University of Alberta

INTELLIGENT METHODS FOR ATMOSPHERIC VERTICAL LEVEL REPRESENTATION AND PRECIPITATION TYPE CLASSIFICATION

by

Apinya Horthong 

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2008

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-47265-1
Our file Notre référence
ISBN: 978-0-494-47265-1

Abstract

Many attempts have been made in weather forecasting to predict future atmospheric conditions. As a part of automated weather forecasting, we present novel techniques to the precipitation type classification. A prime parameter to the precipitation classification is the vertical temperature profile. It consists of temperature and dew point temperature values at several vertical levels.

A representation of vertical temperature profile is required for the classification system. We propose the application of genetic algorithms to search for the optimal location of the vertical levels whose corresponding temperature values best approximate and represent sounding temperature data under the least square error criteria. As a result, the selected vertical levels are used as a baseline to retrieve temperature profiles as the attributes to a precipitation type classifier.

The quality of the setup of vertical levels is assessed by its performance on the representation of the sounding profile with temperatures at selected vertical levels. As the approximated temperature profiles have been used to train the precipitation type classifier, the classification accuracy can also be used as an indicator to measure the quality of the vertical levels. A neural network has been built as the precipitation type classifier with the under-sampling method applied to handle the imbalanced class problem.

The results demonstrate that the optimal vertical levels obtained using genetic algorithms outperform both the standard levels from European Centre for Medium-Range Weather Forecasts (ECMWF) and equal step vertical levels constructed by a simple method that equally divide the vertical range to select the levels. Lastly, the incorporation of the re-sampling method to manage the training data improves the performance on an important rare event class, the freezing rain.

Acknowledgements

I would like to thank my supervisor Dr. Petr Musilek for his insightful support, useful suggestion, and supervision throughout my program. Dr. Musilek was always there to listen and give me advice. He also encouraged me to ask questions and express my ideas. His continuous guidance and support enabled me to complete this thesis successfully.

I would especially like to thank my co-supervisor Prof. Witold Pedrycz for his guidance and invaluable suggestion. His guidance always allows me to pursue new ideas and perspectives as well as his research group meeting which provides me broad areas of knowledge.

I would also like to thank my colleagues, Dan Arnold, for a great source of information as well as Yifan Li for his useful discussion and constructive criticism during research group meetings.

I wish to thank Ejike Ofuonye and Chuck Mackay for their assistants in revising English in this thesis.

I would also like to extend my warmest thanks to those who are closest to me for their help and encouragement throughout my program. My warm thanks also go to Thai Student Association (TSA) for making Edmonton more like home.

Last but not least, I owe my loving thanks to my family and Tee for their love, understanding, and countless support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Knowing future weather conditions is important for planning our activities as well as keeping out of weather hazards. Farmers need to know weather conditions to plan planting or harvesting. Sailors use weather information for scheduling and for preventing weather-related incidents before sailing. In 1998, the big ice storm that hit Eastern Canada and Northeastern USA was recorded as the worst storm of the century [13]. The precipitation lasted the longest in recorded history, starting out as a cold drizzle before turning into six days of snow, freezing rain, and ice pellets. The weight of ice pulled down thousands of power lines and millions of people were without electricity. According to Environment Canada, at least 25 people died, many people suffered from hypothermia, 945 people were injured, and over 4 million people in Ontario, Quebec, and New Brunswick lost power. Estimated cost of damages caused by this ice storm was $5,410,184,000 [13]. This devastating disaster reveals the need to automate and improve the accuracy of weather forecasts to prevent potential losses and damages. The purpose of this thesis is to construct a classification system for precipitation type prediction. In line with this goal, a method of representing and preparing input data for the learning classifier is developed as well.

Many attempts have been made through centuries to forecast weather. Begin-

ning with early civilizations, people predicted the seasonal changes by monitoring astronomical and meteorological events [29]. Although forecasters could make predictions based on observations and weather lore, it is evident that as more interests in the area has been developed, observation of nature became insufficient for accurate forecasting, and that sophisticate techniques are required to better understand atmospheric conditions.

Several instruments have been invented to measure the atmospheric properties such as thermometer to measure temperature, hygrometer to measure humidity, and barometer to measure pressure. In modern weather forecasting, more weather information is collected through radiosondes or weather balloons – instruments for collecting weather data including humidity, temperature, and pressure at high altitude, and sending them back to ground stations. These measurements are also known as soundings. After the radiosonde is launched, the data are used for weather analysis and to construct computer models for weather forecasting. In our study, the soundings are important as sources of data to train both the classifier and the data fitting algorithms discussed later in this thesis.

An automated weather forecast has been made possible by current modern technologies such as powerful computers, fast telecommunications, and advanced observing satellites. Meteorological data are collected over a wide area from many observations through sensing systems such as radars, satellites, and other meteorological sensors and instruments [29]. They are then sent via communication networks for analysis and compilation by forecasters.

An automated weather forecast can be made by the insertion of observed meteorological data into a computer model which predicts the future state of the atmosphere. Weather types, including precipitation, can be identified using a classification system, which deploys the atmospheric parameters obtained from the numerical model to classify the weather conditions into categories. The required parameters to the system depend on the design of the application and the availability of data. At this point, the accuracy of the automated forecast will depend on both efficiency and performance of numerical models and classification process.

As a part of automated weather forecast, the task of constructing an accurate

classification system is necessary and definitely important. To accomplish this task, firstly, data preparation has to be made to give understandable information to a computer before being used as inputs of the classifier. Then the classifier is built and trained. The process of constructing the classification system for precipitation type classification will be discussed in this thesis.

1.2 Task description and thesis objective

1.2.1 Data representation

It is a challenging task to represent raw data from real-world problems in machine learning. In some cases, data are well constructed through observation processes and are ready for use; however, in many applications the knowledge retrieved from raw data must be preprocessed and represented in the format that machines understand and be able to use for learning. Typically, there are two types of attribute values we represent the data: numeric and symbolic values. Numeric values can either be continuous values (floating point values) or discrete values. Symbolic values represent an interpretation of the observed data. For example, observed temperature values can be expressed as hot, mild, or cold symbolic values by pre-defining that the warm and hot temperature level is the temperature above 30 °C, the mild temperature level is the temperature range between 15 °C and 30 °C, and the cold temperature level is the temperature below 15 °C. Then, we use these attribute values, called features, as description of inputs to learn a function that generates desired outputs.

In classification problems, it is essential to extract and construct attributes from the observing data. These attributes must be important for discriminating the events and representing the raw data either in numerical or nominal form which is meaningful for learning. The accuracy and efficiency of classification depends on the quality of the attributes used in learning.

In this thesis, we are interested in classification of a precipitation type given a vertical temperature profile (sounding data). The vertical temperature profile de-

scribes temperature and dew point temperature values for different pressure levels. Nevertheless, it is not a trivial task to represent the profile for machine learning techniques. One difficulty is how to select pressure levels that provide significant information about temperature and dew point temperature which are sufficient for predicting the precipitation type.

In general, a limited number of mandatory levels and significant pressure levels is used to represent the vertical temperature profile. Mandatory levels are fixed in numbers and locations, but significant levels are varied in both properties; thus, they cannot be used directly for learning. Instead of using a standard set of pressure levels, an optimal set of pressure levels can be retrieved by using an efficient optimization method. An objective of this thesis is to utilize machine learning methodology in selecting a set of significant pressure levels by using available sounding data. For this reason, an application of genetic algorithm (GA) is proposed to find this optimal set of pressure levels that are significant for predicting type of precipitation.

1.2.2 Precipitation type classification

The primary goal of the study is to construct a classification system to predict precipitation. The system requires atmospheric parameters from observations or numerical model outputs along with their corresponding precipitation types as pairs of input and desired output for training the classifier. Generally, sounding data does not inform us of the weather type occurring during the period of radiosonde observation. Hence, our first mission is to construct a training data set which provides both information of examples and their corresponding classes. Next, the classifier must be designed properly to predict the precipitation type. A neural network is considered as a precipitation type algorithm because of its ability to learn adaptively and to model complex relationships between inputs and outputs.

In supervised learning, given a set of examples containing description of inputs (features) and desired outputs, a learning function is optimized to generate outputs for unseen inputs. A classification problem is described as a problem whose output is a symbolic value (class). For example, given descriptions of a mushroom such as

cap shape, cap color, stalk surface, ring type and so on; a system classifies whether the mushroom is poisonous or edible. When the classes of interest are two classes between true and false values, we call these problems as binary classification problems. Likewise, problems that have more than two classes are called multi-class classification problems.

The precipitation types of interest in this thesis are snow, rain, and freezing rain. These events do not occur uniformly, so it is crucial to capture the rare events which have potential to cause damages. In precipitation type classification, the number of examples in freezing rain class is very small compared to the other classes which are represented with a large numbers of examples. In this case, a classifier tends to predict samples as the majority class over the minority class, freezing rain. We have to ensure that the measurement metric are sufficient to measure the overall performance of the classifier. Also, a method to handle the rare events should be applied in order to solve this problem.

1.3 Approach

In this study, we apply machine learning techniques to the problem of weather type classification, specifically the precipitation types of rain, snow, and freezing rain. The atmospheric parameters used are the vertical temperature profiles of sounding data: temperature and dew point temperature. Three contributions are presented including (1) feature construction using genetic algorithms which select the important pressure levels from the sounding data, (2) a neural network model as a classifier to predict the precipitation types, and (3) the use of an under-sampling method to handle the rare class problem, freezing rain.

The data used in this study are vertical temperatures and dew point temperatures from sounding data obtained from radiosonde database access, produced by NOAA's National Climatic Data Center (NCDC). Most of the data are available at significant and mandatory pressure levels. Since the data are available at different numbers of pressure levels depending on the height reached by the balloon, our

first task is to find the vertical levels that fit all data instances in such a way that the temperatures at these levels are significant for the classification task e.g. pressure levels whose temperatures undergo significant changes.

In this research, we deploy the genetic algorithms (GAs) to find the significant vertical levels since GAs is capable of rapidly searching for optimal solution in a large problem space. Possible sets of pressure levels are represented by chromosomes which undergo the evolutionary process. As a result, the evolutionary process will provide the optimal set of pressure values which will be later used to obtain temperature and dew point values as the attributes for the classification process.

Once temperature features are extracted from the available sounding data, the neural network is used to predict the precipitation types. The architecture of the network composes of input layer, one hidden layer, and output layer. The number of input units is the same as the number of the features and the number of output units is the same as the number of classes. The number of hidden units is selected empirically by observing performance on the development set. The network has been trained with training data until a stopping criteria is achieved.

In this thesis, an under-sampling technique is also explored to handle the imbalanced class problem. This method will randomly eliminate some examples of the majority classes to balance class distribution. As a result, a data set with balanced class distribution is obtained to train the classifier. A better performance is expected by applying this method.

1.4 Overview of this thesis

The thesis will be described as follows: background and description of related algorithms, meteorological data and its use in precipitation prediction will be described in Chapter 2. In Chapter 3, design of genetic algorithms to search for the optimal set of vertical levels will be provided. Chapter 4 will describe the use of the neural network to classify precipitation type as well as the implementation of random under-sampling method to handle the imbalanced class problem. The evaluations

of the proposed methods will be given in Chapter 5. Lastly, the conclusion and possible further studies will be explored in Chapter 6.

Chapter 2

Background

2.1 Vertical temperature profile data and precipitation type classification

A vertical measurement of atmospheric conditions aloft from sea level to an altitude approximately 30 km above the sea level can be done through radiosonde. The radiosondes are launched from radiosonde stations twice daily, at 0000 and 1200 UTC [20]. These weather balloons measure various meteorological parameters including temperature, pressure, moisture, and wind information at various atmospheric levels. Nearly continuous stream of information, called radiosonde observation (RAOB), is transmitted back to a ground-based receiver during the balloon ascending. After the radiosonde has been launched, the upper air stations report the radiosonde observation data for certain pressure levels to the National Meteorological Center for analysis and for use in numerical weather prediction models. Only the temperature and dew point data for mandatory pressure levels and significant pressure levels are encoded for transmission because at these pressure levels, temperatures and dew point temperatures have significant changes detected from the sounding plot [20]. The examples of soundings are shown in Figures 2.1 and 2.2 [1], corresponding to rain and freezing rain from St.Johns station respectively.

[1] The figures are taken from [30]

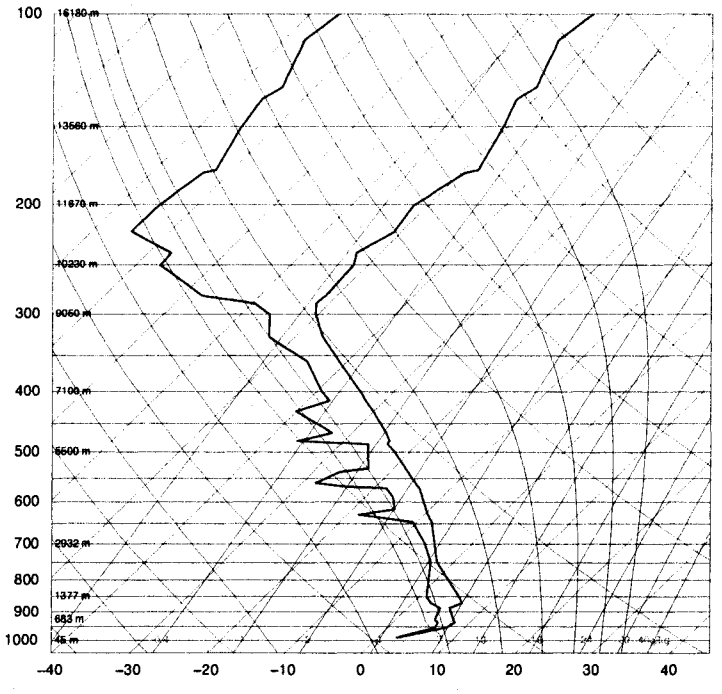


Figure 2.1: A sounding example of rain (00:00,01/05/2007)[30]

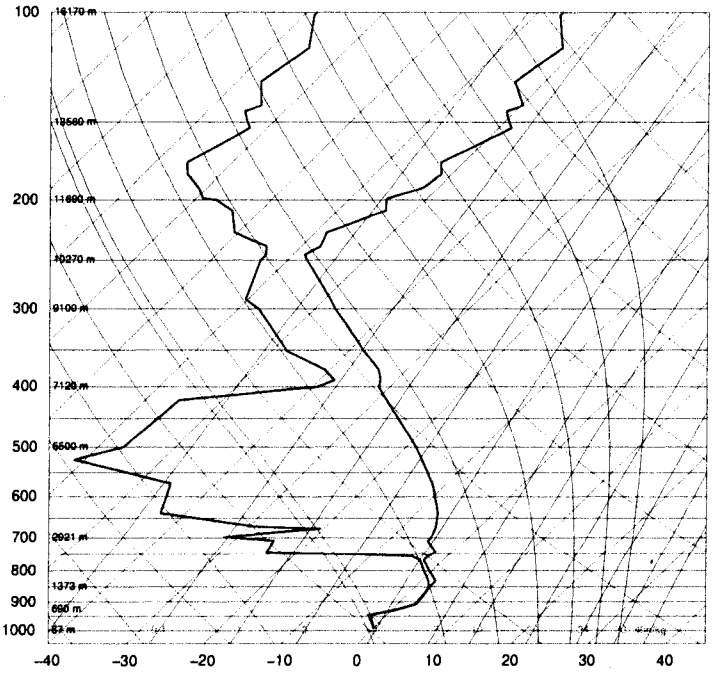


Figure 2.2: A sounding example of freezing rain (00:00,20/04/20)[30]

Many methods have been proposed to determine precipitation types using the vertical thermodynamic data. Some algorithms will be briefly described here.

Bourgouin (2000) [3] introduced an area method for diagnosing precipitation type. The fundamental idea of this method is to determine the areas created by a vertical temperature profile as illustrated in Figure 2.3 where the precipitation type can be identified by determining the area in the plot. When the temperature falls through the freezing layer and develops the area in warm layer aloft and the area in cold layer near the surface, as in Figure 2.3.a, freezing rain and ice pellets are possible. If the area above the freezing layer (warm layer) aloft is too small, the freezing rain and ice pellets may not be formed, instead the precipitation will be snow. This is because of the fact that there is no sufficient warming energy to melt hydrometeors in the warm layer aloft. In Figure 2.3.b, precipitation types are varied, depending on the size of the area developing above surface and aloft. It may produce rain, snow, freezing rain, and ice pellets ,or a mix of them. Regarding the requirement of the area above freezing layer to produce liquid precipitations, rain is expected if temperature is greater than 0 °C and a large area occurs above the surface as in Figure 2.3.c; however, if the area is small, snow or mix of rain and snow can be produced instead of rain. Figure 2.3.d shows a case of snow, since the temperature along all pressure levels is below a freezing layer.

