)

Where,

- \( P \) = Pressure
- \( F \) = Force
- \( A \) = Area upon which force is exerted.
**Air pressure** is the force exerted in all directions as a result of weight of all the air above it. Force exerted by mass of a column of air above a given point is called air pressure.

\[
P = \frac{\rho \cdot h \cdot x \cdot A \cdot g}{A} = \rho \cdot h \cdot x \cdot g
\]

……………………..(1.31)

Since

where

\[
F = \frac{\text{Mass} \times \text{acceleration}}{\text{Area}}
\]

\[
\text{Mass} = \rho \cdot x \cdot V = \rho \cdot x \cdot h \cdot A
\]

……………………..(1.32, 1.33, 1.34)

*Where,*

\[\rho = \text{Density of liquid (Mercury}=1.3\times10^3 \text{ kg m}^{-3})\]

\[h = \text{Height of liquid (Mercury height} =76 \text{ cm})\]

\[g = \text{Acceleration due to gravity (980 cm s}^{-2})\]

**Vapour Pressure** is the partial pressure exerted by water vapour in air. If we sum up all partial pressures caused by each atmospheric constituent, then it is called atmospheric pressure (P).

\[
P = \rho \text{ (N}2\text{)} + \rho \text{ (O}2\text{)} + \rho \text{ (H}2\text{O)} + \rho \text{ (Ar)} + \rho \text{ (Co}2\text{)} + ....... .............(1.35)
\]

*Where,*

\[\rho \text{ (N}2\text{)} = \text{Partial pressure due to nitrogen}\]

\[\rho \text{ (O}2\text{)} = \text{Partial pressure due to oxygen}\]

\[\rho \text{ (H}2\text{O)} = \text{Partial pressure due to water vapour}\]

\[\rho \text{ (Ar)} = \text{Partial pressure due to argon}\]

\[\rho \text{ (CO}2\text{)} = \text{Partial pressure due to carbon dioxide}\]
**Actual Vapour Pressure (AVP)** is the partial pressure exerted by water vapour in given volume of air at a given temperature. Actual vapour pressure is called as saturated vapour pressure when air is saturated. It may also be defined as the pressure exerted by water vapour actually present in air. It is the saturated vapour pressure at dew point.

**Most widely used units for the measurement of Air Pressure is:**

- **Height of Hg** i.e. inches, cm or mm of Hg.
- **Bar**: It is the force equal to 106 dynes per square centimeter.
- **Pascle (Pa)**: It is the force of 1 Newton per square meter = N/m^2\n- **Atmosphere** = 1 atm = 29.92 inches or 76 cm or 760 mm of Hg

In this research Data of Actual Vapor Pressure has the unit mm of Hg

**1.8.2.1 Instruments for the Measurement of Air Pressure**

- **Anemometer**: The instrument which is used to measure wind velocity is called as Anemometer e.g. wind cup anemometer, automatic anemometer etc.
- **Anemograph**: The instrument which gives a continuous record of wind velocity with time on a chart.
- **Wind vane**: The instrument which is used to measure wind direction is known as wind vane.
- **Aero-vane**: The instrument, which measures both wind velocity and wind direction.
1.8.3 HUMIDITY AND EVAPORATION

Humidity represents the presence of water vapour in air at a time and place. Water vapour constitutes only a small proportion of air which varies from nearly zero to maximum of 4 per cent by volume. Water changes its states: solid, liquid and gas from one to another through the process of melting, evaporation, condensation, sublimation and freezing.

Evaporation is a process by which water in liquid form changes to gaseous form. It may be defined as the water loss in vapour form from a soil/water surface. The energy used in this process is called as latent heat. The energy of 540 cal/g of water is used in this process.

Absolute Humidity (AH) may be defined as the mass of water vapour per unit volume of air or it is the ratio of mass of water vapour to the volume of air. The unit is g/cm³.

\[ AH = \frac{\text{Mass of water vapour (mv)}}{\text{volume of the air (V)}} = \frac{mv}{V} \] ..........................(1.36)

Relative Humidity (RH) is defined as ratio of the mass of water vapour present in air at a temperature to the mass of maximum water vapour which that air can hold at the same temperature. It may also be defined as the ratio of actual vapour pressure to the saturated water vapour pressure.
Where,

**AVP** = Actual vapour pressure

**SVP** = Saturated vapour pressure

**Evapotranspiration (ET)** is the water loss in vapour form from a water surface or land surface is called as evaporation. While, the loss of water in vapour form from plants is known as transpiration. A combination of evaporation and transpiration is termed as evapotranspiration. Evapotranspiration is combined loss of water in vapour form soil and plants surface. In a crop field, the rate of water loss through evaporation is higher during initial phase of crop development and water loss through transpiration is higher during latter period of crop cycle. The rate of evapotranspiration is highest during maximum ground cover by canopy i.e. maximum leaf area period.

### 1.9 WEATHER FORECASTING

Weather forecasting is foretelling the coming weather in advance. It may be defined as advance information about the probable weather conditions for few days to follow. So the time for which weather forecast is made is also important. It can be termed as lead time. The accuracy of a forecast decreases with
increasing lead period and decreasing area under forecast. Different weather phenomena can be forecasted with different lead time.

1.9.1 EXPERIMENTAL WEATHER FORECAST AGENCIES

Experimental forecast for the year 2000-2006, weather based on the Indian Meteorological Department (IMD)’s dynamical forecast system was generated. For this purpose, observed sea surface temperature data have been used. In addition, IMD has also taken into account the experimental forecasts prepared by the national institutes like Indian Institute of Tropical Meteorology, Pune, Indian Institute of Science, Bangalore, Space Applications Centre, Ahmadabad, National Aerospace Laboratories (NAL), Bangalore and Centre for Mathematical Modeling and Computer Simulation (CMMACS), Bangalore, National Centre for Medium Range Weather Forecasting (NCMRWF), Noida and operational/experimental forecasts prepared by international institutes like the National Centers for Environmental Prediction (NCEP), USA, International Research Institute for Climate and Society (IRI), USA, Meteorological Office, UK, the European Center for Medium Range Weather Forecasts (ECMWF), UK and the Experimental Climate Prediction Center (ECPC), USA.

1.9.2 TYPES OF WEATHER FORECASTS

On the basis of advance time span, forecast can be grouped into three categories as shown in figure 1.22.
1.10 INTRODUCTION TO PROPOSED RESEARCH WORK

Since last decade soft computing approach has emerged as an advanced technology with successful applications in many fields. Weather forecasting is the application of soft computing which combines science and nature to predict the state of weather for future at a given location.

Soft computing is an innovative approach to construct computationally intelligent systems that are supposed to process human-like expertise within a specific domain, adapt themselves and learn to do better in changing environments and explain how they make decisions. The weather forecasting model based on soft computing is easy to implement and produces desirable forecasting result by training on the given dataset (Abraham et al, 2004).
Neuro Fuzzy is a combination of Artificial Neural Network and Fuzzy Logic in such a way that Neural Network learning algorithms are used to determine the parameters of Neuro Fuzzy. That is why Neuro Fuzzy is well suited to the problem of weather forecasting and improve the weather forecasting accuracy.

In this research work, Neuro Fuzzy approach is applied on various weather forecasting parameters with different data sets and the result is compared with statistical method. Result shows that Neuro Fuzzy approach is much adaptive on all data sets and provides better approximation to weather prediction which is taken as output variable.

1.10.1 AIM OF RESEARCH

The aim of this research is to find out how well the proposed soft computing models are able to understand the behavior of input parameters so that weather prediction can be made. This would help us to anticipate the weather with maximum degree of confidence level in coming season.

1.10.2 SIGNIFICANCE OF RESEARCH

Weather forecasting greatly contributes towards making short term adjustments in daily activities, which minimize losses due to adverse weather conditions. Weather forecast is very important because it can be used to protect life and property. Forecasts based on temperature, wind speed and relative humidity are very important attributes in agriculture sector as well as many industries which largely depend on the weather condition. For example, heavy rains and flood may cause disaster, an extended period of dry weather may cause drought and
same can be think about for pilots, fishermen, mountain climbers etc. Therefore, having accurate weather forecasting information may allow these stakeholders to make good decision on managing their activities.

1.10.3 PROBLEM DEFINITION

The objective of this work is to analyze, process and normalize the metrological weather parameters which are major responsible for weather prediction. Then to develop an approach based on the proposed soft computing models (ANFIS) and traditional data extracting techniques (ARIMA). Later to evaluate Moving Average Error (MAE), Root Mean Square Error (RMSE) and $R^2$ (Coefficient of determination) using the soft computing models and traditional data extracting techniques. Finally to compare the results between the intelligent systems by using traditional data extracting techniques (ARIMA) and soft computing techniques (ANFIS) to predict weather forecasting more accurately. The final goal of this work is to suggest an improved methodologies for potential knowledge workers of the intelligent system in their decision making process. The aim of my study is to exploit the tolerance for imprecision, uncertainty, appropriate reasoning and partial truth in order to achieve traceability, robustness, and low cost solutions by different kinds of soft computing methodologies available for improvement of data mining system to avoid inconsistency in knowledge presentation which is provided to end users in the form of different kinds of patterns. The intelligent system projects are very expensive projects over other traditional information systems, therefore it is
highly desirable that they produce the expected results and help in decision making as an easy process.

1.10.4 NEED FOR STUDY

Weather is generally referred as the atmospheric conditions that comprise the state of the atmosphere in terms of temperature and wind and clouds and precipitation. It is of great interest to people everywhere, from meteorologists, the scientists, farmers, sailors etc., a weather forecast is simply a scientific estimate of future weather condition. Weather forecasting involves a combination of computer models, observations and knowledge of trends and patterns. So long, the techniques used for weather forecasting is probability distribution, curve fitting, normalization which tend to unpredictable results.

Studies have shown that accuracy in the presentation of knowledge was ambiguous and vagueness because of that the decision making may be leads to wrong direction. Because of increasing expectations of new customers, the functionalities which are provided by the system sometimes may not be satisfied. That’s why there is a challenge to all researchers to develop a new shape for existing systems and make flexible systems for forth coming business community people. Here I will bit concentrate on this area in my research.