The area is computed as follows

$$c_p|Area| = c_p \oint T\, d\, \ln\theta = c_p\, \bar{T}_l\, \ln(\frac{\theta_{top}}{\theta_{bottom}})$$

where c_p is the heat capacity at constant pressure, T is the absolute temperature, θ is the potential temperature, θ_{top} is the potential temperature at the top of the layer, θ_{bottom} is the potential temperature at the bottom of the layer, and $\bar{T}_l$ is the average temperature in the layer extending from θ_{top} to θ_{bottom}.

By defining the area between 0 °C and the environment temperature above the freezing layer as positive area and the below freezing layer as negative area, the discriminating criteria between precipitation types can be defined as follows.

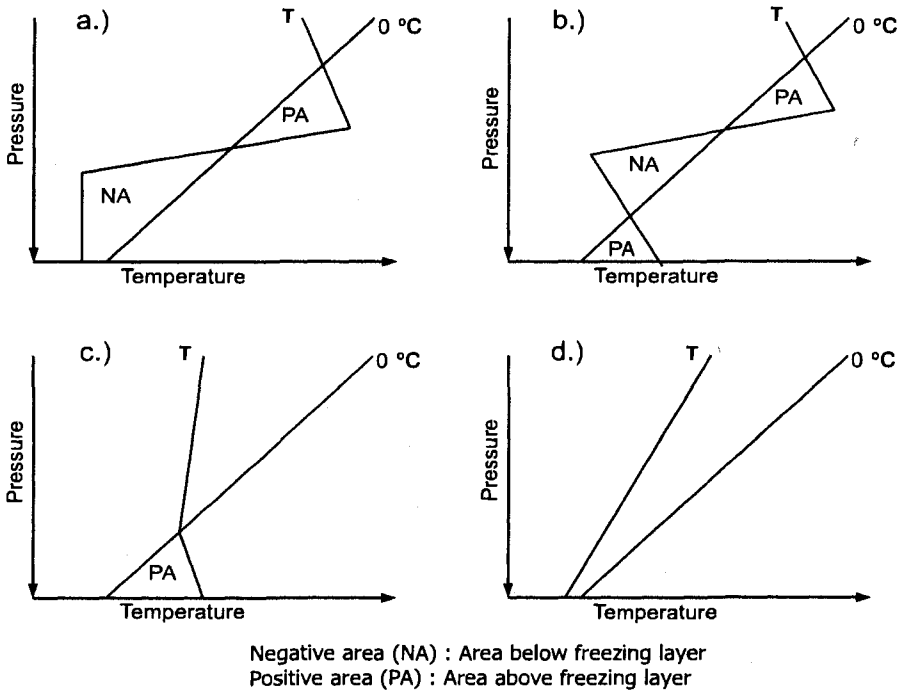


Figure 2.3: The samples of possible occasions of the vertical temperature profiles which produce the area.

Freezing rain and Ice pellets.

Negative and positive areas can be used as criteria to discriminate ice pellets from freezing rain. When plotting freezing rain, ice pellets, and mixed freezing rain and ice pellets as a function of negative and positive areas, a separating equation that represents the equal likelihood of freezing rain and ice pellets can be written as follows.

$$NA = 56 + 0.66PA \tag{2.1}$$

Ice pellets are identified with a larger negative area (NA), so the larger area of NA than the one given by equation 2.1 for a specific positive area (PA) will result in ice pellets. On the other hand, a smaller value of NA will result in freezing rain. However, there are transition zones above and below the separation line where both freezing rain and ice pellets can be observed, so by adding and subtracting 10 Jkg^{-1} to the equation to cover the transition zone, ice pellets can be determined if

$NA > 66 + 0.66PA$, freezing rain can be determined if $NA < 46 + 0.66PA$, and both freezing rain and/or ice pellets can be determined if $46 + 0.66PA \leq NA \leq 66 + 0.66PA$

Rain and Snow

The precipitation type of rain, snow, and mix of the two are likely to be found under the following conditions.

Snow, if PA < 5.6 J kg^{-1}

Snow and/or rain, if 5.6 J $kg^{-1} \leq$ PA ≤ 13.2 J kg^{-1}

Rain, if PA > 13.2 J kg^{-1}

James Ramer (1993) [38] developed a method to determine precipitation types by using temperature (T), relative humidity (RH), and wet-bulb temperature (T_w) on different pressure levels. The algorithm firstly checks T_w. If it is > 2 °C at the surface, the precipitation will be rain. T_{wS} represents the minimum wet-bulb temperature and snow is expected if $T_w < T_{wS}$. In the study, the author used T_{wS} = -6.6 °C.

If these conditions are not satisfied, the algorithm will find the precipitation generating level by using the bottom pressure p_1 and top pressure p_2, and minimum relative humidity R_n. Possible generating layer can be found when $p_1/p_2 > \delta_g$ and $R_n > R_g$ (in the study δ_g is a constant = 1.02 and $R_g = 0.90$) where the highest considered layer is $\leq$ 400 mb. If the generating level is needed but cannot be identified, the algorithm will terminate and no precipitation type is identified. If the generating level is identified, the algorithm begins to calculate the ice fraction at the surface. Before doing so, the algorithm checks if $T_w < -6.6$°C at the generating level, and $T_w < 0$ °C at all other levels below the generating level, snow will be expected. If $T_w \geq -6.6$°C at the generating level, the precipitation is rain ($I = 0$), otherwise, it will be snow ($I = 1$). When the precipitation is mixed between liquid and solid phase ($0 < I < 1$), the ice fraction is determined by

$$\frac{dI}{dln(p)} = \frac{(0\,°C - T_w)}{E} \tag{2.2}$$

where p is pressure, and E is melting energy that is $E = RE_0$, where R is the relative humidity and E_0 is an adjustable constant, $E_0 = 0.045$ °C in the study. Once the value of I is calculated, precipitation type can be determined by using I and T_w values at the surface. The thresholds for I_0 and I_1 are set at 0.04 and 0.85 respectively. If $I > I_1$, a solid precipitation type is predicted and if $I < I_0$, a liquid precipitation type is predicted. Therefore, I value between I_0 and I_1 indicates a mixed precipitation type. If $T_w < 0$ °C at the surface for liquid and mixed precipitation types, freezing rain is diagnosed.

Baldwin et al.(1994) [1] developed an algorithm which uses a decision tree to determine precipitation types. The algorithm has been applied to National Meteorological Center's (NMC's) mesoscale Eta Model. This algorithm is referred as BTC algorithm[2]. It classifies precipitation as rain, snow, freezing rain or ice pellets from thermodynamic vertical profile. The algorithm starts by checking the initial state of precipitation if it is supercooled water or ice. The area between wet-bulb temperatures T_w and 0 °C is used to identify the cold and warm layers for a particular location. The area is used along with the surface temperature to determine the precipitation type [40].

To simplify the algorithm, following variables are used to describe atmospheric conditions for further use in describing the algorithm.

T_C : Coldest temperature at any level with a pressure $\geq$ 500 mb

T_{CS} : Coldest temperature in a saturated layer

T_0 : Temperature at lowest layer

$Area_{-4}^{Tw}$: Area of sounding between -4 °C and T_W

$Area_{0}^{Tw}$: Area of sounding between 0 °C and T_W of the surface-base layer

$Area_{0P_{150}}^{Tw}$: Area of sounding between 0 °C and T_W within the lowest 150 mb

[2]Most of the reviews for this algorithm are described in [40]

The BTC algorithm identifies the precipitation type by the following criteria.

- Snow

  $T_C \leq -4\ ^\circ C$ and $Area_{-4}^{Tw} < 3000$ deg.m

- Freezing rain

  $T_{CS} > -4\ ^\circ C$ and $T_0 < 0\ ^\circ C$

  $Area_0^{Tw} > -3000$ deg.m and $Area_{-4}^{Tw} > 3000$ deg.m and $T_0 \leq 0\ ^\circ C$

- Ice pellets

  If $T_{CS} \leq 4\ ^\circ C$ and $Area_{-4}^{Tw} > 3000$ deg.m, ice pellet is diagnosed when

  $Area_0^{Tw} \leq -3000$ deg.m

  $Area_{0P_{150}}^{Tw} \leq -3000$ deg.m and $Area_0^{Tw} < 50$ deg.m

- Rain

  If $T_0 > 0\ ^\circ C$, rain is diagnosed when

  $T_{CS} > -4\ ^\circ C$

  $Area_{-4}^{Tw} > 3000$ deg.m and $Area_{0P_{150}}^{Tw} > -3000$ deg.m or $Area_0^{Tw} > 50$ deg.m

2.2 Vertical coordinates

In a numerical weather forecast model, it is important to represent the vertical structure of the atmosphere with proper vertical coordinates to achieve better resolution and forecasts [15]. In the same way, vertical temperature profiles can be depicted on the appropriate vertical coordinates instead of the actual pressure and height surfaces to escape the confusion at ground level. Some of the most popular coordinate systems used in current numerical models are Sigma(σ), Eta(η), and Theta(θ) [15]. The Sigma is the coordinate system used in this thesis and a brief description of this system will be given in this section.

Figure 2.4 shows the Sigma vertical diagram with 5 levels. The Sigma coordinate is a ratio of difference between the pressure at a point (P) and the top pressure

(P_T) of the domain to difference between the pressures at ground level (P_G) and the top pressure (P_T). In other words, the Sigma coordinate can be calculated by the following equation:

$$\sigma \equiv \frac{P - P_T}{P_G - P_T} \tag{2.3}$$

Coordinate value ranges between 0 and 1 shown in the diagram. The black section in the bottom of the diagram represents the topography of the landscape, while the vertical axis denotes the pressure in the atmospheric column.

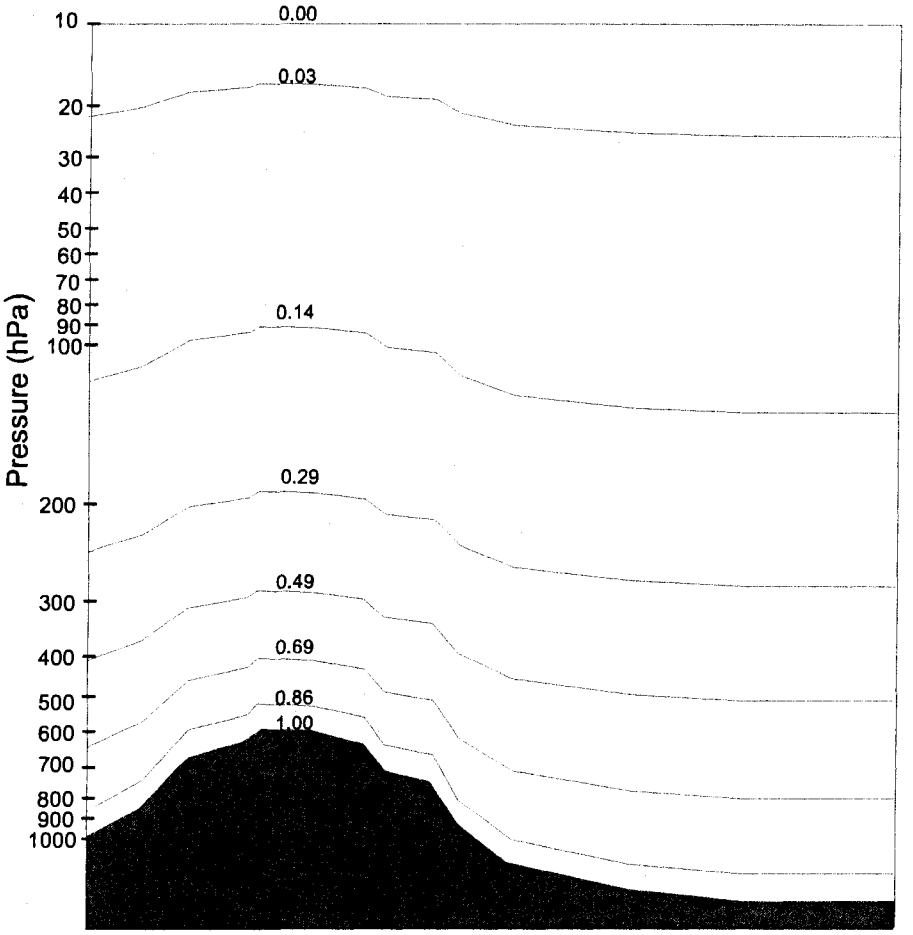


Figure 2.4: Sigma coordinate model system

While the fundamental base of the Sigma coordinate system is at the ground surface, the Eta(η) coordinate system instead is at mean sea level. The eta coordinate system defines the vertical of a particular point in the atmosphere as a ratio

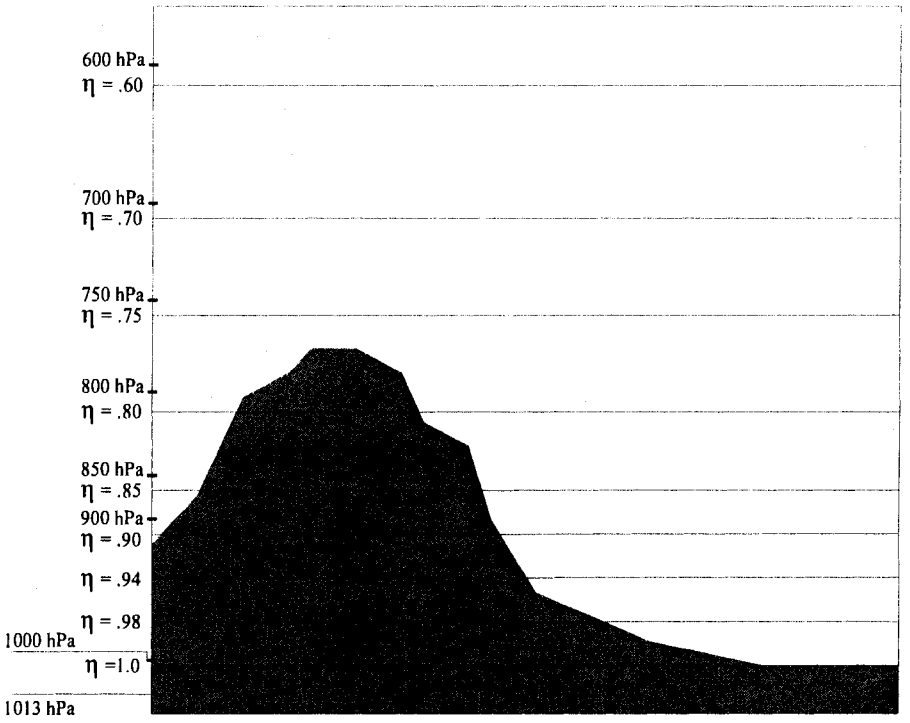


Figure 2.5: Eta coordinate model system

of pressure difference between that location and the top of the domain to pressure difference between the sea level below that point and the top of the domain. Therefore, the eta coordinate system has surfaces that remain relatively horizontal at all times as shown in Figure 2.5.

2.3 Data fitting and piecewise linear representation

The sounding is a graph plotted between pressure and temperature values collected using radiosonde observation. This information is originally nearly continuous; however, during transmission to the National Meteorological Centre for analysis, only temperature profile at mandatory and significant pressure levels are encoded and transmitted. To reconstruct a continuous graph from available data points, one can linearly connect each data point to create the graph. Despite its simplicity, the method may encounter problems from outliers and due to the fact that the underlining function is restricted to a linear function.

A data fitting is a process of finding a function relationship between input and output variables given available data points. We can form this problem in the context of optimization as we are searching for a function that provides the minimum errors given each input x with respect to the output y. The function can be based on linear, polynomial, or other functions. The errors can be based on fitting criteria; for example, with a least-square fitting criterion, the Root Mean Square (RMS) function is used to calculate errors between the generated output y' and true output y. The RMS errors can be calculated using the following equation:

$$RMS = (\frac{\sum_{i=1}^{m}(f(x_i) - y_i)^2}{m})^{\frac{1}{2}} \tag{2.4}$$

where m is a number of available data points, $f(x_i)$ is a generated value from a function f given an input value x_i, and y_i is a true output corresponding to the input x_i. The optimization objective is to find a function $f(x)$ such that it gives the minimum RMS errors. When the function $f(x)$ is in the form of:

$$f(x) = a^T x + b \tag{2.5}$$

This fitting problem can be solved via quadratic programming (QP) [4]. However, it is difficult to solve QP problems in practice due to a large number of variables which depend on a number of data points (m) and degree of polynomials (n).

Alternatively, piecewise-linear functions have been studied to approximate data fitting. A high degree polynomial function can be approximated by several linear functions in several regions that are segmented from the original graph. The problem is to find end points of each segment by connecting these end points using either linear interpolation or linear regression functions. A linear interpolation function connects one end point to another end point with a simple straight line, while a linear regression function finds the best fitting line respecting the least-squares fitting criteria [24]. The objective of this problem is to find the minimum number of segments to achieve errors which are under a threshold (accepted errors).

Many approaches have been proposed for the problem including the dynamic programming-based approach [16], genetic algorithm-based approaches [36, 35]

and convex optimization-based approach [26]. Pedrycz et al., (2004) [35] proposed a Genetic Algorithm (GA)-based technique for piecewise-linear approximations for biomedical data. The result parameters of approximations form features for a classifier; in their case was the pseudo-inverse classifier [10].

2.4 Neural networks

Our study uses a neural network as a classifier to determine precipitation types. Neural networks have been widely used in many machine learning applications, especially in classification problems. This section will introduce neural networks in general and their potential use in precipitation type classification.

An aspect of human intelligence that surpasses computers is that human brains have a mechanism for recognizing objects. Even though a machine possesses high capability of computing complex numeric problems in much less time compared to human brain, it is difficult and requires a lot of resources for a computer to recognize an object using traditional algorithms. This is due to the fact that human brains are very complex, consisting of approximately ten billion neurons interconnected to each other forming networks. A neuron in human brain is composed of a body, an axon, and many dendrites as illustrated in Figure 2.6. Also, a biological neuron connects to thousands of other neurons. These connections of the neurons are similar to the connectivity in a powerful parallel computer. On the other hand, a powerful computer is composed of about ten thousands processing units – much less than those in a human brain. This lack of the number of processing units can be compensated by the speed which computers can process information – approximately million times faster than biological neurons [5].

There have been many studies in machine learning area to make a computer able to learn as humans do. Neural networks are algorithms that mimic human brains. The first neural network was introduced in 1943 by McCulloch and Pitts [28] as a learning algorithm that mimics neurons of human brains. The McCulloch-Pitts neuron is a simple neuron which cannot do much and is composed of a simple computing unit. The simplest neural network called perceptron is based on the

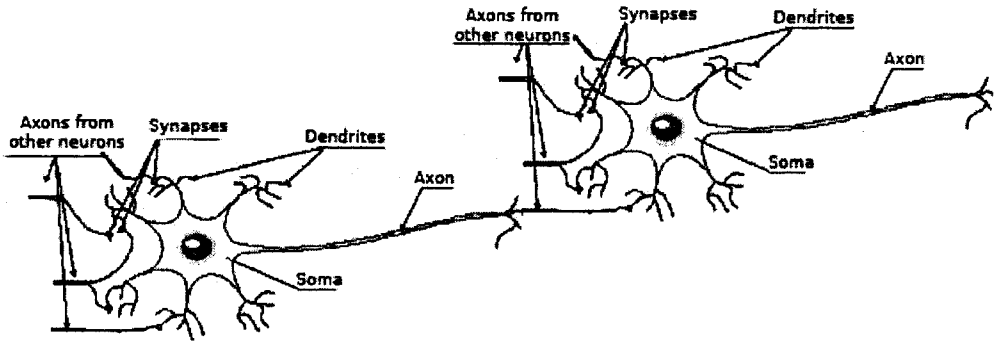


Figure 2.6: Biological neurons

McCulloch-Pitts neuron. It is composed of one input layer of neurons feeding forward to one output layer. At each connection between input and output neurons is a weight indicating how much each input neuron contributes to the output. This architecture is the baseline of current feed-forward artificial neural networks which are also known as multi-layered perceptrons shown in Figure 2.7.

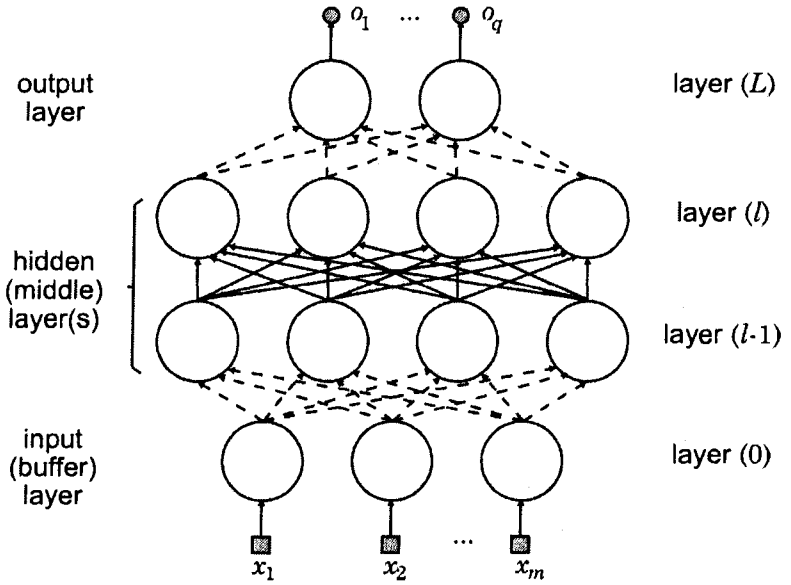


Figure 2.7: Typical architecture of multi-layered neural network

Figure 2.8 shows similarities between artificial neural and a biological neural networks. In biological neural networks, the input signals are sent through dendrites. This process can be compared with the input signals sent through weighted

connections of artificial neural networks. Cell nucleus is where the input signals are processed. It determines whether or not the signals are strong enough to fire the output signal through the axon to the next neuron, and also how strong the output signal will be. This depends on the strength of the stimulus coming into the neuron. Similarly, an artificial neuron computes the sum of product of connection weights and their corresponding input signals to determine if the value is large enough to trigger the output signal. The determination is done by using an activation function or transfer function. Synapses in biological neurons can be represented by the connections of signals sent from one neuron through connection weights of an artificial neural network as the input signal for neurons in the next layer.

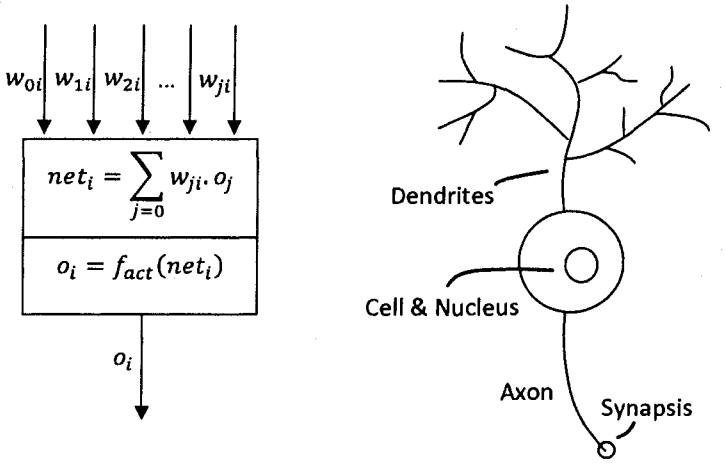


Figure 2.8: Natural neuron vs. artificial neuron

Typically there are three common structures of ANN: single layer feed-forward NNs, multi-layer feed-forward NNs, and recurrent NNs. A single layer feed-forward neural network consists of one input layer and one output layer. The connections are feed-forward from input neurons to output neurons and have no feedback connection. Multi-layer feed forward NNs have input and output layers as same as single layer feed-forward NNs, but also have one or more hidden layers constructed in a feed-forward manner with no feedback connection. Hidden layers can be considered as a black box since they are hidden from outside, unlike input and output layers whose characteristics can easily be observed. Recurrent NNs have at least

one feedback connection. The feedback connections can be self-feedback and feedback to other neurons [5].

There are two basic goals of using artificial neural networks. Researchers who are interested the study of how human brains work would use artificial neural networks to simulate and study how brains work to understand human intelligence. This has been stated in [5] as follows.

> *As with the field of AI in general, there are two basic goals for neural network research. Brain modeling: The scientific goal of building models of how real brains work. This can potentially help us understand the nature of human intelligence, formulate better teaching strategies, or better remedial actions for brain damaged patients. Artificial System Building: The engineering goal of building efficient systems for real world applications. This may make machines more powerful, relieve humans of tedious tasks, and may even improve upon human performance. These should not be thought of as competing goals. We often use exactly the same networks and techniques for both. Frequently progress is made when the two approaches are allowed to feed into each other. There are fundamental differences though, e.g. the need for biological plausibility in brain modeling, and the need for computational efficiency in artificial system building.* (Bullinaria, L1-11)

One of the most powerful features of neural networks is their ability to learn and generalize from a set of training data. The connection weights are adjusted until the final outputs are correct. There are many applications where ANNs have been employed and proven to be efficient methods for dealing with problems including pattern recognition, financial modeling, time series prediction, bioinformatics, etc.

2.4.1 Neural network training

There are two major types of neural network when we consider supervised and unsupervised learning. For supervised learning, the desired output of each input is

specified; the network is trained to map the input to its desired output. The target of the network is to minimize error. An example of supervised algorithms in neural network includes Hebbian learning and error correction learning. There are two categories of supervised learning: structural and temporal. The structural learning attempts to find the relationship between a given input and output. It includes the application of pattern classification and pattern matching. Temporal learning involves the learning of sequential problems, so that a previous response is essential for a current response. Some examples of its applications are prediction problems, simulation and control. Unsupervised algorithms do not require a prior knowledge of the input-output relationship but rather use local information in the data to create clusters according to their collective properties [11]. In this study, we focus only on the supervised neural network since we are concerned with the neural network for classification application. The process of learning will be briefly discussed.

Firstly, we would like to describe offline and online learning. Offline training is used in many applications. It requires all patterns to be present in training. The learning may involve several iterations to achieve the satisfied conditions such as a minimum error or a number of iterations is reached. The learning algorithm adjusts the weights in each iteration corresponding to the error of the previous iteration; for example, back-propagation algorithm adjusts the connection weights of the multi-layers NN, until it achieves a minimum required error or a maximum number of epochs has been reached. After the network has learned, the weights are stored and the network can be used to recall the patterns. A drawback of offline learning is that new training patterns cannot be learned after the network has been trained. Therefore, if we would like to incorporate the new knowledge, it must be added into the training set and the network must be trained again. Although offline learning is mostly used, online learning has been used in some neural networks.

The online learning incorporates new patterns without re-learning the entire set of training data, but the network can learn the new patterns immediately without loss of stored knowledge [11].

Knowledge of the network is stored in the connection weights. How the weights of network are adjusted until the desired output has been achieved will be discussed

in this section. Basically, the learning process of a neural network can be described as the following steps.

- **Initialization:**

  Before training patterns are presented to the network, the connection weights of the network must be initialized. Generally, the value between (0, 1) or (-1, 1) are randomly chosen as the initial weights and this range of values usually work well. Some implementations initialize weights at random locations in a unit hyper-sphere. However, the choice of the initial values of weights depends on the problem and normalization of variables. [27]

- **Output determination:**

  Each training pair of input and output is presented to the network. Once the input pattern is applied to the network, a sum of products of input elements and their corresponding weights is calculated. The output can be determined by applying the activation function to the total net sum. The expression is shown below.

$$Output = f(\sum(x_i w_i + bias)) \tag{2.6}$$

  where f is the activation function

  Figure 2.9 illustrates when an input pattern is applied to a neuron. The output is feed-forward to the next layer.

- **Error correction:**

  The final output of the network will be compared with the desired output. The difference between these two values is the error which will be used to adjust the connection weights. Weight adjustment is determined in proportion to this error. The method to determine the difference between the output and the desired output depends on the algorithms used in the network. For example, the back-propagation uses the gradient descend method which moves the weight adjustment to the opposite direction of the error, downward on

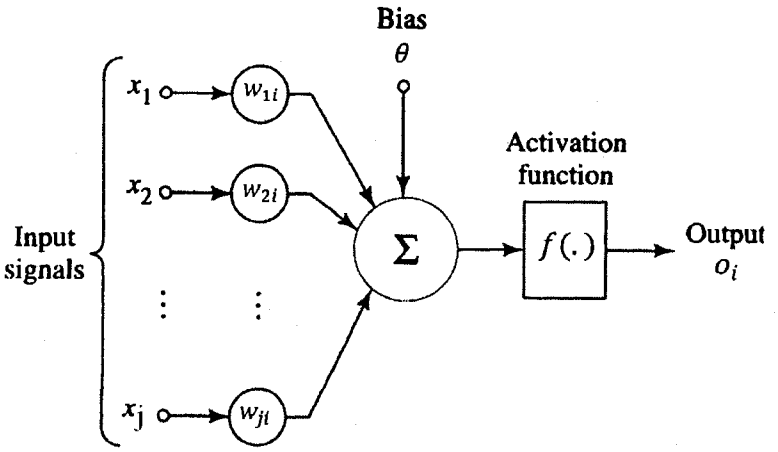


Figure 2.9: Feed-forward NN

the error surface to minimize the error. Cumulative error in back-propagation neural network is described as follows:

$$E_c = \frac{1}{2}\sum_{k=1}^{n}\sum_{i=1}^{q}[t_i(k) - o_i(k)]^2 \tag{2.7}$$

The weight adjustment can be determined by

$$\triangle w = -\eta\frac{\partial E(k)}{\partial w} \tag{2.8}$$

When a sigmoid function is used as the activation function, weight adjustment can be written as follow:

$$w_{new} = w_{old} + \triangle w \tag{2.9}$$

$$\text{For output layer :} \quad \triangle w_{ij}^L = \eta\delta_i^L o_j^{L-1} \tag{2.10}$$

$$\text{Where :} \quad \delta_i^L = (t_i - o_i)f'(tot_i) \tag{2.11}$$

$$\text{For hidden layer(s) :} \quad \triangle w_{ij}^L = \eta\delta_i^l o_j^{l-1} \tag{2.12}$$

$$\text{Where :} \quad \delta_i^l = f'(tot_i^l)\sum_{p=1}^{n_i}\delta_p^{l+1}w_{p_i}^{l+1} \tag{2.13}$$

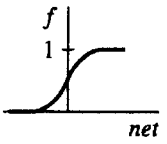
2.4.2 Activation function

The activation function used in neural networks varies by types of applications. A particular type of function may be appropriate for only some applications and it may not be suitable for other applications. The common activation functions in use with neural networks will be discussed as follows.

- **Sigmoid function**

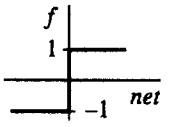
  Sigmoid function or logistic function is a nonlinear and differentiable function used in applications that require the mapping of non-linear process. It is commonly used because it has a simple derivative and limited range. The sigmoid function can be written in the following expression:

$$f = \frac{1}{1 + e^{-\alpha net}} \tag{2.14}$$



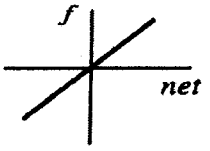
- **Hard limiting**

$$f = \begin{cases} 1, & \text{net} \geq 0; \\ -1, & \text{net} < 0. \end{cases} \tag{2.15}$$



- **Linear function**

$$f = net \tag{2.16}$$



2.4.3 Data scaling

The characteristic of data obtained from a particular problem plays a role in the design of the network architecture. For example, a number of attributes or features in classification task can determine the number of the network inputs and the number of classes can determine the number of output neurons. In addition, a structure of the network also depends on how the data is encoded e.g. binary encoding, real number encoding, and so on.

Several neural networks need the data to be scaled before processing. The choice of scaling methods depends on the problem and can affect the performance of the network. The group of input vectors to be scaled together is also varied. Some implementations scale all input vectors together into the same scale while some applications scale each attribute of the data set separately or scale similar attributes together.

The following is the typical size scaling method:

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}} \tag{2.17}$$

2.5 Genetic Algorithms (GAs)

Genetic Algorithm is a heuristic searching method inspired from evolution biology and natural selection that the fittest offspring will survive to the next generation. At each cycle of the evolution, genetic algorithms basically include these operators: inheritance or reproduction, mutation, and recombination to change the gene pool of a population over time in order to obtain a better generation. A computer simulates evolutionary operations in order to find the optimal solution by representing the possible solutions in chromosomes and evolving to better solutions. Typically, a solution is encoded with a string of binary numbers; however other types of encoding are also possible. The advantage of GA is that it can rapidly search a large, poorly understood problem, and it is excellent for tasks that require optimization. It is also highly effective to solve problems that have a vast number of possible solutions.

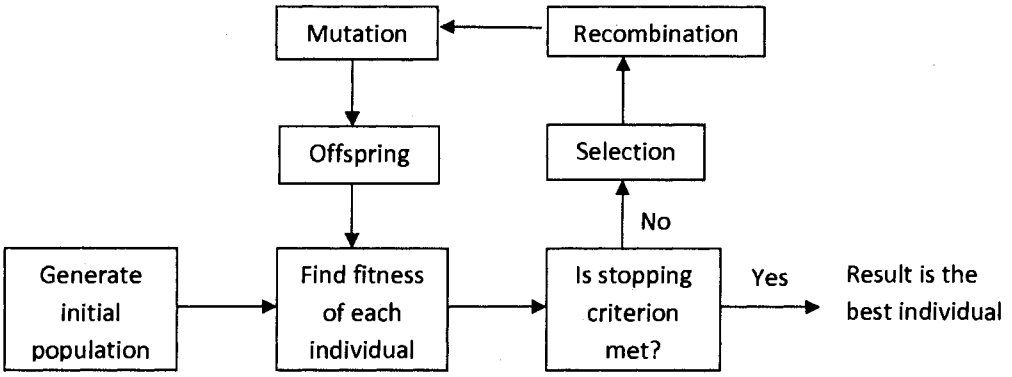


Figure 2.10: Typical genetic algorithm process

The evolutionary process of simple Genetic algorithms is illustrated in Figure 2.10, starting with the initialization of a population representing the set of possible solutions. Generally, the first population is randomly generated at a required number of population that has been defined in advance. The randomly generating method for the first population makes it possible to have initial solutions posted throughout the problem space.