1.11 ORGANIZATION OF THE THESIS

The entire research work is divided into five chapters along with the references used and list of publications.
The first chapter is introductory chapter which describe background substance for the research problem. It describes in detail various soft computing approaches including fuzzy logic, fuzzy inference systems, neuro fuzzy approaches and then Adaptive Neuro Fuzzy Inference Systems, which is the basic model used in the research work done to forecast weather. This chapter also provides bridge between fuzzy logic and weather forecasting. Then the chapter contains a brief idea of statistical model ARIMA. All MatLab Toolboxes that are used in doing this research work has been explained very clearly in order to present a very clear idea of how forecasting is done. In this chapter itself Brief information of Weather Terminologies, its elements and there measurement instruments is provided so that the raw data collection knowledge is enhanced. Then a brief detail on the weather details of Visakhapatnam is given.

The second chapter is endowed with review of literature available in the area of weather forecasting. In short, this chapter discussed about the literature review of weather forecasting, ANFIS model, ARIMA model to provide a better understanding to choose the best technique to forecast weather before proceeding to the research methodology used and experiment and implementation process. Besides that, gaps of weather forecasting which are a big hindrance in developing an appropriate model have also been discussed in order to use as a reference to this thesis report. Based on these gaps we have decided some objectives to be achieved in this research which are explained in
last section of this chapter. On the basis of these objectives, the Research Methodology will be described in next chapter.

The third chapter describes the research methodology that is used to model the ANFIS architecture and ARIMA model in order to forecast weather. This chapter includes Methods of data collection, Primary source of data, setup of Neuro fuzzy model and their hybridization learning algorithm. Moreover means of error performance that is MAE, RMSE, and co-efficient of determination are also defined in this chapter.

The forth chapter is dedicated to all the Experiments done in MatLab, SPSS, Excel Sheet along with discussion of results of all these experiments on the basis of minimum error rate. First of all the values of all input and output variables are shown in tabular form to have a brief idea regarding the raw data. Then the data is analyzed and processed to make it normalized. This section includes only those experiments which are successful to forecast the weather. Last part of this chapter contains result discussion and comparison. In this part result of all the successful Models is shown with their output and conclusion. At last, comparison between Statistical Model and ANFIS Model for the minimum error is done in order to find the final conclusion and essence of the research.
The *fifth chapter* is the essence of the thesis which is the conclusion part. It provides in detail the conclusion, contribution of this whole study in various fields and scope for further research.

*At last*, references are given.
2.1 HISTORICAL BACKGROUND OF WEATHER FORECASTING

Over 70 per cent of the world's population livelihood depends upon the weather, of which the Asia's weather forecasting is the largest. Accurate predictions of the weather, at least a season in advance, are therefore crucial for the farmers, mariners, investors etc., Further more, the Asian weather forecasting is a key component of the earth's climate system, having important tele-connections with global weather and climate (Walter Maner, 1997).

Following the Great Indian Drought of 1877, H.F. Blanford, who had established The India Meteorological Department in 1875. Later, in the early part of the 20th century, Sir Gilbert Walker initiated extensive studies of global tele-connections which led him to the discovery of Southern Oscillation. Walker introduced, for the first time, the concept of correlation for long-range forecasting of the Asian weather and his findings are relevant even today.

Generally, there are two methods that are used in weather forecasting one is empirical approach and other is dynamical approach (Lorenz, 1969). The empirical approach is based upon the occurrence of analogues and is often referred to by meteorologists as analogue such as changes in barometric pressure, current weather conditions, sky condition to determine the future
conditions (Ozelkan, 1996). This approach normally is useful for predicting local-scale weather if recorded cases are very large in number.

Dynamical approach is based upon equation and forward simulations of the atmosphere and is often referred to as computer modeling which involves pattern recognition skills, knowledge of model performance and knowledge of model biases (Lorenz, 1969). This approach is normally useful for modeling large-scale weather phenomena and may not predict short-term weather efficiently. Most of the weather forecasting systems are combined techniques of empirical approach and dynamical approach. However, not much attention has been paid to the use of soft computing in weather forecasting.

Weather forecast systems are among the most complex equation systems that computer has to solve. A great quantity of data, coming from satellites, ground stations and sensors located around our planet send daily information that must be used to foresee the weather situation in next hours and days all around the world. Weather reports give forecast for next 24, 48 and 72 hours for wide areas (Pasero, 2004). Weather forecasts provide critical information about future weather. There are various techniques involved in weather forecasting, from relatively simple observation of the sky to highly complex computerized mathematical models (M. Tektas, 2010).

2.2 APPROACHES FOR WEATHER FORECASTING

In this chapter, the literature survey on Fuzzy Logic (FL), Neural Network (NN) and Neuro-Fuzzy (NF) systems are collectively called soft computing
techniques in solving real world problems given. In the first section, studies on FLSs, in the second section cases in which NNs were applied are discussed. In the last section, the scholar focus on ANFIS and ARIMA models and its development stage and implementation are also shown.

2.2.1 FUZZY LOGIC SYSTEMS

In 1965, the concept of fuzziness was first proposed by Zadeh. He aimed to describe complex and complicated systems using fuzzy approximation and introduced fuzzy sets. “Generally, fuzzy logic can be considered as a logical system which were developed by using if-then rules that provide a model for modes of human reasoning that are approximations rather than exact” (Rutkowska 2002).

Fuzzy Logic, another kind of soft computing technique, it can also be of great use in weather or atmospheric data analysis and prediction. Being capable of dealing with linguistic labels or linguistic variables. In analyzing atmospheric variables, fuzzy computing methodology takes place an important role.

Fuzzy Logic in detecting severe updrafts. Fujibe (1998) classified the pattern of precipitation at Honshu with fuzzy C-means method. Galambosi et al (1999) investigated the effect of ENSO and macro circulation patterns on precipitation at Arizona using Fuzzy Logic. Vivekanandan et al (1999) developed and implemented a fuzzy logic algorithm for hydrometer particle identification that is simple and efficient enough to run in real time for operational use. Fuzzy logic systems had found successful applications in wide variety of fields such as: automatic control, pattern recognition, signal processing, expert systems, communication, system identification and time series prediction (Czogala and Leski 2000).

Hansen (2000) applied fuzzy k-NN weather prediction system to improve the technique of persistence climatology by achieving direct, efficient and expert like comparison of past and present weather cases. Shao (2000) established fuzzy membership functions, based on cloud amount, cloud type, wind speed and relative humidity, to compose a fuzzy function of weather categorization for thermal mapping. Liu and Chandrasekar (2000) developed a fuzzy logic and neuro-fuzzy system for classification of hydrometer type based on polarimetric radar measurements, where fuzzy logic was used to infer hydrometer type, and the neural network learning algorithm was used for automatic adjustment of the parameters of the fuzzy sets in the fuzzy logic system according to prior knowledge.

“As the complexity of a system increases, our ability to make precision and yet significant statements about the behavior diminishes until a threshold is reached beyond which precise and significance become almost mutually exclusive characteristics.” (Zadeh 1971).

2.2.2 NEURO COMPUTING

A neural network is a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain (Tsoukalas and Uhrig 1997). The formal realization that the brain in some way performs information processing tasks was first spelt out by McCulloch and Pitts (1943). They represented the activity of individual neurons using simple threshold logic elements, and showed how networks made out of many of these units interconnected could perform the logical operations. Rosenblat (1959) developed the concept of perceptron, a generalization of the McCulloch and Pitts concept of the functioning of the brain, by adding learning (Lisboa 1992). These studies were the initiations of NNs.
Hu (1964) initiated the implementation of Artificial Neural Network, an important Soft Computing methodology in weather forecasting. McCann (1992) developed Artificial Neural Network models to give 3-7 hr forecast of significant thunderstorms on the basis of surface based lifted index and surface moisture convergence. The two neural networks produced by them were combined operationally at National Severe Storms Forecast Center, Kansas City, Missouri to produce a single hourly product and was found to enhance the pattern recognition skill.

Mohandes et al (1998) applied Artificial Neural Network for prediction of wind speed. Gardner and Dorling (1998) discussed the proficiency of Multilayer Perceptron as a suitable model for atmospheric prediction. Lee et al (1998) applied Artificial Neural Network in rainfall prediction by splitting the available data into homogeneous subpopulations. Wong et al (1999) constructed fuzzy rule bases with the aid SOM and back propagation neural networks and then with the help of the rule base developed predictive model for rainfall over Switzerland using spatial interpolation. Bruton et al (2000) developed ANN models for estimating daily pan evaporation. The results were compared with those of multiple linear regression and Priestly-Taylor model and they found that the ANN model provided the highest accuracy.


Maqsoos et al (2002) established the usefulness of Artificial Neural Network in atmospheric modeling explained its potential over conventional weather prediction model. Rashidi and Rashidi (2004) developed an Artificial Neural Network in the form of Multilayer perceptron to predict the solar activity,
which is noticeably important for forecasting space weather. Chaudari and Chattopadhaya (2005) developed an Artificial Neural Network model for prediction of some surface parameters during pre-monsoon thunderstorms over northeastern part of India.

2.2.3 NEURO FUZZY SYSTEMS

In most fuzzy systems, fuzzy rules were obtained from the human expert. However, every expert does not want to share his knowledge and there is no standard method that exists to utilize expert knowledge. As a result, ANNs were incorporated into fuzzy systems to be able to acquire knowledge automatically by learning algorithms. The learning capability of the NNs was used for automatic fuzzy if then rules generation (Czogala and Leski 2000).

The connection of fuzzy systems with an ANN is called neuro-fuzzy, NF, systems. Like in NNs where knowledge is saved in connection weights, it is interpreted as fuzzy if then rules in NF systems. The most frequently used NN in NF systems is radial basis function neural network, RBFNN in which each node has radial basis function such as Gaussian and Ellipsoidal. Their popularity is due to the simplicity of structure, well-established theoretical basis and faster learning than in other types of NNs. Also, there are many developed fuzzy neural networks (FNN) as NF algorithms in literature. Adaptive network based fuzzy inference system, ANFIS, is one of them. It is type of RBFNN.

Jang (1992) proposed to use the ANFIS architecture to improve the performance of the fuzzy controllers. The performance of the fuzzy controller
relies on two important factors: knowledge acquisition and the availability of human experts. For the first problem, Jang proposed the ANFIS to solve the automatic elicitation of the knowledge in the form of fuzzy if then rules. For the second problem, that is how the fuzzy controller is constructed without using human experts; a learning method based on a special form of gradient descent (back propagation) was used. The proposed architecture identified the near optimal membership functions and the other parameters of a controller rule base for achieving a desired input-output mapping.