After the initial state, each individual is evaluated and assigned with its fitness value. The objective function must be defined beforehand as a function to identify how much an individual fits to the solution of the problem. Then, multiple individuals are selected based on their fitness. The greater the fitness to the objective function, the better the chance to be selected. Next, pairs of parents are selected to recombine. The crossover/recombination changes the gene pool of the mates and then offspring are created. The new generation is evolved to the next generation by the same procedure and iterates until the stopping criterion is met. The summary of simple genetic algorithms is shown below.

1. Population is randomly generated.
2. The fitness value of each individual in the population is calculated
3. Select two chromosomes as parents from the population according to their fitness values

4. Perform crossover and mutation and obtain offspring to the next generation

5. Calculate individual fitness value of offspring.

6. Place new offspring in a new population

Repeat step 3 to 6 until stopping conditions is satisfied

The following section will give a brief explanation of operators and methods in typical genetic algorithms.

- **Selection**

  There are some selecting methods commonly used in the application of genetic algorithms. They are listed as follows:

  - Roulette-wheel selection

    Each individual is given a probability of being selected according to its fitness value. A higher fitness value indicates a higher chance to be selected for reproduction. Similarly to a roulette wheel, the whole area of the wheel represents total probability. The portion in the wheel is assigned to each individual according to its probability value. Once every individual has been assigned with its portion, a random number is generated to select the individual whose portion spans over such random number. The process is repeated until the required number of selected individuals is achieved.

  - Ranked-roulette wheel selection

    This method of selection is similar to roulette-wheel selection. The only difference is that individuals are ranked base on their fitness values and portions of the wheel are assigned to individuals based on the ranking positions instead of their fitness values.

  - Tournament selection

    The set of individuals are randomly selected from the population and the fittest individual is selected as the parent. The process repeats until

the number of population is satisfied. It is called tournament since the only winner will be chosen as a parent.

- **Crossover**

  Crossover combines the information of two or more chosen individuals (parents), which have been selected in the selection process, to create new individuals. It can be done by combining the values of the parents. The choice of choosing crossover method depends on the representation of the chromosomes which represents the solution of a particular problem. Some of crossover methods are listed here.

  - Single point/Multiple points crossover

    This crossover method locates the variables in the parents for the required number of points and exchanges the segments of the parents that have been separated by the points. For example, in the single point crossover method, parent chromosomes 101101 and 111100 might have a randomly cutting location at the third bit – written as 10|1101 and 11|1100 which result in the offspring of 101100 and 111101.

  - Uniform crossover

    While single/multiple point crossover methods select a separated location(s) for producing offspring, the uniform crossover generates a set of bits indicating which bits will be exchanged. For example, the parents are 101101 and 111100. The first set of generated bits is 110010 and the second set of generated bits is 111101. Thus, the result of the offspring of the example parents are: 111101 and 111100.

  - Intermediate crossover

    Unlike the previous two crossover methods, the intermediate crossover method is mainly used for real-valued individuals. Some of intermediate crossover are described as follows:

Simple intermediate recombination

Parents are x_1,..., x_m and y_1,..., y_m

$Child\ 1\ :\ x_1, ..., x_k, \alpha x_{k+1} + (1-\alpha)y_{k+1}, ..., \alpha x_m + (1-\alpha)y_m$

$Child\ 2\ :\ y_1, ..., y_k, \alpha y_{k+1} + (1-\alpha)x_{k+1}, ..., \alpha y_m + (1-\alpha)x_m$

Single intermediate recombination

Parents are x_1,..., x_m and y_1,..., y_m

$Child\ 1\ :\ x_1, ..., x_{k-1}, \alpha x_k + (1-\alpha)y_k, x_{k+1}, ..., x_m$

$Child\ 2\ :\ y_1, ..., y_{k-1}, \alpha y_k + (1-\alpha)x_k, y_{k+1}, ..., y_m$

Whole intermediate recombination

Parents are x_1,..., x_m and y_1,..., y_m

$Child1 : x_i^{new} = \alpha x_i^{old} + (1-\alpha)y_i^{old}, i = 1, 2, 3, ..., m$

$Child2 : y_i^{new} = \alpha y_i^{old} + (1-\alpha)x_i^{old}, i = 1, 2, 3, ..., m$

- **Mutation**

A mutation process alters some parts (bits or values) of the offspring that evolve at random with a low probability (mutation probability or mutation rate) pre-defined by users. There is no definite rule for defining this parameter; however, normally the mutation probability parameter is defined as 1/n where n is a number of variables in an individual. The mutation rate controls the likelihood of mutation events. For real-valued individuals, the mutation process means that the variable values are changed with randomly-generated values. In the same way for binary-valued individuals, the mutation process means the flipping of variable from 0 to 1 or from 1 to 0 as there are only two possible values for each variable.

2.6 Handling imbalance class method

Very often in classification problems we encounter imbalanced data sets. The imbalance data set occurs when one class in the data set is represented with a small number of data points while other classes are much greater in number [22]. Dealing with imbalanced data is also known as dealing with rare classes, or with skewed data. The rare class problem can be identified by two occasions.

Firstly, the class is rare compared to other classes. We can compare this situation with the phrase "find a needle in a haystack". The difficulty of finding a needle is not because the needle is small, but because of the large number of threads of hay that hinder the searching ability [42]. In the same way, in classification problem, the classifiers are hindered by the overwhelming amount of majority class data.

Secondly, the class is rare because there are very few data points in the data set that causes the lack of the information. It is difficult to find the regularities of the pattern with a small number of data. The learning algorithms cannot generalize the class that does not provide enough information for algorithms to learn.

When dealing with imbalanced data sets, classifiers tend to predict samples as the majority class and ignore the minority class [43]. This problem is mainly due to the following reasons:

Firstly, standard machine learning algorithms are built to achieve overall accuracy which the minority class contributes very little [41]. Considering a network transmission data consisting of 99% of regular transactions and only 1% of intrusions, a classifier can achieve 99% accuracy by assigning all examples to the regular transmissions. If the minority class is the class of interest, saying that the classifier is designed to detect the intrusion, this classifier is futile because it achieves a high accuracy on the majority class but cannot detect the minority class.

Secondly, classifiers usually assume that the algorithms will be employed on data drawn from the same distribution as the training data [41, 37]. In many real-world problems, the training data does not have the same class distribution as the

testing data and they are rarely the same. The training data might be imbalanced but the unseen data might not be and vice versa.

Finally, standard classifiers always assume that the errors from different classes have the same costs. The classification costs for each class in several applications are different. Consider the application of credit card fraud detection, hardware fault detection, medical diagnostics, and so on. The misclassification of the important class in these applications is very costly. If the actual cost of misclassification is known, the correct threshold can be determined for a classifier. [41].

For these reasons, many studies have been proposed to deal with the problem both in data and algorithmic levels. The solution in the data level includes different forms of re-sampling techniques which re-balance the distribution of underlining classes in the training set. Two common re-sampling techniques are under-sampling and over-sampling.

- Under-sampling

  The under-sampling method eliminates the majority class examples to balance the class distribution. A simple method of under-sampling is where the examples of majority class are randomly eliminated. This method is called random under-sampling. A major draw-back of randomly under-sampling method is that the method can eliminate potentially useful data that could be important for the learning process [25]. Other than using the random method to eliminate the examples, the discarded examples could be selected determinedly by choosing the majority examples that are closest to the minority class examples or ones that are far from the decision boundary since these examples are less relevant for learning.

- Over-sampling

  The over-sampling method generates the minority class examples to eliminate the imbalance. The examples can be replicated from the available minority class examples. By this method, the training size will increase without any

gain of information since the new information is replicated from the existing ones. Chawla et al.(2002) [6] introduced an over-sampling method which synthetically generates the minority class examples called SMOTE (**S**ynthetic **M**inority **O**ver-sampling **TE**chnique). SMOTE computes k-nearest neighbors (5-NN in SMOTE) for each minority example and selects some of them according to the re-sampling rate. The new examples are synthetically generated along the line between the selected nearest neighbor(s) and the minority class example. A drawback of this method is that it generates minority class examples without considering the majority class and may cause overgeneralization [43].

With the studies of Japkowicz 2000 [21] and Drummond 2003 [9], it was observed that under-sampling methods perform better than over-sampling methods.

Other than the solutions at the data level, the solutions to imbalanced class problems in the algorithmic level have been explored. Barandela et al. [2] proposed the weighted distance function to internally bias the discrimination procedure. The weights are assigned to the respective classes other than the individual examples in order to compensate the imbalance without changing the class distribution. With the Knn based classification, the weighting factor for the majority class is greater than that for the minority class; therefore, the distance to the minority examples is reduced and the tendency to find the minority class is increased. Pazzani et al. [34] published a paper to minimize classification cost rather than to minimize the classification error since the cost of misclassification is not equal for each class. Similarly, Domingoes(1999) [8] incorporated costs in decision making to define unequal misclassification costs between classes.

2.7 Evaluation Metrics

In a classification problem which involves imbalance data set, appropriate evaluation metrics must be employed to evaluate the performance of the algorithms. This is due to the fact that the accuracy measurement is not sufficient to measure the

performance of a classifier as the overall performance of the classifier is mostly contributed from the majority class rather than minority class. Because of this, additional metrics are necessary. Some measurement metrics will be briefly described here as the evaluation metrics to measure the performance of the classifier.

The classification of imbalanced data problem usually considers the problem as a two-class problem. A multi-class problem can be simplified into two class-problems [25]. The simple way to represent the possible predicting outcome is to represent the decisions in the contingency table shown in table 2.1.

Predict answer	Actual answer	
	Yes	No
Yes	Hit	False alarm
No	Miss	Correct rejection

Table 2.1: Contingency table

Using the terminologies from information retrieval, Hit, False alarm, Miss, and Correct rejection are denoted with True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN), respectively. Regarding the classification problem, the positive class is the class of interest or the minority class in the imbalance class problem. When using these terminologies in the classification problem, TP and TN represent the number of positive and negative examples correctly classified, and FN and FP represent the number of misclassified positive and negative examples respectively.

In the same way, many meteorological phenomena can be considered as two-class or binary event in that the event will or will not take place or we can refer to them as a yes/no problem [23]. The evaluation techniques commonly used in the weather-prediction application includes probability of detection (POD), false alarm rate (FAR), Critical Success Index (CSI), Heidke Skill Score (HSS), and Bias.

- **Probability of detection (POD)**

  Probability of detection, POD, is the proportion of the correctly classified examples. It is also known as a Hit rate. POD can be defined by:

$$POD = \frac{TP}{TP + FN} \tag{2.18}$$

- **False Alarm Ratio (FAR)**

  The false alarm ratio is the proportion of false alarms on the event being predicted or the ratio of the number of incorrect predictions to overall predictions. FAR is defined by:

$$FAR = \frac{FP}{TP + FP} \tag{2.19}$$

- **Critical Success Index (CSI)**

  Critical Success Index (CSI) does not take into account the non-occurrence events and non-predicted events. The CSI is calculated as the ratio of the true positive (TP) and the sum of the false positive (FP), false negative (FN) and true positive (TP) as defined in Equation 2.20. The value is in the range of [0, 1] where the perfect performance score of CSI is 1 when there are no FP and FN cases, and the worst performance score is zero when the TP value is zero. Unlike POD and FAR, the CSI takes into account both false alarms and missed events. If false alarms and miss events are greater, the CSI value is smaller. Therefore, it is a more stable measurement. The CSI is sensitive to the climatology of the events and tends to give poorer scores for rare events [14].

$$CSI = \frac{TP}{TP + FP + FN} \tag{2.20}$$

- **Heidke Skill Score (HSS)**

  HSS is defined by:

$$HSS = \frac{2 * (TN * TP - FP * FN)}{(TN + FP) * (FP + TP) + (TN + FN) * (FN + TP)} \tag{2.21}$$

- **Bias**

  A bias in weather forecast is the ratio of positive forecasts to the number of observed events [33] as shown in equation 2.22

$$Bias = \frac{TP + FP}{TP + FN} \tag{2.22}$$

Chapter 3

Implementation of GAs to find the optimal set of vertical levels

The implementation of Genetic Algorithms (GAs) which search for the optimal set of vertical levels will be described in this chapter. GAs typically start by randomly generating individuals in the population representing a group of possible solutions. Each individual is evaluated by determining its fitness to the objective function. Then, the selection operation selects pairs of individuals based on their fitness to reproduce the offspring to the next generation. The reproduction procedures exchange the gene pools of parents through a crossover operation and adapt genes of children by a mutation operation. These operations develop and introduce the new characteristics of the individuals to a better solution. Finally, the new population is evolved by iterations with the same algorithm until the stopping condition is reached e.g. the required number of generations is attained, the fitness of the best individual converged to a satisfactory level, etc.

We implement our genetic algorithm system approach by using MATLAB. The methodologies as well as an overview of the system to find the optimal locations of pressure levels will be explained in section 3.1.

3.1 Methodologies

The overview of the system is shown in Figure 3.1 and 3.2.

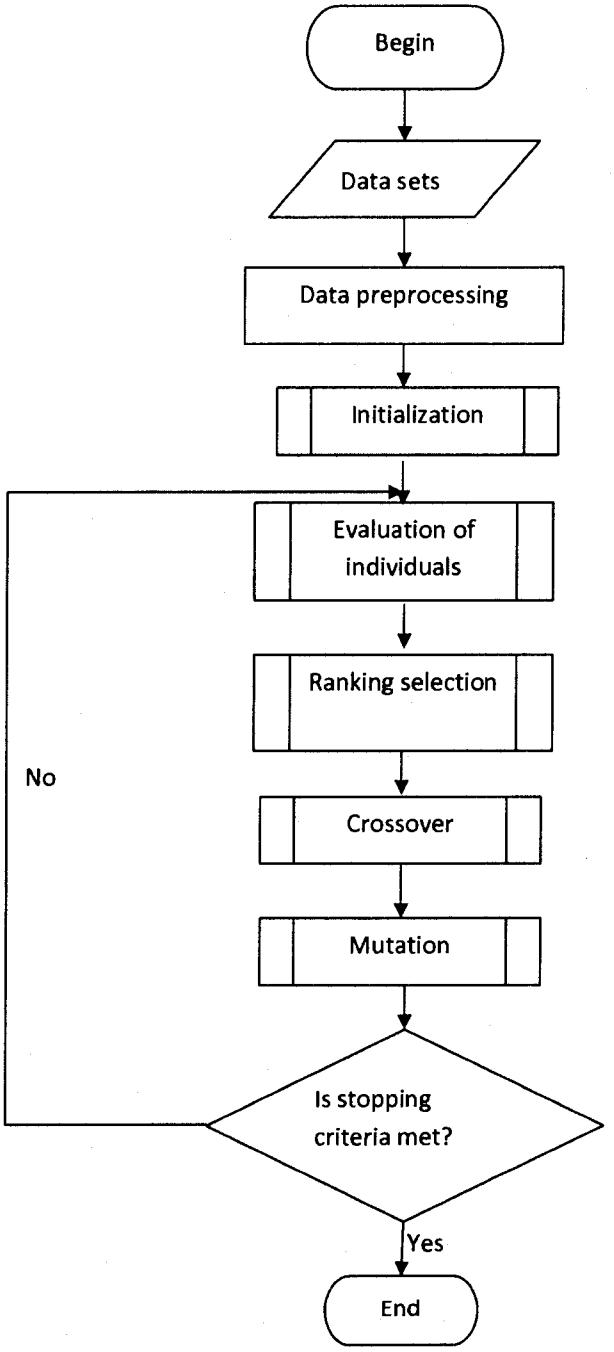


Figure 3.1: Overview of the GA main system

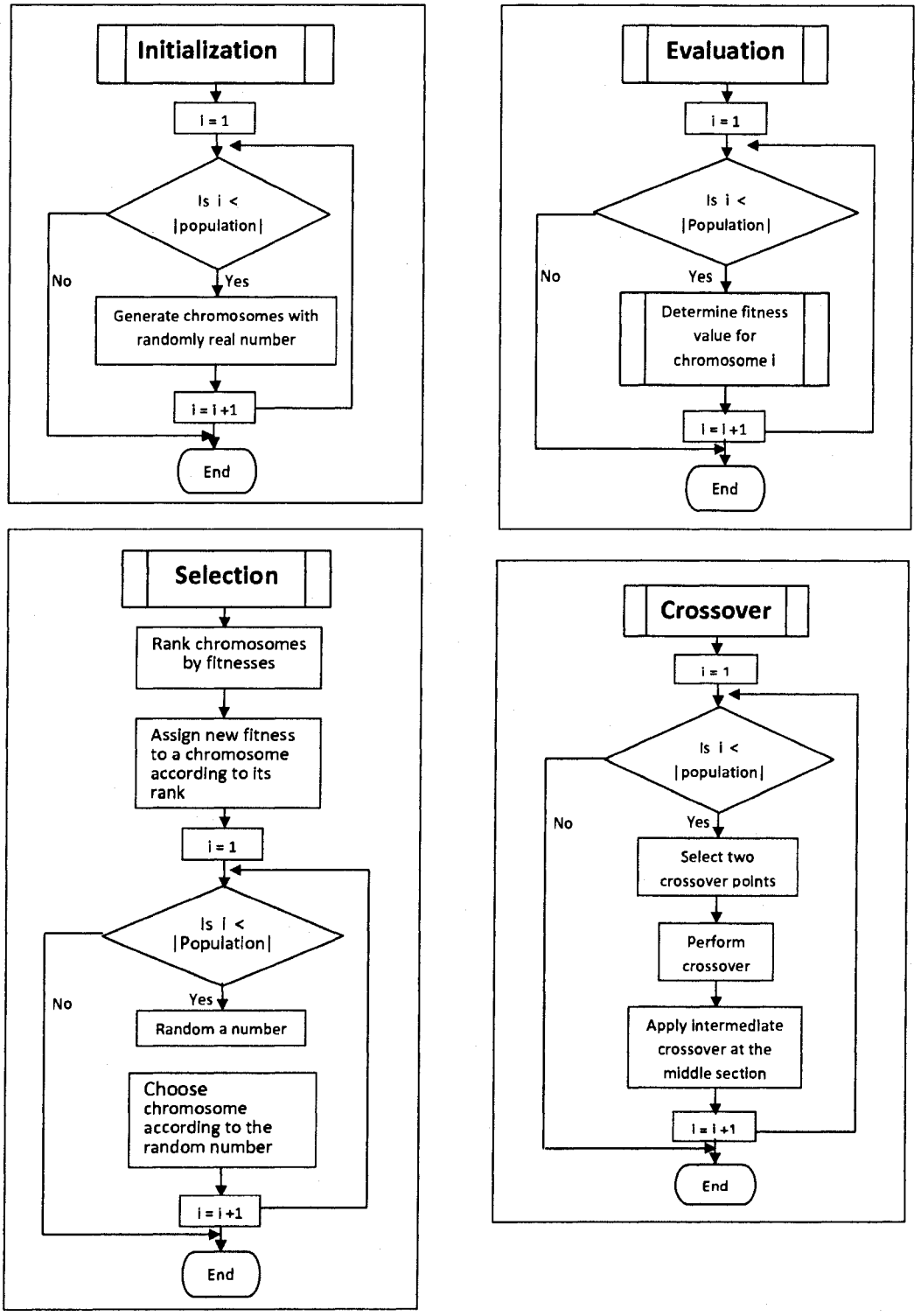


Figure 3.2: Main components of the GA system

3.1.1 Chromosome encoding

An important issue of using the genetic algorithms is how we encode the solution into chromosomes. There are several methods to encode the chromosomes in GAs. Traditionally, binary encoding is used to represent the solution. A binary encoded chromosome is a string of 0s and 1s; each value represents the absence and presence of the searching components of the optimal solution. In our approach, the chromosomes are encoded with real numbers instead of the traditional binary numbers. Each individual is a string of real numbers which represent the possible vertical levels. The encoded values are in the range of (0, 1) indicating vertical levels in the Sigma(σ) coordinate system.

3.1.2 Initialization

In this step, the first generation is obtained. The population number and the length of the chromosomes have been predefined. Chromosomes in the population are the strings of real numbers which are randomly generated. Each number represents the vertical level in the Sigma coordinate system. The length of a chromosome indicates the number of required vertical levels which always includes the ground and the highest levels. The generated chromosomes always contain the vertical level at ground (represented by 1) and the highest level that the radiosonde has been reached (represented by 0).

3.1.3 Evaluation method

The fitness function used to determine the quality of the chromosome is based on how well the chromosome represents the vertical temperature data. The fitness value can be evaluated by using the least-square fitting criteria. Root Mean Square Error (RMSE) is used here. The fitness value is determined by the following procedure; the pseudo code of the fitness function is described in Algorithm 1. The evaluation method of our approach is also illustrated in Figure 3.3. The figure shows the example of a chromosome whose selected vertical levels are at 100, 200, 300,..., 1000 mbar. The vertical temperature profile is constructed by connecting

the linear line between two adjacent temperature values of selected vertical levels. The grey area is the error between the constructed vertical temperature profile (illustrated with the dotted line) and the actual temperature profile (illustrated with the solid line). To determine this error, we calculate the difference between these two lines by discretizing the area into small steps.

Fitness determination of an individual procedure:

1. The temperature and dew point temperature are calculated for each pressure level in a chromosome.
2. For all training data points, the vertical profiles for each chromosome and training data are constructed from available temperature and dew point temperature.
3. The root mean square error is calculated by determining the difference between the chromosome vertical temperature profile and training data vertical temperature profile.
4. 100 equal intervals are used to evaluate this error. The missing temperature profile is calculated by the interpolation between two available levels.

3.1.4 Selection method

After the evaluation process, the chromosomes are scored by their corresponding fitness values. We scale the fitness scores with a ranking method. A ranking roulette wheel selection method is used to select the potential chromosomes. The procedure of the ranking roulette wheel selection starts by ranking the individuals by their fitness values and then receives new fitness values from their ranking positions. For example, the worst chromosome (evaluated by using its fitness value comparing with others in the population) receives the fitness value of 1, the second worst receives the fitness value of 2, and so on, until the best one receives fitness value of n, equivalent to the population number in the problem. These new fitness values will

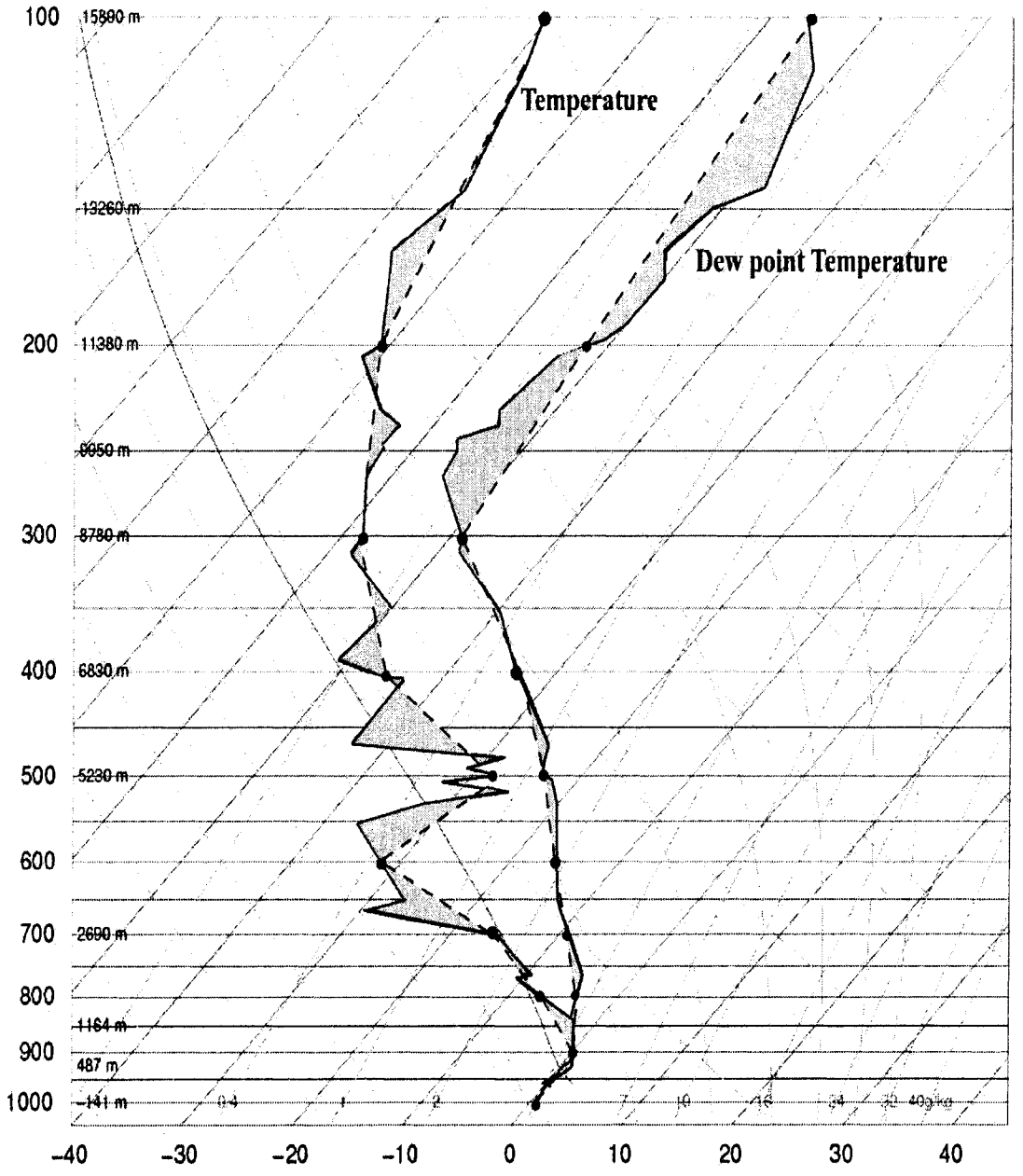


Figure 3.3: Error determination for an example chromosome whose selected levels are 100, 200, 300,..., 1000 mbar

Algorithm 1 Fitness function

Input: Training data consists of pressure, temperature, and dew point temperature (p, t, td), and chromosome c

Output: Fitness value f

1: $ErrorT = 0$
2: $ErrorTd = 0$
3: **for all** training data point (p, t, td) **do**
4: $\hat{t}$ = interp(p, t, c)
5: $\hat{td}$ = interp(p, td, c)
6: **for** $i = 0$ to 1 with stepsize s **do**
7: t_i = interp(p, t, i)
8: td_i = interp(p, td, i)
9: $\hat{t}_i$ = interp$(p, \hat{t}, i)$
10: $\hat{td}_i$ = interp$(p, \hat{td}, i)$
11: $ErrorT+ = \sqrt{(t_i - \hat{t}_i)^2}$
12: $ErrorTd+ = \sqrt{(td_i - \hat{td}_i)^2}$
13: **end for**
14: **end for**
15: **return** $f = \frac{ErrorT+ErrorTd}{N}$

be used as the probabilities of being selected. The chance that an individual will be selected increases with its fitness value. This selection method is done similarly to a roulette wheel method but the difference is that fitness values assigned by ranking are used rather than raw fitness values obtained by the fitness function.

3.1.5 Crossover

Two-point crossover with an intermediate crossover is used for the crossover operation. The selection method selects potential chromosomes to reproduce offspring to the next generation. Pairs of chromosomes are then chosen as parents for the reproduction process. The two crossover points are randomly selected for each pair of parents. Then, the gene pools of parents are exchanged according to the two crossover points.

Example:

Parent 1 = [a b c d e f g]

Parent 2 = [1 2 3 4 5 6 7]

Crossover points (at random) = 2,5

Child 1 = [a b 3 4 5 f g]

Child 2 = [1 2 c d e 6 7]

In addition to a simple exchange of gene pools, the intermediate crossover is employed in the middle section between the two cutting points. The intermediate crossover reproduces new genes by using weighted average of the parents' gene values. This is controlled by a single parameter, called *Ratio*.

$child1 = parent1 + rand * Ratio * (parent2 - parent1)$ [1]

If *Ratio* is in the range [0,1] then the children are produced within the hypercube defined by the parents locations. If *Ratio* is in a larger range, for example 1.1, then children can be generated outside the hypercube. At the end of the crossover procedures, children are obtained. Note that the best individual of each generation will always be copied to the next generation.

3.1.6 Mutation

A uniform mutation is used. A single bit in the chromosome is chosen and a random probability P identifies if the bit value will be modified. The modified bit will be replaced with a new random number. The mutation rate value is set beforehand to determine the frequency of mutation occurrences. The procedure of the mutation is illustrated in Algorithm 2

Algorithm 2 Mutation function

Input: chromosome input c, mutation rate m.
Output: chromosome output.

```
1: for all gene g in chromosome c do
2:    generate uniform random number r
3:    if m > r then
4:       alter value of gene g
5:    end if
6: end for
7: return c
```

[1] The description of intermediate crossover is described in section 2.5

Chapter 4

Neural network for precipitation type classification

4.1 Neural network implementation

We evaluate the quality of the selected vertical levels with the performance of precipitation type classification by using the temperature and dew point temperature values at these vertical levels. We implement a neural network classifier to predict a precipitation type. The implementation detail of our classification approach will be described in this chapter. In addition, the under-sampling method will also be discussed as a method to deal with the unbalanced class problem. The method will be applied to balance the data before training the classifier.

4.1.1 Neural network architecture

In general, the multi-layered neural networks consist of multiple layers. The number of neurons in each layer depends on the problem. Commonly, the number of input neurons in the classification problem is constrained by the number of attributes in the problem. The number of input neurons can also be alternatively designed according to the problem. The number of output neurons is constrained by the number of outputs required by the problem. There is always at least one hidden layer in the network. Research indicates that using one hidden layer with a sufficient number of hidden neurons in the network can well approximate any finite function.

The neural network in our approach consists of three layers: input layer, hidden layer, and output layer. All layers are fully connected in a feed-forward manner. The

input layer is connected to the hidden layer and the hidden layer is connected to the output layer. The topology of the three-layer neural network is shown in Figure 4.1. The implementation description of each layer is provided as the following.

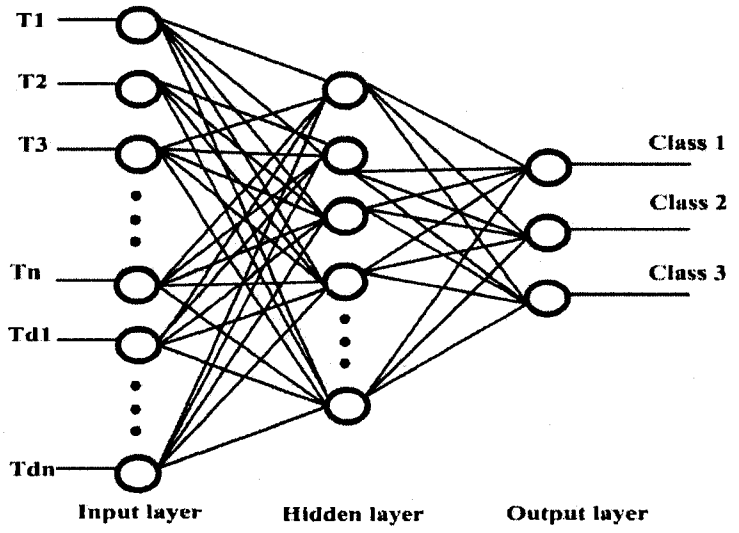


Figure 4.1: Topology of the neural network for precipitation type classification

- Input layer

  The input layer is composed of the input neurons in the same number as there are temperature and dew point temperature values for a set of vertical levels that we would like to evaluate. The number of input neurons is twice the number of vertical levels because we obtain the temperature and dew point temperature for each vertical level in a level set. The activation functions of the input neurons are the linear functions since we directly use the temperatures and dew point temperatures as the features of the domain.

  If n represents the number of vertical levels, the input vector $\vec{X}$ can be written as $\vec{X} = [T_1, T_2, T_3, ...T_n, Td_1, Td_2, Td_3, ..., Td_n]$, where T is a temperature value and Td is a dew point temperature value.

- Hidden layer

  The hidden layer consists of hidden neurons which can be adjusted. The number of hidden neurons will be adjusted according to the problem. It is

unknown beforehand and is necessary to be derived by trial experiments. If the number of hidden neurons is too small, the complexity of the network is too simple and it cannot learn a complex function or it can lead to the under-fitting problem. Even though, the larger number of hidden neurons in the network makes the network more powerful and achieves a better capability to learn a complex function, we cannot keep increasing the number of hidden neurons. Too many hidden neurons may lead to the over-fitting problem, as well as unnecessarily enlarging the complexity of the network which consumes more time and computational resources in learning. The activation function for the hidden neurons in our implementation is sigmoid function.

- Output layer

  The output layer consists of output neurons in the same number of classes. Each of the output neuron indicates a specific class in the problem. If the output value of the output neuron indicating class x is greater than a specified threshold, we can determine that the network classifies the given input as class x. Nevertheless, it is possible that more than one class is obtained at a time. For this reason, another criterion is necessary to control the classifier to provide only one class at a time. We consider the output neuron which provides the maximum output value as the result of classification. Therefore, both threshold and maximum output value will be considered together to specify the output class. In the case that no output value is greater than the threshold, the classifier is designed to provide no class as the result.

4.1.2 Methodologies

The implementation of our neural network approach is based on Fast Artificial Neural Network Library (FANN) [1]. The FANN library is a free open source neuron network library which implements multi-layered artificial neural networks in C. The summary of the implementation using the FANN library will be described in this section.