In 1992, Uchikawa et al. presented a fuzzy modeling method using fuzzy neural networks, FNNs, with the back propagation algorithms. They proposed three types of NN structures of which the connections weights have particular meanings for getting fuzzy inference rules for tuning membership functions. These structures are categorized into FNNs and these different types FNNs realize three different types of reasoning. Rao and Gupta (1994) described the basic notions of biological and computational neuronal morphologies and the principles and architectures of FNNs. Two possible models of FNN were given. In first one, the fuzzy interface provides an input vector to a multilayered network in response to linguistic statements. Then the NN can be trained to yield desired output. In the second scheme, a multilayered NN drives the fuzzy inference mechanism. It was pointed out that using FNN approaches having the potential for parallel computation could eliminate the amount of computation required.
In another paper, Uchikawa et al. (1995) presented a new design method of adaptive fuzzy controller using linguistic rules of fuzzy models of the controlled objects. FNNs identify fuzzy models of nonlinear systems automatically with the back propagation algorithm in this method. Authors also presented a rule-to-rule mapping method for describing the behavior of fuzzy dynamical systems. Using this methodology, first, the control rules are modified by considering rule-to-rule transitions. After that, designed controller was implemented with another FNN. The adaptive tuning of the control rules was done using the fuzzy model of the controlled object by utilizing the derivative value from the fuzzy model. A second order system was simulated to show the feasibility of the proposed design method.

2.2.4 ARIMA MODEL

ARIMA processes have been studied extensively and are a major part of time series analysis. They were popularized by George Box and Gwilym Jenkins in the early 1970s; as a result, ARIMA processes are sometimes known as Box-Jenkins models. Box and Jenkins (1970) effectively put together in a comprehensive manner the relevant information required to understand and use univariate ARIMA processes.

The method has proved remarkably robust to a wide range of time series, and is optimal for several processes including the ARIMA(0,1,1) process (Chatfield et al. 2001). There is also a multiplicative version of the Holt-Winters method, and damped trend versions of both Holt’s linear method and the Holt-
Winters method (Makridakis et al. 1998). None of these methods are explicitly based on underlying time series models, and as a result the estimation of parameters and the computation of prediction intervals are often not done. However, all the above methods have recently been shown to be optimal for some state space models (Hyndman et al. 2008), and maximum likelihood estimation of parameters, statistical model selection and computation of prediction intervals is now becoming more widespread.

2.2.5 GAPS IN EXISTING RESEARCH

After a comprehensive study made on the existing literature, a lot of limitations/gaps have been found in the area of weather forecasting:

- Majority of work reported for weather forecasting problems has been done using various statistical methods like Curve Fitting, Regression Analysis etc. which have their own limitations. Hence a more attention is required towards a new approach for weather forecasting.

- Most of the works reported on weather forecast has paid focus on objective weather forecasting system, not so much attention has been given for the subjective weather forecasting which produces more accurate results.

- There is limited work towards hybridization of Neural Network and Fuzzy System in weather forecasting. Hence more emphasis is required towards it.
2.3 OBJECTIVES OF PROPOSED RESEARCH WORK

1. To identify and characterize (Analyze, process and normalize) the metrological weather parameters which are major responsible for weather prediction.

2. To develop an approach based on the proposed soft computing models (ANFIS) and statistical techniques (ARIMA).

3. To evaluate Moving Average Error (MAE), Root Mean Square Error (RMSE) and $R^2$ (Coefficient of determination) using the soft computing models and statistical techniques.

4. To assessment the feasibility of the proposed soft computing approach for weather forecasting and compare the results(MAE, RMSE, $R^2$) based on the performance between the intelligent systems by using traditional data extracting techniques(ARIMA) and soft computing techniques(ANFIS) to predict weather forecasting more accurately.

2.3.1 PROPOSED APPROACH

In the proposed research soft computing approach is applied to weather analysis in Visakhapatnam, Andhra Pradesh, India. A new approach is proposed that use concept of fuzzy logic in meteorology system. One method to understand such type of inconsistent problem is by using a technique that learns the pattern using the previous data like an adaptive neural network system until it reaches the required level of training error. Such technique is possible by using a
hybridization of a neural network and Fuzzy Inference System. This is Adaptive Neuro Fuzzy Inference System called ANFIS.

ANFIS is integration of Artificial Neural Networks and fuzzy logic methods. It has the inherent quality to capture the benefits of both these methods in a single framework. ANFIS eliminates the basic problem in fuzzy system design (defining the membership function parameters and design of fuzzy if–then rules) by effectively using the learning capability of ANN for automatic fuzzy rule generation and parameter optimization. (Bacanli et al., 2009). These limitations have been a big reason behind the formulation of an intelligent hybrid systems that overcomes the limitations of neural networks and fuzzy systems. Fuzzy systems required to have an automatic adaption procedure which is comparable to neural networks. Hybridizing both approaches should include advantages and exclude disadvantages of both the techniques.

CONCLUSION

From the survey of literature, it is concluded that soft computing techniques especially, Neural Network and Fuzzy System has become interesting preference for researchers to solve weather forecasting problems. Development of hybridization of Neural Network and Fuzzy System are still the major issues related to weather forecasting.

Therefore, in the present work, weather forecasting system with very good performance measures including very less Root Mean Square Error (RMSE) have been considered. An attempt has been made to develop hybrid algorithm
that is based on combination of powers of two algorithms Neural Network and Fuzzy System for weather forecasting has been done.
CHAPTER 3
DESIGN METHODOLOGIES

3.1 INTRODUCTION

Weather forecasting is one of the most challenging real time problems around the globe because of both its practical value in meteorology and popular sphere for scientific research. In consistency of meteorological factors, especially those weather parameters which are major responsible for weather forces us to develop an approach which can recognize such an inconsistent pattern and use it for future prediction of weather which will be very much feasible. So long, the techniques used for weather forecasting is probability distribution, curve fitting, normalization. Still our predications are unpredictable. There is a need to develop a new approach that use concept of soft computing instead of the traditional crisp approaches to predict weather more accurately.

The daily meteorological data consisting of Temperature, Relative Humidity and Vapour Pressure of Visakhapatnam will be analyzed using Statistical method and ANFIS in this study to conduct a performance comparison based on the minimum error rate in terms of MAE and RMSE in weather forecasting. The main Problem of this research work is to test the ability of ANFIS Model to predict weather Phenomena with lesser error.
Hu (1996) initiated the implementation of Artificial NN, an important Soft Computing methodology in weather forecasting. Liu and Chandrasekar (2000) developed a fuzzy logic and neuro-fuzzy system for classification of a hydrometer type based on polarimetric radar measurements where fuzzy logic was used to infer a hydrometer type, and the neural network learning algorithm was used for automatic adjustment of the parameters of the fuzzy sets in the fuzzy logic system according to prior knowledge. Ozelkan and Duckstein (1996) compared the performance of regression analysis and fuzzy logic in studying the relationship between monthly atmospheric circulation and precipitation. Cook and Wolf (1991) developed a neural network to predict the average air temperatures. Fuzzy logic can also be of great use in the atmospheric data analysis and prediction.

3.2 STEPS OF DESIGN METHODOLOGY

In order to understand the main problem of the research work, it can be considered in the following two steps as shown in figure 3.1.

Step 1 (P1):

Weather data is in very huge amount. It consists of $13.68 \times 10^3$ (=6.25years x 365 days x 6 main parameters) entities and constitutes several parameters namely, MAXTEMP °C, MIN TEMP °C, AVP(mm)M, AVP(mm)E, RH(%) M, RH(%) E. The main problem of this part is to analyze and process the weather data of study area, Visakhapatnam. The variables have to be chosen in such a way that it include the effect of weather profile. Then the problem comes to
decide the weather variables which have significant impact on output variable. We need to decide various inputs and output parameters using various statistical techniques like Karl Pearson’s coefficient of correlation, probability distribution of various parameters etc. and then process the data using standard normalization function.

**Fig. 3.1: Design of Research Methodology**

**Step 2 (P2):**

Then we need to develop various FIS and ANFIS with different data size, with
different membership function like (3,-3,-3,-1), (3,-5,-5,-1), (3, -3, -5, -1), (5, -5, -5, -1), (5, -3, -3, -1), (5, -5, -3, -1) and (5,-3,-5,-1) with different epochs, and using subtractive clustering or grid partitioning technique to find error rate between predicted and observed value.

On the other hand, same data is used to develop statistical model(ARIMA) and error is calculated between predicted and observed value. Lastly, Comparison between Statistical Model(ARIMA) and ANFIS Model for the minimum error rate is done.

3.3 DATA ANALYSIS

3.3.1 DATA COLLECTION

In order to forecast weather, we need to characterize a lot of parameters which directly or indirectly affects it.

The variables of weather forecasting are maximum temperature (MAX TEMP °C), minimum temperature (MIN TEMP °C), Actual Vapor Pressure of morning (AVPmm M), Actual Vapor Pressure of evening (AVP mm E), Relative Humidity in percentage in morning (RH % M), Relative Humidity in percentage in evening RH(%) E, Wind Speed (WS Km/h), Bright Sunshine Hours, Pan Evaporation (mm) and Local Monsoonal Precipitation (mm).These variables also represent a strong correlation with the other parameters.

3.3.1.1 INPUT AND OUTPUT VARIABLES

In these Experiments, firstly input data is loaded as X vector (n x 3) whose first
column is the data of Temperature (TEMP °C), second column Average Vapor Pressure (AVP mm), third column is Relative Humidity (RH) and the output is loaded as Y, a column matrix containing the data of weather profile (WP °C) as defined in chapter 1. List of input and output vector is also given in table 3.1.

**Table 3.1: List of Input and output variables**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Weather Parameter</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>Temperature Temp(°C)</td>
<td>Input</td>
</tr>
<tr>
<td>X2</td>
<td>Average Vapour Pressure AVP(mm)</td>
<td>Input</td>
</tr>
<tr>
<td>X3</td>
<td>Relative Humidity RH(%)</td>
<td>Input</td>
</tr>
<tr>
<td>Y1</td>
<td>Weather Profile WP (°C)</td>
<td>Output</td>
</tr>
</tbody>
</table>

**3.3.1.2 PRIMARY SOURCE OF DATA**

In this research, the 6+ years of weather data of Visakhapatnam, Andhra Pradesh, India is taken from Weather Station (India Metrological Department-IMD) situated at Hyderabad. This process of recording the meteorological data is also verified by personal interviews and checking the procedures at Met Observatory situated at Visakhapatnam.