[1]The FANN library can be found at `http://leenissen.dk/fann/`

We use fixed topology training in which topology of the network is determined in advance. The weights will be altered through a training process to minimize the difference between the desired output and the system output. There are three main components of our neural network classifiers system which are used in training: training data manipulation, implementation of the neural network, and training/testing the neural network.

The training data are manipulated before training. It is helpful to preprocess the data before training to ensure they fall into a specified range. The training data are scaled into the range of [0, 1] where the value of 0 indicates the minimum temperature value and value of 1 indicates the maximum temperature value in the data file. The training data is separated into folds in order to perform the cross validation and stored into files for later use. Each fold contains randomly selected data from all classes. The data in each class are evenly and randomly selected for each fold, so the folds will maintain the same distribution as that of original data. While using the library, the data is simply read by the fann_read_train_from_file function and stored in a structure called fann_train_data. To read the training file to train the network, the data must be in the following format.

num_train_data num_input num_output
input data separated by space
output data separated by space
.
.
.
input data separated by space.
outputdata seperated by space

The neural network described in Section 4.1.1 is created by using the standard creation function from the library which creates a standard fully connected backpropagation neural network. There is a bias neuron in each layer (except the output layer), and this bias neuron will be connected to all neurons in the next layer. When running the network, the bias nodes always emit 1.

The training process requires a training set which is comprised of examples including network inputs and target outputs. During training, the connection weights

and biases of the network are altered iteratively to reduce the network error function. The average mean square error (MSE) between the network output and the desired output is used to determine the error of the network. The connection weights and biases of the network must be initialized before training. The weights are random in the range of [-1, 1] in our approach. The weights are adjusted according to the error function until the minimal error is achieved. The batch or offline training is used to train the network, so the network is trained to reduce the error as a whole for every epoch. The error obtained by MSE of training does not indicate the performance of the classification. The performance in training indicate the difference between all system output bits to the target output while the performance of the classification indicates the classification accuracy, which is either correct or incorrect. Therefore, after the network is trained, the classification performance of the network will be evaluated.

After training is finished, the network is saved for later use. The trained network is evaluated on the testing data for assessing the performance of the classification. The maximum output bit will be considered as the result of the classification. However, the maximum value from the output neurons alone is not enough to determine the class since output values from output neurons can all be very small. Therefore, the threshold value is applied as a minimum value to allow the output value to be considered as a predicted class. The performance of the classifier is assessed by POD (Probability Of Detection), FAR (False Alarm Ratio), and CSI (Critical Success Index).

4.2 Incorporating under-sampling method to handle rare class

One approach to resolve the problem of rare-event classification is to re-sample the training data in such a way that the distribution of classes in the training data is balanced. A random under-sampling method randomly removes the instances of the majority classes from the training set, such that the sizes of majority classes are reduced and the class distribution in the training set is more balanced. This method

Under-sampling with	Freezing rain	Rain	Snow
0%	201	201	201
10%	201	221	221
20%	201	241	241
30%	201	261	261
40%	201	281	281
50%	201	302	302
60%	201	322	322
70%	201	342	342
80%	201	362	362
90%	201	382	382
100%	201	402	402

Table 4.1: Training instances for the under-sampling method

is applied in the data preparation process to manipulate the data before training the classifier. The neural network classifier is sensitive to the problem of imbalance class distribution, so we expect that the under-sampling method can ameliorate the effect of the unbalanced class and improve the performance of the classification of the freezing rain class.

We incorporate the under-sampling method to the precipitation type class classification by randomly eliminating the snow and rain class examples from the data set for variety sizes of eliminations. We investigate the performance effects of changing the size of snow and rain class data to be equal and larger than the freezing rain class size by 10% to 100%. The data set of each size is used to train the neuron network to predict the precipitation type. We construct the data sets at these required numbers of examples by a uniformly random method. The method selects the rain and snow data points from the original data set until the required size is achieved.

For example, there are 201, 571, and 404 training instances from freezing rain, rain, and snow classes respectively. With under-sampling at 10%, 20%, 30% greater than the number of freezing rain class, the examples from snow and rain classes are randomly selected for 221, 241, and 261 instances from the original instances. The rest of snow and rain data points are discarded. Table 4.1 shows numbers of examples of each class when training with under-sampling method.

Chapter 5

Discussion of Results

5.1 Data sets

Two data sets have been constructed for training and evaluating the proposed methods. One is the data set used for the data representation model which employs genetic algorithms to find significant vertical locations. The other is the data set used for training and evaluating the classifier in the precipitation type classification task.

5.1.1 Data set for data representation

The data set which is used to train genetic algorithms consists of available vertical levels and their corresponding temperature and dew point temperature values. These data are retrieved from Radiosonde database [18] produced by NOAA's National Climatic Data Center (NCDC). The data consists of the observations from St. John's, Newfoundland (YYT) station from 01/01/2000 to 31/12/2002. It was retrieved for all pressure levels at the time of 00:00 and 12:00 UTC. The sounding data obtained from this source are in the format shown in table 5.1.

The first 4 lines of the sounding are identification and information lines[1]. Missing or no reported data were recorded with 99999. The record time zone is in UTC. The type of identification line is described below.

254 = indicates a new sounding in the output file

1 = station identification line

[1]Technical information of the sounding data was obtained from [17]. The original source is referred to [39]

			Header line			
254	HOUR	DAY	MONTH	YEAR	(blank)	(blank)
1	WBAN	WMO	LAT D	LON D	ELEV	RTIME
2	HYDRO	MXWD	TROPL	LINES	TINDEX	SOURCE
3	(blank)	STAID	(blank)	(blank)	SONDE	WSUNITS
			Data lines			
	PRESSURE	HEIGHT	TEMP	DEWPT	WIND DIR	WIND SPD
9						
4						
5						
6						
7						
8						

Table 5.1: Original data format from Radiosonde database produced by NOAA's National Climatic Data Center (NCDC)

2 = sounding checks line

3 = station identifier and other indicators line

4 = mandatory level

5 = significant level

6 = wind level (ppbb) (gts or merged data)

7 = tropopause level (gts or merged data)

8 = maximum wind level (gts or merged data)

9 = surface level

The original data are re-formatted to form the training data set. All available pressure levels (the identification lines 9, 4, 5, 6, 7, and 8) and their corresponding temperatures and dew point temperatures are retrieved to construct the data set. In the original sounding data, the temperatures and dew point temperatures are in tenths of degrees Celsius and pressures are in tenths of millibar, so these variables are converted to degrees Celsius and millibars respectively. Each data point in the data set is composed of pressure levels paired with their temperature and dew point temperature values at a particular date and time. Due to the fact that the number

and location of pressure levels are different for each time the radiosonde has been launched, each data point contains different numbers and locations of pressure levels. For this reason, some data points which contain very few number of pressure levels will be eliminated from the data set.

In addition, each data point is encoded by using Sigma encoding. This method scales the pressure values into the range of [0,1], where the value of 1 represents the pressure at ground level and the value of 0 represents the highest pressure level that the balloon has reached. There are some missing and unreported values from the original data files, so these incomplete data will be removed from the data set. The data points whose lowest pressure values are greater than 500 mbar or highest pressure values are less than 950 mbar are considered as the outliers and are filtered out from the data set in order to maintain the consistency of the data.

5.1.2 Data set for precipitation type classification

In the precipitation type classification, the data set are created by using two data sources. One provides the vertical temperature and dew point temperature information retrieved from the sounding data, while the other provides the precipitation observations or class information of the corresponding sounding data. The first data source is obtained from the Radiosonde database [18], the same data source previously described in the section 5.1.1. The sounding data of the St.John's, Newfoundland, station have been retrieved for the period of 1994 - 2007. This station was selected because it has the most freezing rain hours in Canada.

In general, a neural network classifier requires a limited dimension of feature space as inputs, so the data set must be constructed with a particular number of features. Since the numbers and values of available pressure levels are different for each time radiosonde is launched, a set of pressure levels must be predefined in order to use as the baseline for retrieving the temperature values. The significant pressure levels that have been predefined for retrieving the temperature data sometimes do not exist in the sounding data, so the temperature values will be retrieved by the interpolation between two available pressure levels. As the result, each data

point is composed of the temperature values and dew point values in a predefined number.

The precipitation type observations or class information are retrieved from the climate database of Environment Canada's official online-presence for meteorological information and public forecasts [31]. Only observations of rain, snow, freezing rain, and freezing drizzle classes are considered, so other observations from this data source are discarded. Note that we also consider observations of freezing drizzle in our study because freezing drizzle and freezing rain observations are very similar. Thus we consider both of them as freezing rain class. The descriptions[2] of rain, snow, freezing rain, and freezing drizzle are described as follows:

- Freezing rain (FZRA, ZR): Freezing rain is liquid precipitation that reaches the surface in the form of drops that are greater than 0.5 millimeters in diameter. The drops then freeze on the earth's surface.
- Freezing Drizzle (FZDZ, ZL): Freezing Drizzle is liquid precipitation that reaches the surface in the form of drops that are less than 0.5 millimeters in diameter. The drops then freeze on the earth's surface.
- Snow (SN, SNW, S): Snow is an aggregate of ice crystals that form into flakes. Snow forms at temperatures below freezing. For snow to reach the earth's surface the entire temperature profile in the troposphere needs to be at or below freezing. It can be slightly above freezing in some layers if the layer is not warm or deep enough to melt the snow flakes.
- Rain (R, RA): Rain is liquid precipitation that reaches the surface in the form of drops that are greater than 0.5 millimeters in diameter. The intensity of rain is determined by the accumulation over a given time.

The records from both data sources are matched by using date and time information. Note that the class information from Environment Canada's climate database is recorded in local time, while the sounding data from Radiosonde database are

[2]The description were obtained from [19]

recorded in UTC. Thus, the time difference must be considered when matching the data from these two sources, both in the period of standard and daylight saving time. After matching the temperature profile with its corresponding class, the total data for training the classifier consists of 1176 data points. The statistics of the data are shown in table 5.2. The observations of rain, snow, freezing rain, and freezing drizzle are sometimes observed with other weather types. These mixtures of observations are also collected and represented by the dominant precipitation. Each precipitation type statistics including its combination with other observations are shown in the tables 5.3, 5.4, 5.5, and 5.6.

Precipitation type	# of instances
Freezing rain	57
Rain	571
Snow	404
Freezing Drizzle	144
Total	1176

Table 5.2: Data set statistics

Freezing rain type	# of instances
Freezing rain/	6
Freezing rain/Blowing Snow/	1
Freezing rain/Fog/	31
Freezing rain/Freezing Drizzle/Fog/	2
Freezing rain/Freezing Fog/	3
Freezing rain/Ice Pellets/	1
Freezing rain/Ice Pellets/Fog/	9
Freezing rain/Snow Showers/	1
Freezing rain/Snow/Fog/	1
Freezing rain/Snow/Ice Pellets/	1
Freezing rain/Snow/Ice Pellets/Blowing Snow/	1
Total	57

Table 5.3: Data set statistics of the freezing rain and its combinations with other observations

Rain type	# of instances
Rain	123
Rain/Drizzle/Fog	66
Rain/Fog	370
Rain/Ice Pellets/Fog	1
Rain/Snow	3
Rain/Snow/Fog	8
Total	571

Table 5.4: Data set statistics of the rain and its combinations with other observations

Snow type	# of instances
Snow	264
Snow/Blowing Snow	77
Snow/Fog	47
Snow/Freezing Fog	2
Snow/Ice Pellets	5
Snow/Ice Pellets/Blowing Snow	3
Snow/Ice Pellets/Fog	3
Snow/Snow Grains	2
Snow/Snow Grains/Fog	1
Total	404

Table 5.5: Data set statistics of the snow and its combinations with other observations

Freezing drizzle type	# of instances
Freezing Drizzle/	13
Freezing Drizzle/Fog/	75
Freezing Drizzle/Freezing Fog/	18
Freezing Drizzle/Ice Pellets/Fog/	1
Freezing Drizzle/Snow Grains/	5
Freezing Drizzle/Snow Grains/Fog/	3
Freezing Drizzle/Snow Grains/Freezing Fog/	1
Freezing Drizzle/Snow Showers/	2
Freezing Drizzle/Snow Showers/Fog/	2
Freezing Drizzle/Snow/	7
Freezing Drizzle/Snow/Blowing Snow/	2
Freezing Drizzle/Snow/Fog/	13
Freezing Drizzle/Snow/Ice Pellets/Fog/	1
Freezing Drizzle/Snow/Snow Grains/Fog/	1
Total	144

Table 5.6: Data set statistics of the freezing drizzle and its combinations with other observations

5.2 Application of GAs to find the optimal set of vertical levels

A simple genetic algorithm is used in our experiment. Each data point is encoded by using the Sigma (σ) coordinate system. The chromosomes are encoded with real numbers in range of [0,1] and represent the sets of possible solutions (vertical levels). The required number of pressure level is 31 levels in this experiment, so there are 31 genes in a chromosome, each of which represents a selected pressure level in Sigma encoding format. The GA was trained on 2119 data points of the St.John's, NF station, in the period of 2000 - 2002. The parameter setting for training the model is described below.

GA parameters

- Population size: 50
- Population initial range (0, 1)
- Stopping criteria

  Number of generations: 250 or

  Stall generation limitation: 10
- Selection method: Ranked roulette wheel
- Crossover method: Two point crossover with intermediate crossover in the middle section
- Mutation: Uniform mutation with m-rate = 0.05
- Number of individual copied to the next generation: 1

Figure 5.1 illustrates the best and average fitness scores at each generation during training. The fitness score is the average of root mean square errors (RMSE) between temperature profiles of the data points and the temperature profiles created at the selected pressure levels. The black and blue dots denote the best and mean

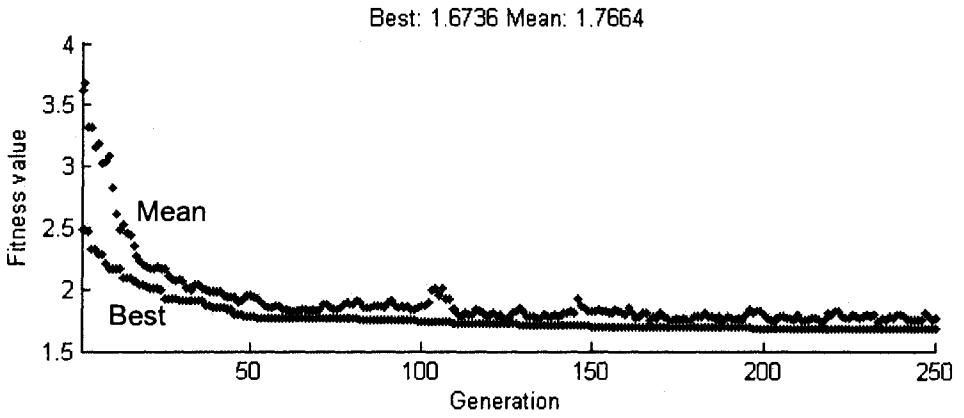


Figure 5.1: Best and mean individual fitness scores at each generation

fitness values respectively. The result indicates that the training starts to converge at the 50^{th} generation.

In this experiment, we compare the obtained pressure levels from our approach with two other sets of pressure levels: standard pressure levels from ECMWF (European Centre for Medium-Range Weather Forecasts) and synthetic pressure levels which are created by selecting the pressure at equal step in the range of 0 to 1. The equal step of the pressure level is the simplest method to create a set of pressure levels without any prior knowledge. We use the RMSE and MAE to evaluate the pressure levels selected by our approach. The average RMSE and MAE on testing data (2081 data points from the St.John's station in the period of 2003 - 2005) for the three models are shown in the table 5.7. The RMSE and MAE are calculated based on 100 discretized steps along the graph.

The results show that our approach (GAs) provided the best approximation of the data with a limited number of pressure levels compared to the data approximated by using standard levels from ECMWF and equal step pressure levels. Specifically, we obtained 13.81% relative RMSE reductions over the standard pressure levels from ECMWF.

The pressure levels in the Sigma encoding format obtained by using GAs as well as the standard pressure levels from ECMWF and equal step pressure levels

Pressure levels	Measurement	
	RMSE	MAE
GAs	1.704	0.836
ECMWF	1.977	0.882
Equal	2.249	0.934

Table 5.7: Average RMSE and MAE on testing data set

are illustrated in figure 5.2.

When using these pressure levels to obtain the temperature and dew point temperature values, the sounding of a data point can be drawn for such pressure levels.

A data point of freezing rain class is chosen as an example to illustrate the approximation of the data by using the three pressure level sets. The sounding of freezing rain data point (01/02/2000 12:00 UTC) is illustrated in Figure 5.3. For a better resolution, Figure 5.4 illustrates the sounding of the same data point at the pressure levels 0.6 to 1 where the temperature profile has undergone significant changes by moving from below-freezing layer to above-freezing layer and turning back to below-freezing layer that creates the positive area aloft and negative area at the surface. The pressure levels in this range are very important to predict the potential freezing rain.