A facility, either on land or sea with instruments and equipment for observing atmospheric conditions to provide information for weather forecasts and to study the weather and climate is called a weather station. The measurements that are
taken at a weather station include temperature, Vapour Pressure, humidity, wind speed, wind direction and precipitation amounts.

Wind measurements are taken as free of other obstructions as possible, while temperature and humidity measurements are kept free from direct solar radiation. Manual observations are taken at least once daily, while automated observations are taken at least once an hour. Weather Station of Visakhapatnam, the eastern coastal port of Andhra Pradesh, India, is located between $17^0 42' 0''$ N to $83^0 18' 0''$ E, Latitude : 20.0, Longitude : 77.0 as shown in figure 3.2.

Fig.3.2: Continental Location of Visakhapatnam, Andhra Pradesh, India.

### 3.3.1.3 RAW DATA COLLECTION

Variables like MAX TEMP °C, MIN TEMP °C, AVP (mm)M, AVP(mm)E, RH(%) M, RH(%) E, AVS WS Km/h, Bright Sunshine Hours, Pan Evaporation (mm) are considered as input variables and weather profile(°C) as output variable which depends upon input variables.

The data of every year is shown as sample raw data in table 3.2.
Table 3.2: Sample Raw Data Set (Before processing)

In the proposed research, 6+years of weather data is collected, out of which 85% of data is used as training data set and remaining 15% of data is used as testing data set.

<table>
<thead>
<tr>
<th>MAX</th>
<th>MIN</th>
<th>AVP_M</th>
<th>AVP_E</th>
<th>RH_M</th>
<th>RH_E</th>
</tr>
</thead>
<tbody>
<tr>
<td>36.257</td>
<td>21.428</td>
<td>18.871</td>
<td>18.371</td>
<td>79.142</td>
<td>44.714</td>
</tr>
<tr>
<td>35.142</td>
<td>23.142</td>
<td>18.8</td>
<td>16.957</td>
<td>71.0</td>
<td>36.857</td>
</tr>
<tr>
<td>38.0</td>
<td>23.828</td>
<td>20.014</td>
<td>20.271</td>
<td>66.857</td>
<td>38.428</td>
</tr>
<tr>
<td>38.728</td>
<td>25.814</td>
<td>20.814</td>
<td>20.171</td>
<td>67.0</td>
<td>42.857</td>
</tr>
<tr>
<td>36.328</td>
<td>24.751</td>
<td>23.871</td>
<td>22.742</td>
<td>80.0</td>
<td>53.571</td>
</tr>
<tr>
<td>38.971</td>
<td>26.642</td>
<td>23.25</td>
<td>21.357</td>
<td>64.857</td>
<td>38.142</td>
</tr>
<tr>
<td>38.285</td>
<td>25.428</td>
<td>22.642</td>
<td>20.042</td>
<td>75.714</td>
<td>47.0</td>
</tr>
<tr>
<td>36.057</td>
<td>24.785</td>
<td>23.742</td>
<td>22.257</td>
<td>80.714</td>
<td>58.714</td>
</tr>
<tr>
<td>31.614</td>
<td>23.571</td>
<td>23.914</td>
<td>24.485</td>
<td>90.285</td>
<td>75.75</td>
</tr>
<tr>
<td>34.257</td>
<td>24.842</td>
<td>22.471</td>
<td>22.357</td>
<td>81.285</td>
<td>58.857</td>
</tr>
<tr>
<td>34.928</td>
<td>23.3</td>
<td>22.542</td>
<td>22.3</td>
<td>84.714</td>
<td>56.571</td>
</tr>
</tbody>
</table>

3.4 GENERAL FRAMEWORK OF PROPOSED STUDY

3.4.1 DESIGN OF ANFIS

In order to perform weather forecasting using ANFIS and compare the performance of minimum error rate with different data set, the following steps have been followed:
Fig. 3.3: General Framework of Proposed Study

a) Determine weather dataset attributes.

b) Preprocessing the dataset.

c) Modeling of ANFIS architecture.

d) Calculating RMSE using MatLab toolbox *anfisedit*

e) Compare and analysis of result

The neuro-fuzzy learning techniques provide a method for the fuzzy modeling procedure to learn information about data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input-output data. ANFIS constructs an input-output
mapping based on both human knowledge (in the form of fuzzy if-then rules) and simulated input output data pairs. It serves as a basis for building the set of fuzzy if-then rules with appropriate membership functions to generate the input output pairs.

Weather parameters which have high impact on output variable are determined.

Then weather parameters of the whole year are chosen the area under study. Then an ANFIS Model is described and a neuro fuzzy system is trained on the basis of training data set which is nothing but a subset of processed data. After Defining ANFIS Model, it is run with various FIS Algorithm and error tolerance and number of epochs and we will analyze the effect of all these parameters on RMSE using MatLab and predict weather and then using statistical model (ARIMA) we will numerically compare the results. General Framework of proposed study is shown in figure 3.1.

3.4.1.1 SOFT COMPUTING MODEL USED IN THE RESEARCH

Takagi and Sugeno’s fuzzy if-then rules are used as the output WP of each rule is a linear combination of input variables MAX TEMP °C, MIN TEMP °C, AVP (mm)M, AVP(mm)E, RH(%) M, RH (%) E and a constant term. As in these experiments one single output can be taken as linear combination of input parameters therefore in the proposed study Type 3 of fuzzy inference System i.e. Takagi and Sugeno’s fuzzy Model which is nothing but ANFIS model is used.
ANFIS can be used in modeling, estimating and controlling studies in weather forecasting processes similar to other artificial intelligence methods such as NNs and Fuzzy Logic (FL). In this work, the designed ANFIS is utilized as a predictor. Prediction can be done using three input parameters like temperature, relative humidity and pressure of a particular place are collectively known as training data. Mean Average Error (MAE) value, Root Mean Square Error (RMSE) values are calculated for the training data and compared with the test data. In this work, the performances of ANFIS model and ARIMA model are compared with respect to MAE and RMSE values.

### 3.4.1.2 STRUCTURE OF ANFIS

The architecture of Neuro Fuzzy used in this study is ANFIS. This study applies a five layered ANFIS Model and the learning algorithm for training the network is hybridization of forward pass and backward pass. The total of number of nodes for every layer is different for different experiment depending upon the number of membership function of an input variable.

In the figure 3.4, Input 1 is Temperature with 3 membership function (cold, normal, hot), input 2 is Average Vapor Pressure (Average) with 3 membership function (low, average, high), input 3 as Relative Humidity with 3 membership function (less, normal, more) and a single output as Weather Profile whose degree of membership is Linear.
Layer 1 is responsible for mapping of the input variable relatively to each membership functions. In this study, ANFIS will be running in different structures influence by membership functions ((3,-3,-3,-1), (3,-5,-5,-1), (3, -3, -5, -1), (5, -5, -5, -1), (5, -3, -3, -1), (5, -5, -3, -1) and (5,-3,-5,-1)) where (5,-3,-3,-5,-1) represents various fuzzy variable from low to high with input 1 as Temperature with 3 membership function(Cold, Normal, Hot), input 2 as Average Vapor Pressure (Average) with 3 membership function (low, average, high), input 3 as Relative Humidity with 5 membership function(very less, Less, Normal, High, very high) and a single output as Weather Profile whose degree of membership is Linear. Besides this, the membership function “gaussmf” will be used in this study.

The parameters associated with the membership functions are open to change through the learning process. The computation of these parameters is facilitated by a gradient vector, which provides a measure of how well the ANFIS is modeling the input output data for a given parameter set. Once the gradient vector is obtained, back-propagation or hybrid learning algorithm can be applied in order to adjust the parameters.
3.4.1.3 HYPOTHESIS OF THE MODEL USED

ANFIS is more complex than the fuzzy inference systems, and is applicable for specific type of fuzzy inference system options. Specifically, it only supports Sugeno-type systems, and must have the following properties:

- Sugeno-type systems should be of first order.
- There should be a single output, obtained using weighted average defuzzification.
- All output membership functions must either be linear or constant.
- In this model different rules cannot share the same output membership function,
- All fuzzy if-then rules have unity weight.
3.4.1.4 MEMBERSHIP FUNCTION USED

In the Proposed Research Model generalized, a Gaussian membership function (MF) is given by two parameters \( \{c, \sigma\} \):

\[
\text{Gaussian}(x;c,\sigma) = e^{-\frac{1}{2}(x-c/\sigma)^2}
\]

A Gaussian MF is determined by \( c \) and \( \sigma \). \( c \) represents the MFs center and MFs width is calculated by \( \sigma \). These fuzzy inputs are applied to the ANFIS network and we obtain the crisp output.

Changing the values of \( c, \sigma \) will change the shape of gauss function.

3.4.1.5 ANFIS SETUP

According to tools and format available in MatLab, a network is setup in ANFIS editor in mainly two stages - Sugeno ANFIS Setup and Training of ANFIS Model.

In the ANFIS model, fuzzy inputs are developed from crisp input set by membership functions (MF) on the basis of Gaussian MF.

3.4.1.5.1 SUGENO ANFIS SETUP MODEL

To design a network using ANFIS editor, first load input data shown in ANFIS editor separated as training and testing data set. Then generate FIS using Grid Partitioning method. Sample ANFIS information is:

No. of input 3

No. of output 1

No. of input Mfs: 3 3 5

Architecture of ANFIS model (3,- 3,-5,- 1) is shown in next chapters.
3.4.1.5.2 TRAINING OF ANFIS MODEL

Training of ANFIS model is done using hybrid optimization method with error tolerance level 0.00001 for 40 epochs (assumed). Once the training is done, ANFIS model is always ready for the prediction. During this training, ANFIS will learn the whole pattern among different input in various years and within each year itself. RMSE with 0.0001 error tolerance and 40 epochs is given in next chapter.