5.3 Precipitation type classification using NNs

After obtaining the pressure levels from the experiment in section 5.2, we evaluate the quality of these pressure levels on precipitation type classification. The pressure levels from GAs, ECMWF, and equal step levels are used to approximate the data to train a neural network classifier described in Chapter 4. The data set is retrieved from the sounding data of the St.John's, Newfoundland station, from Jan 01, 1994 to Dec 31, 2007. The statistics of the data are provided in section 5.1.2.

The training set consists of 62 features: 31 features are from environment temperature values and 31 features are from dew point temperature values for selected levels. We combine freezing rain and freezing drizzle classes into one class since these two classes are very similar and very difficult to distinguish from each other,

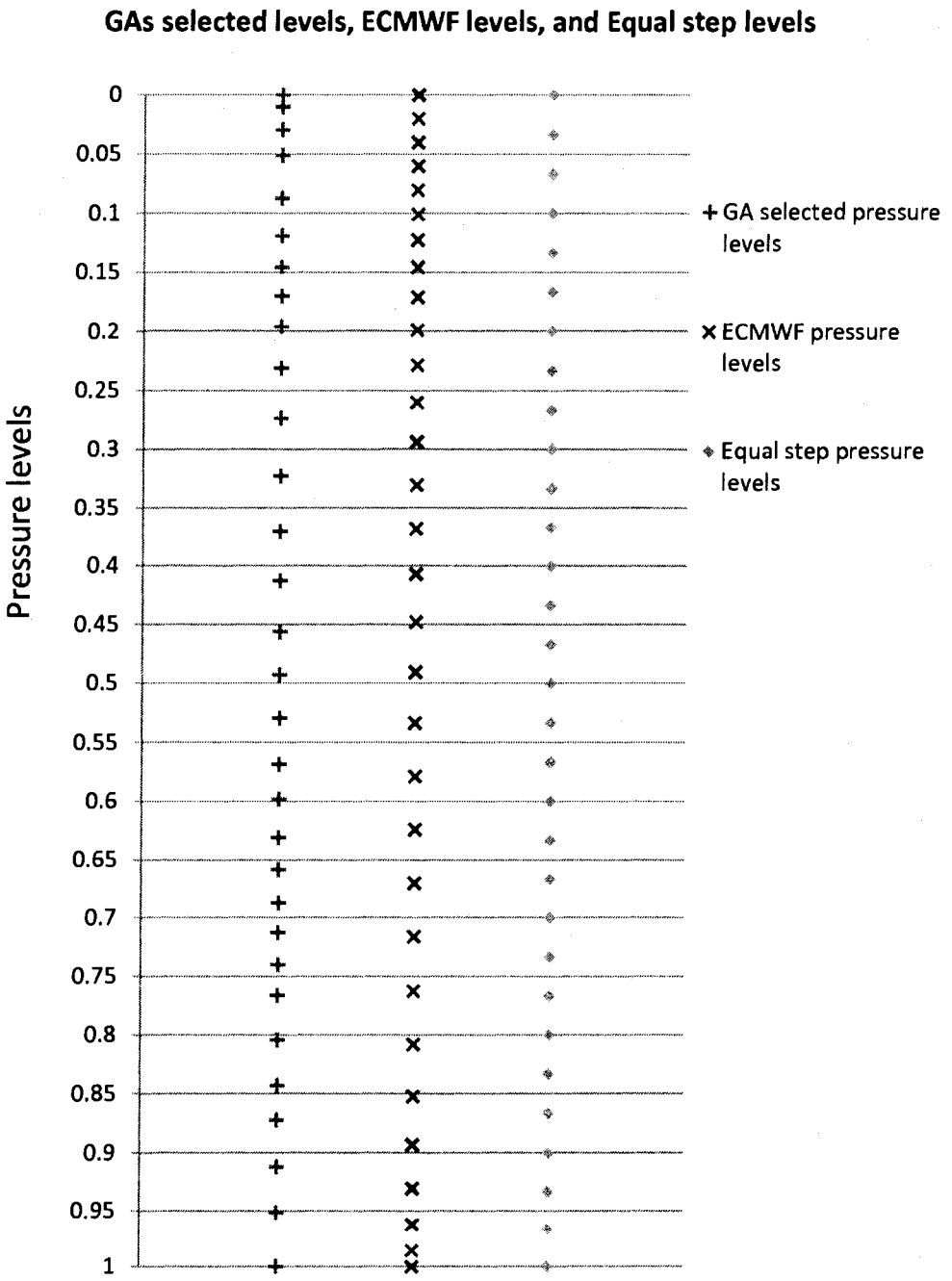


Figure 5.2: Vertical levels selected by GAs vs. standard levels of ECMWF

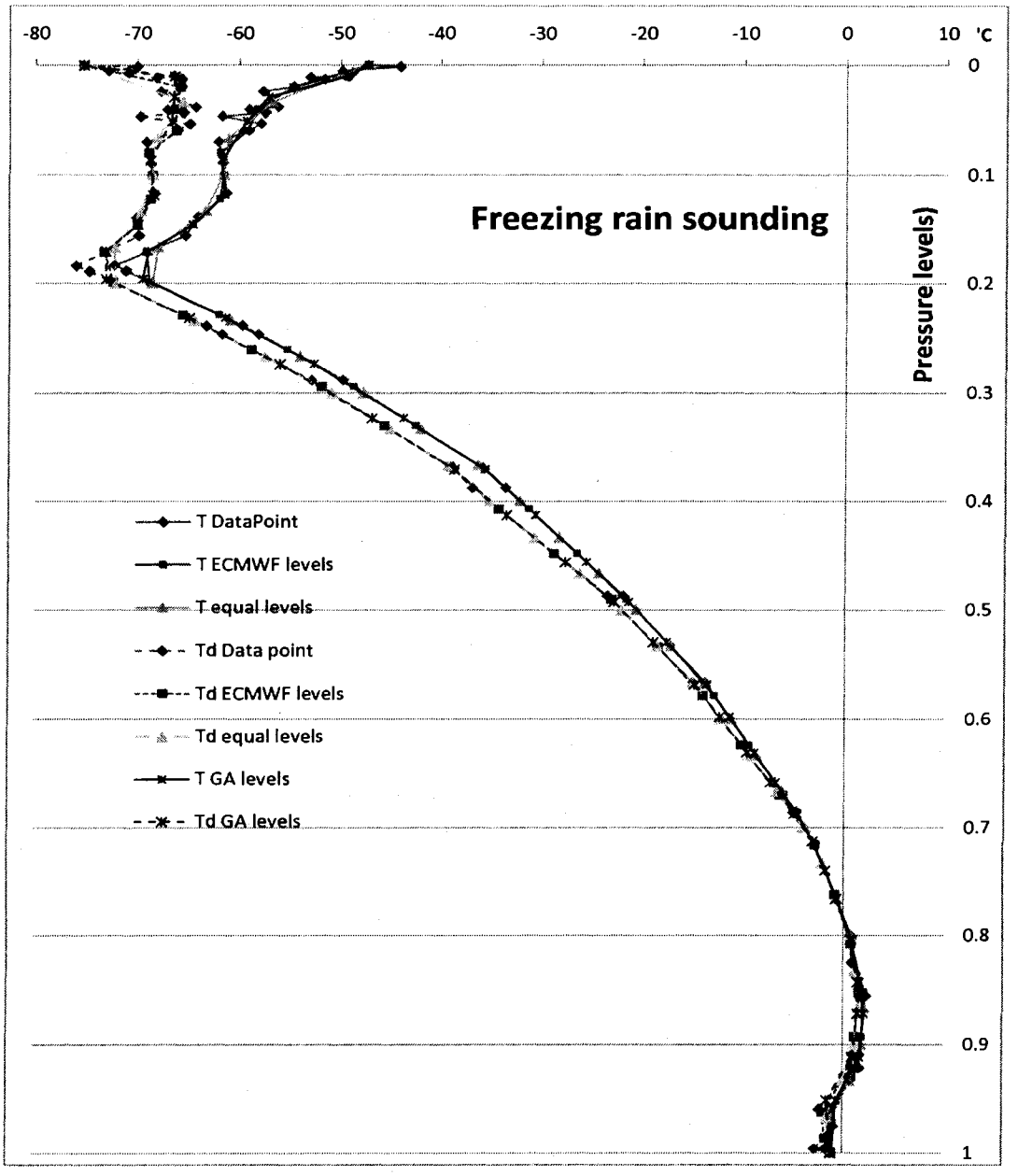


Figure 5.3: Sounding of freezing rain with the GAs selected vertical levels, ECMWF levels, and Equal step levels

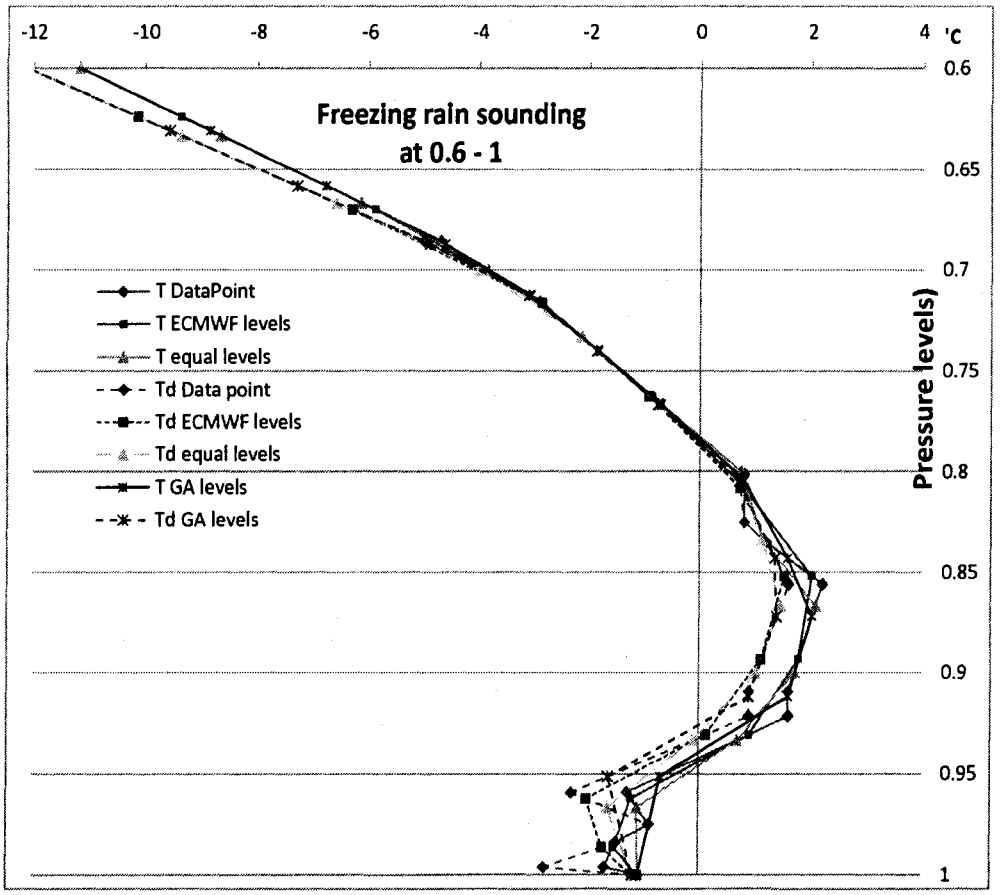


Figure 5.4: Sounding of freezing rain with the GAs selected vertical levels, ECMWF levels, and Equal step levels for range of 0.6 - 1 in Sigma coordinate system

even by human observations. The back propagation neural network was implemented as a classifier to predict precipitation type in this study. The criteria and parameters for the neural network were set as the following.

- Algorithm : Back-propagation
- Validation : 5 fold-cross validation
- System output is the output bit that provides maximum value
- Maximum epochs : 5000
- Number of layer : 3
- Number of input : 62 (equal to the number of features)
- Number of hidden neuron : 10
- Number of output neuron : 3 (Each neuron represents each class)
- Hidden neuron activation function : Sigmoid
- Output neuron activation function : Sigmoid
- Learning rate : 0.2
- Momentum rate : 0.05
- Error base on MSE
- Desired error : 0.01
- Training stopping function : Reach desired error or maximum number of epoch is reached

We evaluate the performance of the classifier by using CSI, HSS, POD, FAR, and BIAS measurements described in section 2.7. The results of the experiment are illustrated in Figures 5.5, 5.6, and 5.7, which show the performance on the freezing precipitation, rain, and snow classes respectively. The figures show the performance of precipitation type classification based on 5-fold cross-validation. We also maintain the class distribution of each fold with the same distribution as that of original data set, so that the data of each class are evenly distributed in each fold. Additionally, the same folds of the data are used for the three sets of pressure levels (GAs, ECMWF, and equal step levels).

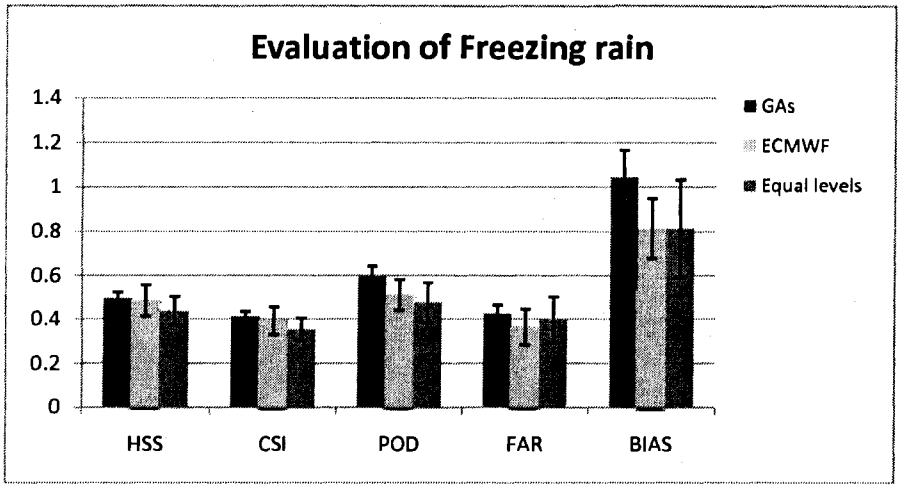


Figure 5.5: Performance on freezing rain classes

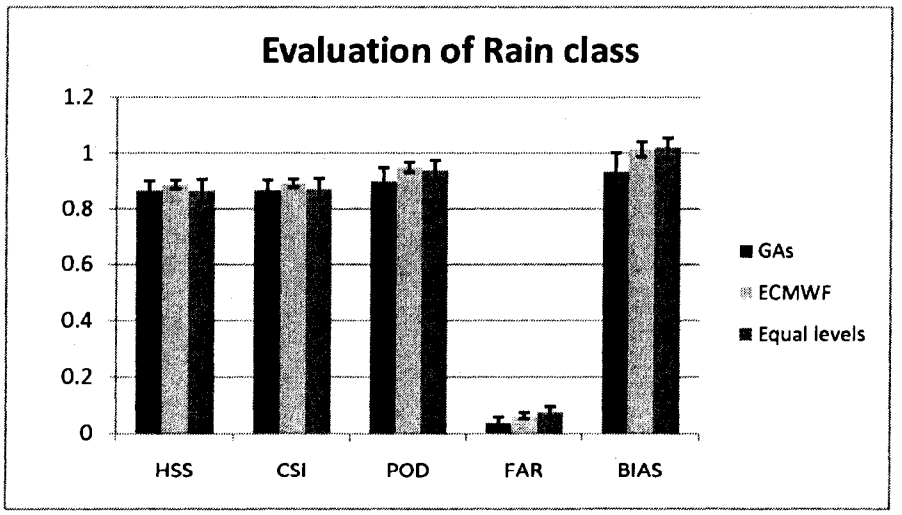


Figure 5.6: Performance on rain class

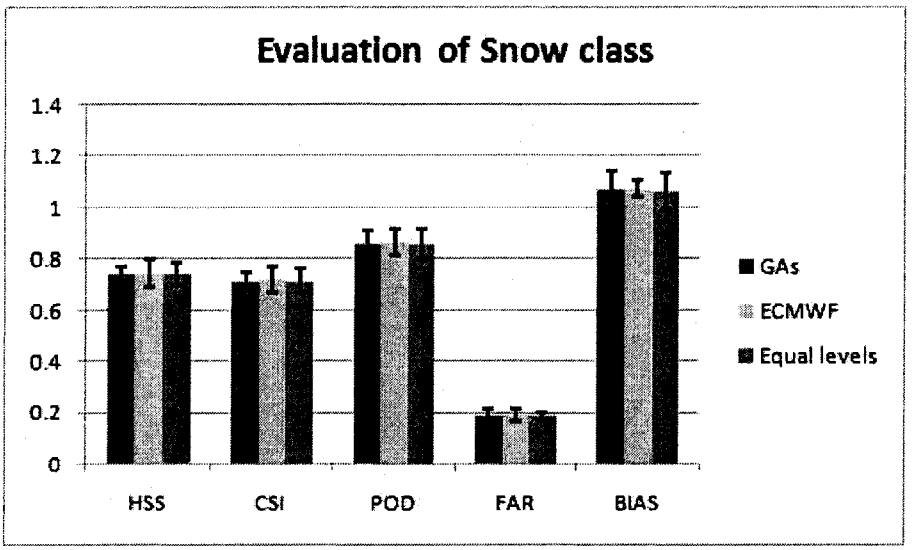


Figure 5.7: Performance on snow class

Ideally, a high performance can be determined by a large value of POD and small value of FAR. The results from the experiment show the equivalent performance for the three pressure level sets on the snow class. For the rain class, the ECMWF levels provide slightly higher values for both POD and FAR. We can use the CSI value to evaluate the performance in this situation since CSI is the function of both POD and FAR. A high value of CSI indicates that the prediction system is of high quality. The CSI value of GAs level is slightly smaller than that of ECMWF value but the difference is not significant. The performance of the system on the freezing rain class is very important in evaluating the system since it is a rare event. The results show that the GA levels provide a better performance on this class than both ECMWF and equal levels. This can be observed by the greater value of POD as well as the CSI value.