MATLAB fuzzy logic toolbox is used to design ANFIS predictor structures. Using the given training data set, the toolbox constructs an ANFIS structure using either a back-propagation algorithm alone, or in combination with least squares type of method (hybrid algorithm). ANFIS model can be generated either from the command line, or through the ANFIS editor GUI. In this study, ANFIS Editor GUI is used to generate the ANFIS models with the chosen design parameters in construction phase. Written MATLAB code is used to train the ANFIS structure in the training step. The use of the ANFIS editor GUI can be found in program help files.

The steps in ANFIS predictor design in this study utilizing the MATLAB fuzzy logic toolbox are as follows:

1. Generated training data is loaded to the Editor GUI.
2. Design parameters, number of input MF, type of input and output MF, are chosen. Thus, initial ANFIS structure is formed.
3. The code for the training is run with the initial structure.

4. ANFIS structure constituted after training is saved to use as a predictor.

3.4.2 DESIGN OF ARIMA MODEL

This thesis focuses on a study model of forecasting weather of Visakhapatnam in three types as temperature, wind speed and pressure by using ARIMA models. That is, this thesis provides ARIMA models to forecast the weather. The training data which is used in ANFIS model, the same data were used by ARIMA model to evaluate performance criterion.

3.4.2.1 THE STEPS IN THE ARIMA MODEL BUILDING

STEP 1: Model Identification

Determine whether the series is stationary or not by considering the graph of ACF. If a graph of ACF of the time series values either cuts off fairly quickly or dies down fairly quickly, then the time series values should be considered stationary. If a graph of ACF dies down extremely slowly, then the time series values should be considered non-stationary. If the series is not stationary, it can often be converted to a stationary series by differencing. That is, the original series is replaced by a series of differences. An ARMA model is then specified for the differenced series. Differencing is done until a plot of the data indicates the series varies about a fixed level, and the graph of ACF either cuts off fairly quickly or dies down fairly quickly.
Model for non-seasonal series are called Autoregressive integrated moving average model, denoted by ARIMA \((p, d, q)\). Here \(p\) indicates the order of the autoregressive part, \(d\) indicates the amount of differencing, and \(q\) indicates the order of the moving average part. If the original series is stationary, \(d = 0\) and the ARIMA models reduce to the ARMA models. The difference linear operator \((\Delta)\), defined by

\[
\Delta Y_t = Y_t - Y_{t-1} = Y_t - BY_t = (1 - B)Y_t
\]

The stationary series \(W_t\) obtained as the \(d\)th difference \((\Delta^d)\) of \(Y_t\),

\[
W_t = \Delta^d Y_t = (1 - B)^d Y_t
\]

ARIMA \((p, d, q)\) has the general form:

\[
\phi_p(B)(1 - B)^d Y_t = \mu + \theta_q(B)\varepsilon_t
\]

Once a stationary series has been obtained, then identify the form of the model to be used

**STEP 2: Model Estimation**

Estimate the parameters for a tentative model has been selected.

**STEP 3: Model Checking**

In this step, model must be checked for adequacy by considering the properties of the residuals whether the residuals from an ARIMA model must has
the normal distribution and should be random. An overall check of model adequacy is provided by the Ljung-Box Q statistic. The test statistic Q is

$$Q_m = n (n + 2) \sum_{k=1}^{m} \frac{r_k^2(e)}{n-k} \sim \chi^2_{m-r};$$

where $$r_k(e)$$ is the residual autocorrelation at lag k,

$$n =$$ the number of residuals,

$$m =$$ the number of time lags included in the test.

If the p-value associated with the Q statistic is small (p-value < α), the model is considered inadequate. The analyst should consider a new or modified model and continue the analysis until a satisfactory model has been determined.

**STEP 4: Forecasting with the Model**

Forecasts for one period or several periods into the future with the parameters for a tentative model have been selected.

**3.5 MEANS OF ERROR PERFORMANCE**

There are various types of data analysis tools used in the experiments such as:

- Moving Average Error (MAE)
- Root Mean Square Error (RMSE)
- Coefficient of determination ($$R^2$$)

MAE, RMSE and $$R^2$$ were used as statistical indicators for assessing goodness of the models.
**MAE:** Mean Average Error is the absolute difference between the forecasted and observed output variable and is given by:

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - o_i| \tag{3.6}
\]

Where \(y_i\) is the predicted value of output variable

\(o_i\) is the observed value of output variable

\(n\) is the total number of observations of data.

**RMSE:** The Root Mean Squared Error (RMSE) is given by:

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - o_i)^2 \tag{3.7}
\]

\[
RMSE = \sqrt{MSE} \tag{3.8}
\]

and is simply the square root of the MSE. MSE indicates a perfect forecast when it exhibits a zero, with errors increasing with larger MSE values. The choice of an error criterion as measure of the performance is a delicate point because it depends directly on the problem to treat (Iqdouret et al., 2006).

The units of RMSE are same as the forecasted and observed data. RMSE is the standard measure of errors in weather forecasting.

**R^2:** Coefficient of determination is given by:

\[
R^2 = \frac{\sum_{i=1}^{n} (o_i - \bar{o})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (o_i - \bar{o})^2 (y_i - \bar{y})^2}} \tag{3.9}
\]
Where $y_i$ is the predicted value of output variable, $o_i$ is the observed value of output variable, $n$ is the total number of observations of data, $\bar{y}_i$ is the mean of predicted variable and $\bar{o}_i$ is the mean of observed variable.

**CONCLUSION**

In this Chapter, the methodology is explained which is used to forecast precipitation by ANFIS model as well as by multidimensional response surface tool. All the resource data has been explained and location of area of study is described. Moreover the Architecture of ANFIS model with all of its five layers has been explained in which the generalized bell function which have used as membership function of input and output variable has been defined. Descriptive statistics of Research Methodology is shown in table 3.3.

**Table 3.3: Descriptive Statistics of Research Methodology**

<table>
<thead>
<tr>
<th>Research Design</th>
<th>Descriptive research Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research Type</td>
<td>Quantitative</td>
</tr>
<tr>
<td>Data collection methods</td>
<td>Primary Data, Secondary Data</td>
</tr>
<tr>
<td>Primary source of data</td>
<td>Meteorological observations at Visakhapatnam Met Observatory</td>
</tr>
<tr>
<td>Sampling size &amp; design</td>
<td>$13.68 \times 10^3 = (6.25 \text{ years} \times 365 \text{ days} \times 6 \text{ parameters})$</td>
</tr>
<tr>
<td>Sampling technique</td>
<td>Average based on daily weather forecast for variables such as Temperature, Vapour Pressure, Relative Humidity</td>
</tr>
<tr>
<td>Tabulation of Data</td>
<td>SPSS, MatLab Data Editor(85% of training data and 15% of testing data)</td>
</tr>
<tr>
<td>Analysis of Data</td>
<td>Moving Average Error, Root mean Squares error, Coefficient of determination($R^2$), ANFIS Architecture with different Membership Function, ARIMA Architecture</td>
</tr>
</tbody>
</table>
In short, this chapter describe in detail statistical model (ARIMA) with the basic definition of MAE, RMSE, co-efficient of determination etc. Last section of the chapter is very important as it explain various MatLab commands, how to load the data, how to create FIS model and how to train data. All the experiments based on these methodology and MatLab commands are done in next chapter.

Results have been obtained by using software.

Result analysis of both models will be discussed in the next chapter.
CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 INTRODUCTION

Based on performance of the models ANFIS and ARIMA the scholar wanted to know the best model out of those two to calculate the weather forecasting error at minimized level. To determine the performance of these two models the scholar in his study used Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and $R^2$ Value and to forecast the parameters, further he used three inputs and one output to forecast the weather using (85% of training and 15% of testing data) in ANFIS model and again he applied the same data in ARIMA model to evaluate the performance criteria. The compared the performance criteria of ANFIS and ARIMA have been shown in the following tables.

A lot of Experiments has been done on various data size, types and on various ANFIS Architecture and ARIMA(0,1,1) model. List of Some successful experiments are Listed below:

**Experiment 1**: Estimate MAE, RMSE, $R^2$ for ARIMA model using SPSS tool.

**Experiment 2**: Estimate MAE, RMSE with Grid Partition Method

**Experiment 3**: Estimate MAE, RMSE with Sub Clustering Algorithm

**Experiment 4**: Create ANFIS Architecture with 3-inputs and single output with Mf (3,-3,-3,-1)
Experiment 5: Create ANFIS Architecture with 3-inputs and single output with Mf (3,-5,-5,-1)

Experiment 6: Create ANFIS Architecture with 3-inputs and single output with Mf (3,-3,-5,-1)

Experiment 7: Create ANFIS Architecture with 3-inputs and single output with Mf (5,-5-5,-1)

Experiment 8: Create ANFIS Architecture with 3-inputs and single output with Mf (5,-3,-3,-1)

Experiment 9: Create ANFIS Architecture with 3-inputs and single output with Mf(5,-5,-3,-1)

Experiment 10: Create ANFIS Architecture with 3-inputs and single output with Mf(5,-3,-5,-1)

4.2 DATA DESCRIPTION

In this research, the 6+ years of weather data of Visakhapatnam, Andhra Pradesh, India is taken from India meteorological department (IMD), Hyderabad. Selection of dataset is important in order to achieve the objective in this study.

In order to forecast weather, there is a need to characterize a lot of parameters which directly or indirectly affects the weather. The following variables are taken into consideration for weather forecasting: maximum temperature (MAX TEMP °C), minimum temperature (MIN TEMP °C), Actual Vapour Pressure of morning (AVP mm M), Actual Vapour Pressure of evening (AVP mm E), Relative Humidity
in percentage in morning (RH % M), Relative Humidity in percentage in evening
RH(%) E.

Fig. 4.1: Visakhapatnam climate graph round the year

4.2.1 TEMPERATURE

Visakhapatnam has dry periods during month December, January, February and
March. Average temperature is 28.4°C. On average the temperature are always
high and warmest month is May, coolest month is January. Value of maximum –
minimum temperature (max-min-temp) is shown in figure 4.2.
Fig.4.2: Max – Min Temperature graph of Visakhapatnam

4.2.2 AVERAGE VAPOUR PRESSURE (AVP)

Actual vapour pressure is called as saturated vapour pressure when air is saturated. It may also be defined as the pressure exerted by water vapour actually present in air. It is the saturated vapour pressure at dew point. Values of Actual vapour pressure observed in morning and evening i.e. AVP_M, AVP_E resp.

4.2.3 RELATIVE HUMIDITY

Range of relative humidity varies from 5 to 100 percent. At Visakhapatnam, the driest month of the year is December. During winter months, relative humidity remains significantly high. Value of RH_M, RH_E is shown in figure 4.3.

Fig.4.3: Relative Humidity graph of Visakhapatnam
4.3 DATA ANALYSIS, PROCESSING AND NORMALIZATION

Further Karl Pearson’s Coefficient of correlation between various input and output variables is calculated. The raw data whose sample data set is shown in Table 3.2 (Refer table in research methodology chapter) is processed using formula so that each value of \( x \) comes under the range \([0,1]\).

\[
\text{norm}(x) = \frac{x - \text{min}}{\text{max} - \text{min}} \\
\]

Where \( \text{norm}(x_i) \) is normalized value of variable \( x_i \)

Min is the minimum value of \( x_i \) for all \( i \).

Max is the maximum value of \( x_i \) for all \( i \).

The whole normalized data shown in Table 4.1 is collected to divide into two parts as follows:

1. Training Data Set (85%)
2. Testing data Set (15%)

This data is uploaded and a Fuzzy inference system is created in ANFIS editor to predict weather (WP \(^\circ\)C).
Table 4.1: Normalized Data after Processing (Sample)

<table>
<thead>
<tr>
<th>Temp</th>
<th>Vapor pressure</th>
<th>Rel Humid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.984357</td>
<td>0.384983</td>
<td>0.994501</td>
</tr>
<tr>
<td>0.909091</td>
<td>0.319889</td>
<td>0.983393</td>
</tr>
<tr>
<td>0.937131</td>
<td>0.518302</td>
<td>0.76122</td>
</tr>
<tr>
<td>0.703955</td>
<td>0.548971</td>
<td>0.59459</td>
</tr>
<tr>
<td>0.66706</td>
<td>0.795579</td>
<td>0.683459</td>
</tr>
<tr>
<td>0.850059</td>
<td>0.718592</td>
<td>0.694568</td>
</tr>
<tr>
<td>0.698052</td>
<td>0.711081</td>
<td>0.772328</td>
</tr>
<tr>
<td>0.317296</td>
<td>1.00025</td>
<td>0.305765</td>
</tr>
<tr>
<td>0.200708</td>
<td>0.873817</td>
<td>0.216896</td>
</tr>
<tr>
<td>0.342385</td>
<td>0.71734</td>
<td>0.527938</td>
</tr>
<tr>
<td>0.571133</td>
<td>0.717966</td>
<td>0.750111</td>
</tr>
</tbody>
</table>

4.4 EXPERIMENTS ON FORECASTING USING STATISTICAL ARIMA MODEL

In this experiment, the weather dataset (training data and testing data) is imported into ARIMA in order to investigate the performance of MAE, RMSE and R Squared.
After importing the data, ARIMA model has to be created. From the tool bar, select Analyze > Forecasting > Create Models. The criteria of the ARIMA model have to be selected through trial and error method. Here the values of p,d,q has entered as 0,1,1.

Fig.4.4: Creating models in SPSS tool for ARIMA
Fig.4.5: Selecting the dependent variables in SPSS tool for ARIMA

Fig.4.6: Setting p,d,q variables as 0,1,1 respectively in SPSS tool for ARIMA
The ARIMA code is generated and the time-series modeler is obtained. Finally the values of MAE, RMSE and R Squared are estimated for training dataset and testing dataset and the results are tabulated in tables 4.5, 4.6, 4.7 respectively.

![ARIMA Code](image)

**Fig.4.7: Code and time series modeler in SPSS tool for ARIMA(0,1,1)**

### 4.5 EXPERIMENTS ON FORECASTING USING ANFIS MODEL

In this experiment, the weather dataset is implemented into ANFIS in order to investigate the performance of its MAE, RMSE and R Squared. The MAE and RMSE are used in these experiments to compare and find the best result. MAE stands for Mean Absolute Error which is the absolute error between the forecast and observation and RMSE is root mean squared error. In all the experiments
which are done in further sub-sections, three input and single output variables are taken whose fuzzy properties are described in table 4.2.

**Table 4.2: Membership functions and list of fuzzy variables**

<table>
<thead>
<tr>
<th>Fuzzy Variables</th>
<th>Membership functions (Mfs)</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature</td>
<td>Very cold, cold, normal, hot, very hot</td>
<td>Input</td>
</tr>
<tr>
<td>Average Vapour Pressure</td>
<td>Very low, low, average, high, very high</td>
<td>Input</td>
</tr>
<tr>
<td>Relative Humidity</td>
<td>Very less, less, normal, more, most</td>
<td>Input</td>
</tr>
</tbody>
</table>

**Experiment 2: Estimate MAE, RMSE with Grid Partition Method**

**Objective:**

This experiment aims to check the effect of grid partitioning method over subtractive clustering algorithm on the selected data set.

**Material used:**

Various fuzzy variables are taken as input and output variables whose value ranges from low to high. Input 1 is Temperature with 3 membership function (cold, normal, hot), input 2 is Average Vapor Pressure (Average) with 3 membership function (low, average, high), input 3 as Relative Humidity with 3 membership function (less, normal, high) and a single output as weather prediction whose degree of membership is Linear.

**Method:**

RMSE is plotted with 40 epochs and 0.00001 error tolerance using grid partitioning method in figure 4.8.
Fig.4.8: Grid partition method for estimating RMSE

Result:

It is clear from figure 4.8 that RMSE in grid partition method is given as 6.901e-005 and average testing error is 0.0069345 as shown in figure 4.9.

Limitations:

It is observed from the figure 4.10 number of fuzzy rules has got increased drastically because in this method each and every index of data is taken into consideration. Moreover time required to create this FIS is very large.
Fig.4.9: Grid partition method for estimating MAE

Fig.4.10: Grid partition method for ANFIS structure (3, -3, -3, -1)
Experiment 3: Estimate MAE, RMSE with Sub Clustering Algorithm

Objective:

This experiment aims to check the effect of subtractive clustering algorithm on the selected data set.

Material used:

When subtractive clustering algorithm is taken, ANFIS editor automatically makes the clusters of selected data set and set the shape and number of membership function.

Method:

In this algorithm various clusters are formed from selected data set to reduce complexity in terms of time and space; however RMSE is more than that of grid partitioning method. Range of influence of clusters is taken as 0.5, squash factor as 1.25, accept ratio as 0.5 and reject ratio as 0.15. In figure 4.12, the Anfis Architecture is initiated with 40 epochs and 0.00001 error tolerance using Subtractive Clustering algorithm. The ANFIS model structure of Subtractive Clustering algorithm is shown in figure 4.11 in which number of membership function is defined automatically depending upon the value taken by each cluster. Figure 4.12 shows the number of fuzzy rules.

Then RMSE is plotted in this algorithm as shown in diagram 4.13 and it is found that the value of RMSE become stable after 2 epochs, however at the end of 40 epochs the value reached at 4.2545e-007.
Fig. 4.11: Parameters for Subtractive Clustering method

Fig. 4.12: Subtractive Clustering fuzzy rules for ANFIS structure (3, -3, -3, -1)
Result:

It is clear from figure 4.13 that RMSE in Subtractive Clustering method is given as $4.2545\times 10^{-7}$ and average testing error is $3.3741\times 10^{-7}$ as shown in figure 4.14.

Limitations:

It is observed from table 4.3 that RMSE and Average Testing error in this experiment 3 of subtractive clustering algorithm provides more accurate values as compared to previous experiment 2 which was of grid partitioning method.

Fig.4.13: Subtractive Clustering method for RMSE plot
Fig. 4.14: Subtractive Clustering method for MAE plot

Table 4.3: Comparison of Grid Partitioning Method and Subtractive Clustering

<table>
<thead>
<tr>
<th>Exp No</th>
<th>Method</th>
<th>RMSE</th>
<th>MAE</th>
<th>No of Fuzzy rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Grid Partitioning method</td>
<td>6.901e-005</td>
<td>0.0069345</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>Subtractive Clustering method</td>
<td>4.2545e-007</td>
<td>3.3741e-007</td>
<td>02</td>
</tr>
</tbody>
</table>

Experiment 4: Create ANFIS Architecture with 3-inputs and single output with Mfs (3-3-3,-1)

Objective:

In this experiment, the weather dataset is implemented into ANFIS Model (3,-3,-3,-1) in order to investigate the performance of its RMSE. The RMSE is used in these experiments to compare and find the best result. MSE stands for Average
square error which is the average squared error between the forecast and observation and RMSE is root mean squared error.

**Material used:**

In this experiment, ANFIS is run in different structures influence by membership functions (3-3-3-1) represents various fuzzy variable from low to high with input 1 as Temperature with 3 membership function (cold, normal, hot), input 2 as Average Vapour Pressure (Average) with 3 membership function (low, average, high), input 3 as Relative Humidity with 3 membership function (less, normal, more) and a single output as weather prediction whose degree of membership is Linear.

![Fig. 4.15 & 4.16: Training of ANFIS hybrid model with 40 epochs and testing of the same model respectively](image)

**Fig. 4.15 & 4.16: Training of ANFIS hybrid model with 40 epochs and testing of the same model respectively**
Method:

The Architecture of the above described ANFIS Model is shown in figure 4.17 which uses Hybrid Learning.

ANFIS information:

Number of nodes: 78
Number of linear parameters: 108
Number of nonlinear parameters: 18
Total number of parameters: 126
Number of training data pairs: 64
Number of checking data pairs: 0
Number of fuzzy rules: 27

Fig. 4.17: ANFIS model architecture (3, -3, -3, -1)
Result:

Figure 4.15 shows the RMSE when the model is run for 40 epochs, using Grid Partition Algorithm and Hybridization of SD and LSE, which is 6.901e-005, quite acceptable to forecast weather. Input-Output Surface view with sample inputs [NaN NaN 0.62 0.16] is shown in figure 4.18 respectively, where NaN is the value that varies according to the variable which is plotted as input. Finally, the model is plotted against testing data set and the graph shown in figure 4.16 that compares between Predicted FIS output and testing data Set.

Fig. 4.18: Surface viewer of the ANFIS model
Experiment 5: Create ANFIS Architecture with 3-inputs and single output with Mfs (3-5-5,-1)

Objective:

In this experiment, the weather dataset is implemented into ANFIS Model (3,-5,-5,-1) in order to investigate the performance of its RMSE.

Material used:

In this experiment, ANFIS is run in different structures influence by membership functions (3-5-5-1) represents various fuzzy variable from low to high with input 1 as Temperature with 3 membership function (cold, normal, hot), input 2 as Average Vapour Pressure (Average) with 5 membership function (very low, low, average, high, very high), input 3 as Relative Humidity with 5 membership function (very less, less, normal, more, most) and a single output as weather prediction whose degree of membership is Linear.

Fig. 4.19 & 4.20: Input, Output data with its MFs and Loading training data set(ooooo), testing data set(*****) respectively.
Method:

The Architecture of the above described ANFIS Model is shown in figure 4.21 which uses Hybrid Learning.

ANFIS information:

Number of nodes: 182
Number of linear parameters: 300
Number of nonlinear parameters: 26
Total number of parameters: 326
Number of training data pairs: 64
Number of checking data pairs: 0
Number of fuzzy rules: 75

Fig.4.21& 4.22: ANFIS STRUCTURE with 3 inputs, 1 output(3, -5, -5, -1) and Comparison of predicted and observed value respectively

Result:

The RMSE when the model is run for 100 epochs, using Grid Partition Algorithm and Hybridization of SD and LSE, which is 9.99e-005, quite acceptable to forecast weather. Input-Output Surface view with sample inputs [NaN NaN 0.62 0.16] is shown in figure 4.24 respectively, where NaN is the value that varies
according to the variable which is plotted as input. Finally, the model is plotted against testing data set and the graph shown in figure 4.22 that compares between Predicted FIS output and testing data set.

Fig.4.23: Rule viewer with 75 rules

Fig.4.24: Surface viewer with reference input [NaN NaN 0.62 0.16]
Experiment 6: Create ANFIS Architecture with 3-inputs and single output with Mfs (3-3-5,-1)

Objective:
In this experiment, the weather dataset is implemented into ANFIS Model (3,-3,-5,-1) in order to investigate the performance.

Material used:
In this experiment, ANFIS is run in different structures influence by membership functions (3-3-5-1) represents various fuzzy variable from low to high with input 1 as Temperature with 3 membership function(cold, normal, hot), input 2 as Average Vapour Pressure (Average) with 3 membership function (low, average, high), input 3 as Relative Humidity with 5 membership function (very less, less, normal, more, most) and a single output as weather prediction whose degree of membership is Linear.

Fig. 4.25 & 4.26: Input, Output data with its MFs(3,-3,-5,-1) and Training of ANFIS hybrid model with 40 epochs respectively.
Method:

The Architecture of the above described ANFIS Model is shown in figure 4.27 which uses Hybrid Learning.

ANFIS information:

Number of nodes: 118  
Number of linear parameters: 180  
Number of nonlinear parameters: 22  
Total number of parameters: 202  
Number of training data pairs: 64  
Number of checking data pairs: 0  
Number of fuzzy rules: 45

Matlab command used is:

```matlab
>>fismat=genfis1(train)
```

```
fismat=

name:’anfis’

type:’sugeno’

andMethod:’prod’
```
orMethod: 'max'
defuzzMethod: 'wtaver'
impMethod: 'prod'
aggMethod: 'max'
input: [1x3 struct]
output: [1x1 struct]
rule: [1x45 struct]

Result:

Figure 4.26 shows the RMSE when the model is run for 40 epochs, using Grid Partition Algorithm, which is 0.00010605 is quite acceptable to forecast weather. Input-Output Surface view is shown in figure 4.30. Finally, the model is plotted against testing data set and the graph shown in figure 4.31 that compares between Predicted FIS output and testing data Set.

Fig. 4.29 & 4.30: Rule viewer with 45 rules and Surface viewer for ANFIS model (3,-3,-5,-1) respectively.
Fig.4.31: Comparison of predicted and observed value \((3,-3,-5,-1)\)

Experiment 7a: Create ANFIS Architecture with 3-inputs and single output with Mfs \((3-3-5,-1)\)

For this experiment the objective and method adopted remains same. While loading the data in the ANFIS editor, training and testing data are interchanged.
Fig. 4.32 & 4.33: Loading training data set (ooooo) and testing data set (*****) and Comparison of predicted and observed value (3,-3,-5,-1) respectively.

ANFIS information:

Number of nodes: 118
Number of linear parameters: 180
Number of nonlinear parameters: 22
Total number of parameters: 202
Number of training data pairs: 11
Number of checking data pairs: 0
Number of fuzzy rules: 45

Result:

The RMSE when the model is run for 40 epochs, using Grid Partition Algorithm, which is 7.2387e-007 is quite acceptable to forecast weather.

Experiment 7: Create ANFIS Architecture with 3-inputs and single output with Mfs (5-5-5,-1)

Objective:

In this experiment, the weather dataset is implemented into ANFIS Model (5,-5,-
5,-1) in order to investigate the performance of the model.

**Material used:**

In this experiment, ANFIS is run in different structures influence by membership functions (5-5-5-1) represents various fuzzy variable from low to high with input 1 as Temperature with 5 membership function, input 2 as Average Vapour Pressure (Average) with 5 membership function, input 3 as Relative Humidity with 5 membership function and a single output as weather prediction whose degree of membership is Linear. All the membership functions are taken according to the table 4.2.

![Image](image.png)

**Fig. 4.34 & 4.35: Input, Output data with its MFs(5,-5,-5,-1) and ANFIS architecture respectively.**

**Method:**

The Architecture of the above described ANFIS Model is shown in figure 4.35 which uses Hybrid Learning.

**ANFIS information:**
Number of nodes: 286
Number of linear parameters: 500
Number of nonlinear parameters: 30
Total number of parameters: 530
Number of training data pairs: 64
Number of checking data pairs: 0
Number of fuzzy rules: 125

Fig. 4.36: Training of ANFIS hybrid model with its MFs(5,-5,-5,-1)
Fig. 4.37: Comparison of predicted and observed value for MFs(5,-5,-5,-1)

Fig. 4.38 & 4.39: Rule viewer and Surface viewer for MFs(5,-5,-5,-1) respectively
Result:

Figure 4.36 shows the RMSE when the model is run using Grid Partition Algorithm, which is 4.5398e-005. Input-Output Surface view is shown in figure 4.39.

Experiment 8: Create ANFIS Architecture with 3-inputs and single output with Mfs (5-3-3,-1)

Objective:

In this experiment, the weather dataset is implemented into ANFIS Model (5,-3,-3,-1) in order to investigate the performance of the model.

Material used:

In this experiment, ANFIS is run in different structures influence by membership functions (5-3-3-1) represents various fuzzy variable from low to high with input 1 as Temperature with 5 membership function, input 2 as Average Vapour Pressure (Average) with 3 membership function, input 3 as Relative Humidity with 3 membership function and a single output as weather prediction whose degree of membership is Linear. All the membership functions are taken according to the table 4.2.
Fig. 4.40 & 4.41: Input, Output data with its MFs(5,-3,-3,-1) and ANFIS architecture respectively.

Method:

The Architecture of the above described ANFIS Model is shown in figure 4.41 which uses Hybrid Learning.

ANFIS information:

Number of nodes: 118  
Number of linear parameters: 180  
Number of nonlinear parameters: 22  
Total number of parameters: 202  
Number of training data pairs: 64  
Number of checking data pairs: 0  
Number of fuzzy rules: 45
Fig. 4.42 & 4.43: Comparison of predicted and observed value with its MFs(5,-3,-3,-1) and Surface viewer for MFs(5,-3,-3,-1) respectively.

Result:

Figure 4.42 shows the RMSE when the model is run using Grid Partition Algorithm, which is 8.9672e-005. Input-Output Surface view is shown in figure 4.43.

Experiment 9: Create ANFIS Architecture with 3-inputs and single output with Mfs (5-5-3,-1)

Objective:

In this experiment, the weather dataset is implemented into ANFIS Model (5,-5,-3,-1) in order to investigate the performance of the model.

Material used:

In this experiment, ANFIS is run in different structures influence by membership functions (5-5-3-1) represents various fuzzy variable from low to high with input 1
as Temperature with 5 membership function, input 2 as Average Vapour Pressure (Average) with 5 membership function, input 3 as Relative Humidity with 3 membership function and a single output as weather prediction whose degree of membership is Linear. All the membership functions are taken according to the table 4.2.

**Method:**

The ANFIS information of the above described ANFIS Model which uses Hybrid Learning is below.

ANFIS information:

- Number of nodes: 182
- Number of linear parameters: 300
- Number of nonlinear parameters: 26
- Total number of parameters: 326
- Number of training data pairs: 64
- Number of checking data pairs: 0
- Number of fuzzy rules: 75

**Result:**

The RMSE when the model is run using Grid Partition Algorithm, which is 9.427e-005.
Fig. 4.44 & 4.45: Input, Output data with its MFs(5,-5,-3,-1) and Comparison of predicted and observed value with its MFs(5,-5,-3,-1) respectively.

Experiment 10: Create ANFIS Architecture with 3-inputs and single output with Mfs (5-3-5,-1)

Objective:
In this experiment, the weather dataset is implemented into ANFIS Model (5,-3,-5,-1) in order to investigate the performance of the model.

Material used:
In this experiment, ANFIS is run in different structures influence by membership functions (5-3-5-1) represents various fuzzy variable from low to high with input 1 as Temperature with 5 membership function, input 2 as Average Vapour Pressure (Average) with 3 membership function, input 3 as Relative Humidity with 5 membership function and a single output as weather prediction whose degree of membership is Linear. All the membership functions are taken according to the table 4.2.
Fig. 4.46 & 4.47: Input, Output data with its MFs(5,-3,-5,-1) and ANFIS architecture with its MFs(5,-3,-5,-1) respectively.

Method:

The Architecture of the above described ANFIS Model is shown in figure 4.47 which uses Hybrid Learning.

ANFIS information:

Number of nodes: 182
Number of linear parameters: 300
Number of nonlinear parameters: 26
Total number of parameters: 326
Number of training data pairs: 64
Number of checking data pairs: 0
Number of fuzzy rules: 75
Fig. 4.48: Comparison of predicted and observed value with its MFs(5,-3,-5,-1).

Result:

The RMSE when the model is run using Grid Partition Algorithm, which is 6.1603e-005. Input-Output Surface view is shown in figure 4.50.
Fig. 4.49 & 4.50: Rule viewer and Surface viewer with its MFs(5,-3,-5,-1) respectively.

4.6 RESULTS OF VARIOUS ANFIS MODELS

Table 4.4 shows comparison between various ANFIS Models. From first two rows of table 4.3, it is clear that RMSE is lesser in Grid partitioning method, however time and space complexity increases.

Table 4.4: Results of various ANFIS Models

<table>
<thead>
<tr>
<th>Exp no</th>
<th>Experiment name</th>
<th>ANFIS model</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Estimate MAE, RMSE with Grid Partition Method</td>
<td>0.000001 error tolerance and 40 epoch applied to (3, -3, -3, -1) ANFIS model</td>
<td>0.0069345</td>
<td>6.901e-005</td>
</tr>
<tr>
<td>3</td>
<td>Estimate MAE, RMSE with Sub Clustering Algorithm</td>
<td>0.000001 error tolerance and 40 epoch.</td>
<td>3.3741e-007</td>
<td>4.2545e-007</td>
</tr>
<tr>
<td></td>
<td>Create ANFIS Architecture with 3-inputs and single output with Mf (3,-3,-3,-1)</td>
<td>ANFIS model(3, -3, -3, -1) 40 epochs</td>
<td>0.0069345</td>
<td>6.901e-005</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>5</td>
<td>Create ANFIS Architecture with 3-inputs and single output with Mf (3,-5,-5,-1)</td>
<td>ANFIS model(3, -5, -5, -1) 100 epochs</td>
<td>0.061105</td>
<td>9.99e-005</td>
</tr>
<tr>
<td>6</td>
<td>Create ANFIS Architecture with 3-inputs and single output with Mf (3,-3,-5,-1)</td>
<td>ANFIS model(3, -3, -5, -1) training dataset</td>
<td>0.04231</td>
<td>0.00010605</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ANFIS model(3, -3, -5, -1) testing dataset</td>
<td>0.20827</td>
<td>7.2387e-007</td>
</tr>
<tr>
<td>7</td>
<td>Create ANFIS Architecture with 3-inputs and single output with Mf (5,-5-5,-1)</td>
<td>ANFIS model(5, -5, -5, -1) 0.00001 error tolerance</td>
<td>0.082426</td>
<td>4,5398e-005</td>
</tr>
<tr>
<td>8</td>
<td>Create ANFIS</td>
<td>ANFIS model(5, -3, -3, -1) 40 epochs</td>
<td>0.038876</td>
<td>8.9672e-005</td>
</tr>
</tbody>
</table>
4.7 COMPARISON BETWEEN ANFIS and ARIMA MODEL

From the above experiments it is clear that the ANFIS architecture (3,-3,-5,-1) produce RMSE lesser than any other architecture. Hence here the comparison between ANFIS architecture (3,-3,-5,-1) vs ARIMA (0,1,1) model is done and results are compared below.

Mean Absolute Error during training and testing phases of ANFIS & ARIMA models.
Table 4.5: MAE comparison between ANFIS and ARIMA for training and testing data

<table>
<thead>
<tr>
<th></th>
<th>Mean Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ANFIS</td>
</tr>
<tr>
<td>Training data</td>
<td>0.04231</td>
</tr>
<tr>
<td>Test data</td>
<td>0.20827</td>
</tr>
</tbody>
</table>

Root Mean Square Error during testing and training phases of ANFIS & ARIMA models

Table 4.6: RMSE comparison between ANFIS and ARIMA for training and testing data

<table>
<thead>
<tr>
<th></th>
<th>Root Mean Square Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ANFIS</td>
</tr>
<tr>
<td>Training data</td>
<td>0.000106</td>
</tr>
<tr>
<td>Test data</td>
<td>7.238e-007</td>
</tr>
</tbody>
</table>

Table 4.7: R Squared comparison between ANFIS and ARIMA for training and testing data

$R^2$ value during testing and training phases of ANFIS & ARIMA models
Now, the developed models are compared using the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE) and coefficient of determination $R^2$ Value. Table-4.5 MAE shows the training data is lower in ANFIS than ARIMA and the scholar got the same results in RMSE Table-4.6 where the training data is lower in ANFIS than ARIMA. Finally in $R^2$ Value, Table-4.7 the results were equal like other two table where the training data is lower in ANFIS than ARIMA. Therefore ANFIS is a better predictor of weather as compared to statistical model (ARIMA) and neuro fuzzy models.

**CONCLUSION**

This chapter can be called as processing unit of the proposed research work. All the experiments have been done in detail with objectives, material used, results and limitations.

It may be concluded that in between all ANFIS models, only (3, -3, -5, -1) model shows lesser output deviation with RMSE 0.00010605. While after a lot of experiments done on the selected data set, it may be concluded that increasing the number of Mfs does not always guarantee to decrease RMSE.
Still, it can be surely concluded that ANFIS model provides a better way to forecast weather parameters as they include in itself the tuning of membership function which have the power of handling fuzzy and linguistic variables. The findings are correct and comparable.
CHAPTER 5

CONCLUSION AND FURTHER RESEARCH

5.1 INTRODUCTION:

In this Research Work, Adaptive Neuro Fuzzy Inference System (ANFIS) for weather prediction in Visakhapatnam using input data of relative humidity, temperature and vapour pressure is applied. ANFIS is relatively new in Visakhapatnam to predict weather as most of the researchers use various statistical and probabilistic methods using meteorological data. The aim of this research is to present a comparative study of the statistical method and hybridization of neuro fuzzy methods. The results show that applications of Adaptive Neuro Fuzzy Inference System (ANFIS) provide better results as compared to Auto Regressive Integrated Moving Average (ARIMA).

5.2 CONTRIBUTION OF THE STUDY:

In the recent times, the tendency toward combining more than two soft computing techniques in one application has been growing.

ANFIS Model using neuro-adaptive learning techniques which are similar to those of neural networks was originally presented by Jang. Given an input/output data set, ANFIS constructs fuzzy inference system whose membership function parameters are adjusted using back propagation algorithm or to her similar optimization techniques has used hybrid genetic and SVD methods to design ANFIS networks.
The hierarchical fuzzy-neural controller is based on a skill knowledge database consisting of the skills acquired by the fuzzy-neuro controller. Those skills were acquired through an unsupervised learning based on Genetic Algorithms. Another very innovative application is the use of Time-Delay Neural Networks for estimating lip movements from speech. The hierarchical fuzzy-neural controller is based on a skill knowledge database consisting of the skills acquired by the fuzzy-neuro controller. Those skills were acquired through an unsupervised learning includes other fields as well, such as chemistry, medicine, information engineering, computational science, networking and distributed computing, and many others. Such a list can be a much extended one and very difficult, if not impossible, to cover in one document.

The most important advantage of using the two models such as ANFIS and ARIMA is the ability to predict natural system’s behavior at a future time, which can be used for lighting control. The implementation of ANFIS model is less complicated than that of sophisticated identification and optimization procedures. Compared to ARIMA, and Fuzzy Logic Systems, ANFIS has automated identification algorithm and easier design and compared to neural networks it has less number of parameters and faster adaptation. This prediction could be utilized as input for the artificial light and shading controls. Possibility to reduce the number of sensors and connections improve the performance of control strategy. The ANFIS based time series prediction for weather forecasting
is unique and novel as it is simple, reliable and easily accessible for different conditions.

5.3 SCOPE FOR FURTHER RESEARCH

An ANFIS is created to help user to use artificial intelligence (AI) such as ANFIS(Adaptive Network-based Fuzzy Inference System) to predict output. ANFIS can predict data using Sugeno FIS (Fuzzy Inference System) to relate membership and tune it using either back propagation or hybrid method. Seven input corresponding parameters namely; power, speed, pressure, focal distance, standoff distance, frequency, duty cycle with their respective two responses surface roughness and kerf width were used for the ANFIS training. The specially designed functions to provide the fullest output of ANFIS capability in few simple steps can be done. This can also be used to educate engineering personals to understand the fundamentals of ANFIS modeling and laser processing. The input parameters can be tested for interpolation and extrapolation levels to validate the trained model. Once this is done, the experimental validation can be conducted.

For making a prediction using time series, a great variety of approaches are available. Traditionally, a time series forecasting problems is tackled using linear techniques such as Auto Regressive Moving Average (ARMA) and Auto Regressive Integrated Moving Average (ARIMA) models.

The future appears to hold a lot of promise for the novel use and combination of these applications. The push for low-cost solutions combined with the need for
intelligent tools will result in the deployment of soft computing systems that efficiently integrate reasoning and search techniques.

The decision makers can use the fuzzy neural network approach to adjust the resource allocation to meet the company’s objectives.

Currently several studies have been done on development of the forecasting models which have focused on engineering performance, quality performance, and sustainability performance. However, innovation performance and the appropriate indicators of long-term technology investment into consideration are rarely concerned. Therefore, the integration of innovation objectives and other objectives are seldom determined.

Future work will address more weather parameters (Rainfall, Precipitation, Pan Evaporation, Bright Sunshine Hours) to be taken into account. Further even expect that the approach will be better in considering bulk data for many years (ex., 50 -100 years) for all the weather parameters.

Through improvement in proposed technique we can apply it on scheduling irrigation, radiation load, anti-forecast measure including choice of site, anti-erosion measure, soil cover and artificial climate of growth rooms or heated structures, animal housing and management, climate control in storage and transport and efficient use of herbicides and fertilizers.

The major role of climatology on the global scale is to ensure the adequate meteorological data to develop research tool and knowledge to improve productivity in term of quality and Quantity.
• Water management in crops can be improved with the aid of knowledge of physical environment.
• Knowledge of weather phenomena can be applied to practical agriculture use from soil layer of deepest plant roots to higher levels of atmosphere.
• Maximum exploitation of natural climatic resources can be made and can be further modified to increase the efficiency of agriculture system.
• Efficient and effective practices can be adopted based on current and probable weather knowledge.
• Losses in transport and storage of farm products can be minimized to some extent with knowledge of probable weather.
• Livestock production can be maximized by proper modification of micro climate of animal and birds sheds.

    Further, coactive neuro fuzzy inference system (CANFIS) can be used for weather forecasting. Since it is a generalized form of ANFIS and allows avoiding some inherent constraints to ANFIS in its original form, we expect to get even better modeling results from CANFIS.
REFERENCES


Math works “MATLAB Fuzzy Logic Tool Box”, 2014.