Further experiments have been conducted to find the best performance that each pressure level set can achieve regardless of the same architecture of the neural network for the three pressure level models. The constraint of the previous experiment is that the three pressure level sets have been used on the same architecture and parameters of the neural network; specifically, the number of hidden neuron has been

fixed for all three sets of pressure levels. Thus, the evaluation of each pressure level set on variable numbers of hidden neurons is worth mentioning in order to find best performance with the most appropriate number of hidden neurons for a particular pressure level model.

We vary the number of hidden neurons from 10 to 35. Figures 5.8, 5.9, and 5.10 show the best classification results from incorporating GAs, ECMWF, and equal pressure levels to classify freezing rain, rain, and snow classes respectively[3]. The best performance achieved by GA levels uses only 10 hidden neurons in the network while the best performance of using ECMWF levels and equal step vertical levels are achieved with 15 and 20 hidden neurons respectively. Use of GA vertical levels and ECMWF levels provide the comparable results in accuracy measured by CSI values, but GAs levels uses the least number of hidden neurons amongst other models. Thus, the set of levels found by GA allows construction of more parsimonious classifiers.

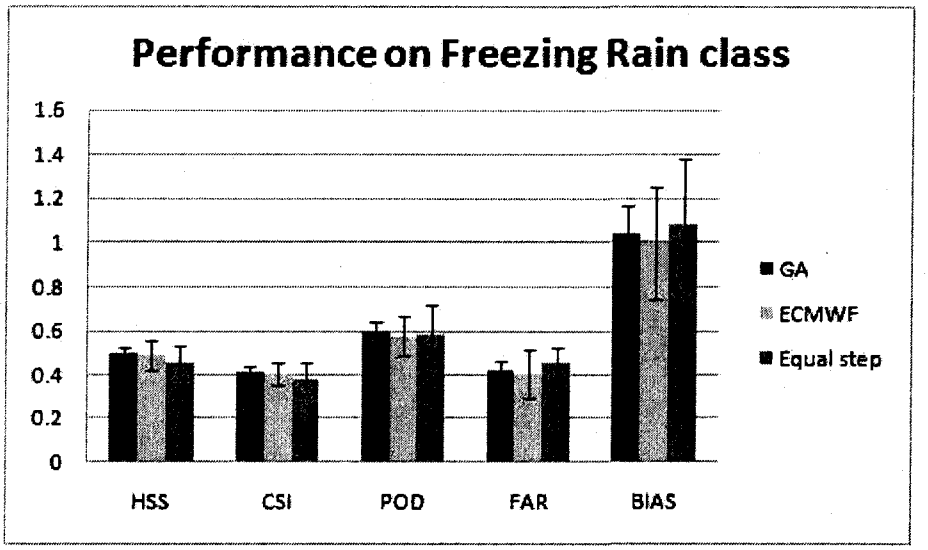


Figure 5.8: The best performance on freezing rain class achieved by each level model

[3]All performance results of using number of hidden neuron from 10 to 35 can be found in the Appendix A.1

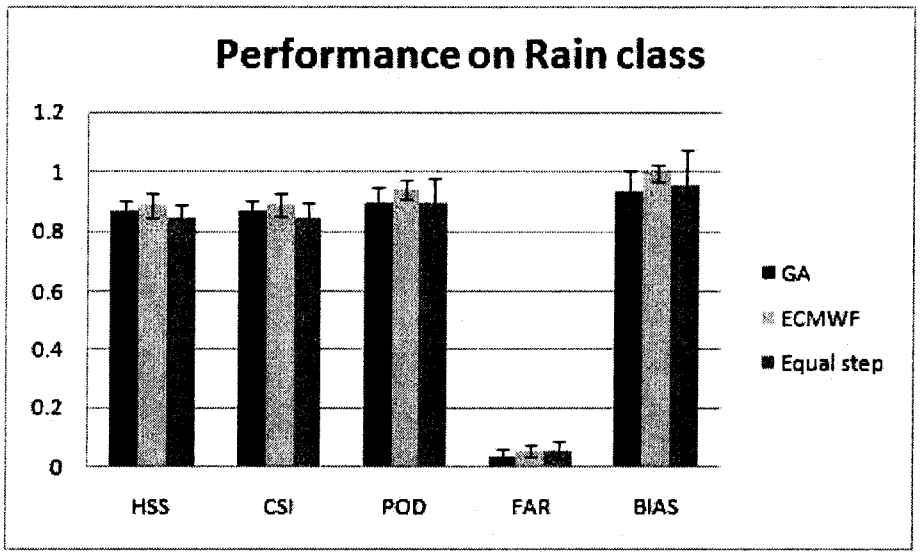


Figure 5.9: Performance on rain class

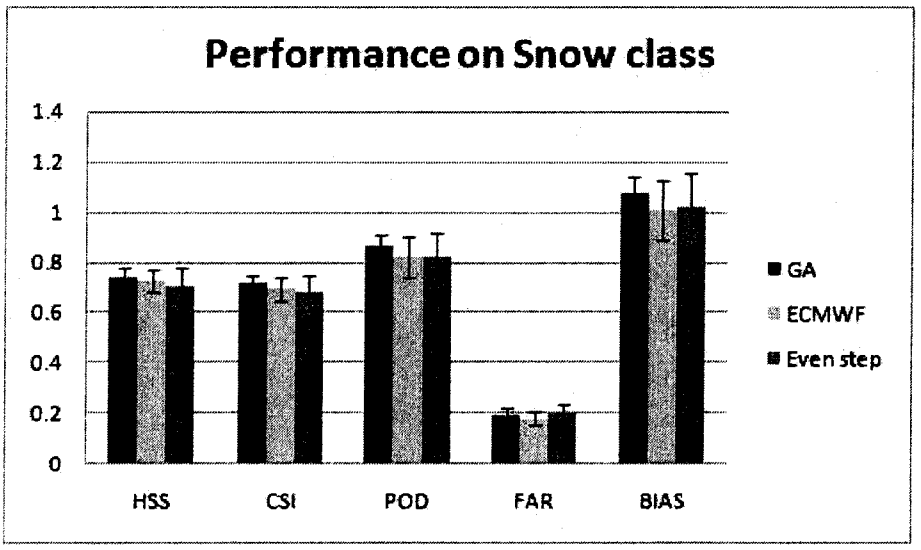


Figure 5.10: Performance on snow class

5.3.1 Using under-sampling to handle the freezing rain class

In this section, we incorporate the under-sampling method to adjust the class distribution to improve the classifier performance on the rare-event class. The under-sampling method eliminates the examples in majority classes to balance class distribution. One of our experiments involves the elimination of the examples in majority classes until the number of examples from both minority and majority classes are equal. We also vary the number of examples to be eliminated since the best class distribution for precipitation type classification is unknown. We can vary the class distributions by eliminating the majority class examples until they are 10%, 20%, 30%, ...,or 100% greater than the number of minority class examples. The performance results of classifier of using under-sampling data set are shown in Figures 5.11, 5.12, and 5.13.

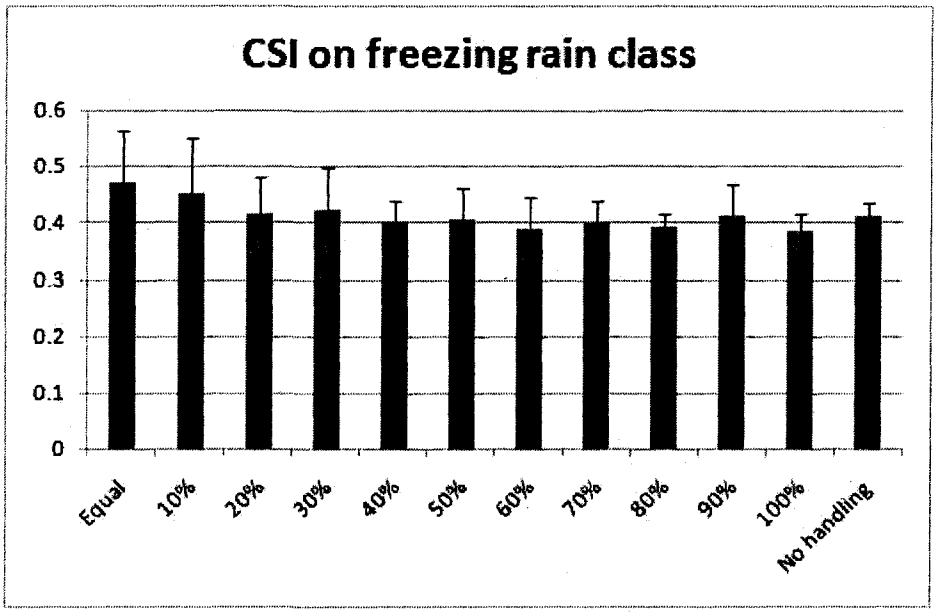


Figure 5.11: Performance on freezing rain class after applying under-sampling method.

Figure 5.11 shows that the performance of the classifier on the freezing rain class tends to decrease when the sizes of other classes are relatively greater than the size of the freezing rain class. Conversely, the performance of the classifier on the rain class and snow class tends to increase with less elimination of these

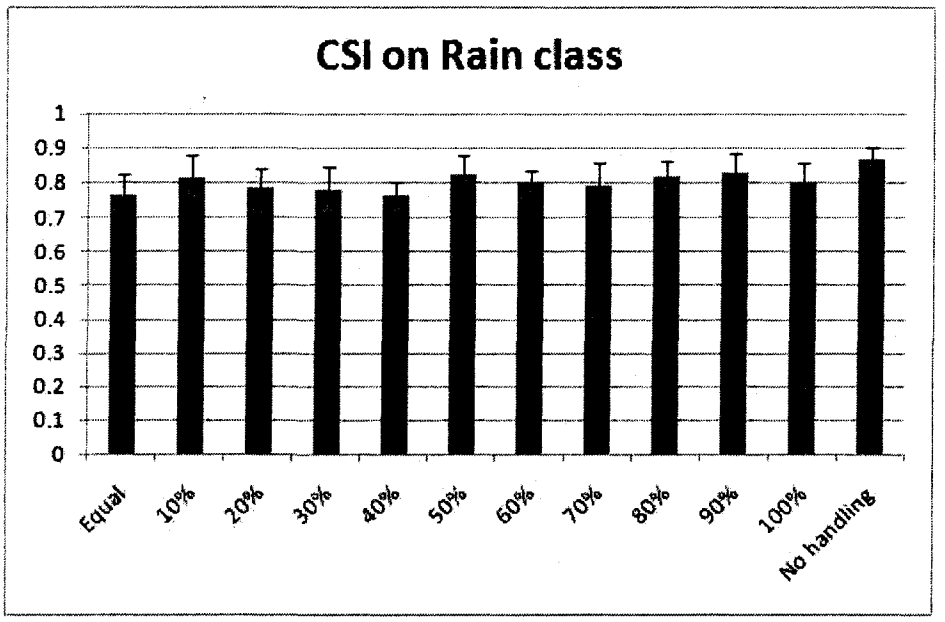


Figure 5.12: Performance on rain class after applying under-sampling method.

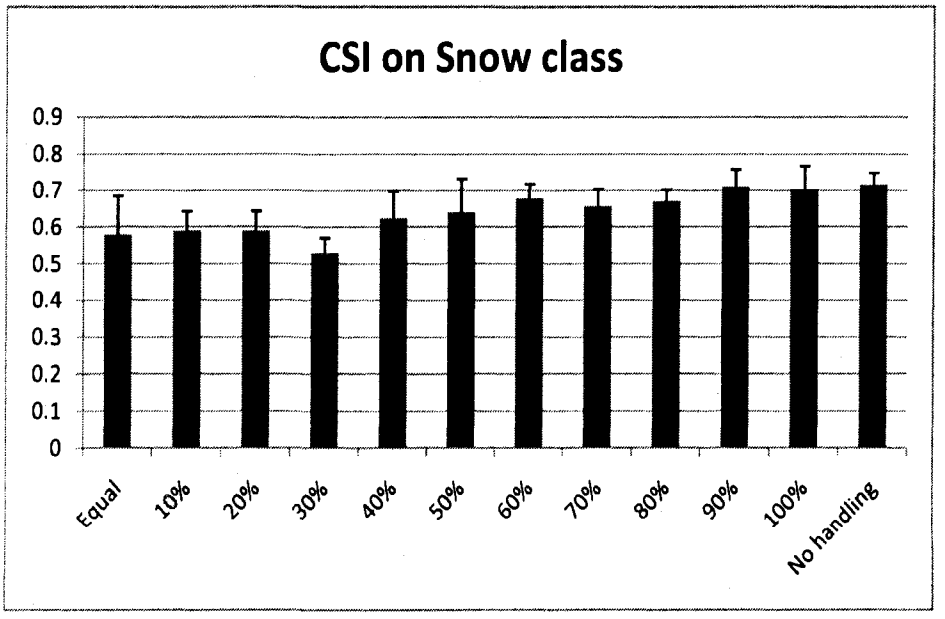


Figure 5.13: Performance on snow class after applying under-sampling method.

class data samples. It is straightforward that more training data generally provide more information of the classes to the classifier and lead to better performance results. The results show that when the freezing rain class data are in the same size as other classes' data, the performance of the classifier on the freezing rain class is improved by as much as 15% CSI. The improvement after applying the undersampling method suggests that the overwhelming instances of other classes degrade the performance of the classifier for the freezing rain class. Nevertheless, the improved performance on the freezing rain class is still in the low range compared to the performance on other classes. One possible reason to explain this phenomenon is that the small amount of freezing rain data in our training data set results in the lack of information for training the classifier to learn the pattern of the class.

5.3.2 A comparison with other precipitation-type algorithms

Precipitation type algorithms in previous studies[4] [1, 38, 3] involve the use of rules and sets of assertions to identify the precipitation types. Bourgouin [3] uses the negative and positive area between environment temperature and freezing layer incorporating predefined constants and rules to discriminate precipitation types. The precipitation type algorithm developed by Ramer [38] uses temperature, relative humidity, and wet-bulb temperature on different levels along with adjustable parameters to create rules. These parameters can be adjusted by varying the values over a wide range and evaluating the results. Baldwin's method [1] exploits atmospheric parameters such as coldest temperatures at many levels, temperature at the lowest layer, areas between 0 °C and wet-bulb temperature, etc. to create a set of rules. These algorithms are considered as rule-based algorithms which require experts in the domain to observe the data and encode their assertions into a rule set.

Instead, our approach is a data-driven method which can learn patterns of precipitation with no requirement of experts. The neural network classifier can learn and associate complex relationships in the data without prior knowledge of the domain, so it is flexible for use to predict precipitation type at any location. Further-

[4]The description of the algorithms are described in section 2.1

more, a back propagation neural network is easy to implement and adjust to predict more types of precipitation.

Due to different testing data for each algorithm, algorithms' quality can be inspected by indirect comparisons. The verification information of each algorithm is described below:

The proposed method was evaluated with 5-fold cross-validation on 1176 observations consisting of 201 of freezing rain[5], 571 of rain, and 404 of snow. Precipitation types occurring simultaneously with other precipitation were also included, but only the major occurring precipitation was selected. The best obtained CSI values for the experiment without using under-sampling method for rain, freezing rain, and snow are 0.91, 0.43, and 0.74 respectively and with under-sampling method are 0.83, 0.62, and 0.72 respectively.

Bourgouin's algorithm was evaluated on a small data set of 46 examples from 12 to 22 January 1995 retrieved from the Canadian operational Regional Finite Element model. It consists of 12 examples of freezing rain, 2 examples of ice pellets, 3 examples of mix of rain and snow, 13 examples of snow, and 16 examples of rain. There is no training process, so 46 examples were all used to verify the set of hand-crafted rules. Precipitation types were categorized into 5 classes. The system achieved CSI scores of 0.85, 0.67, 1.00, 0.86, and 0.94 for freezing rain, ice pellets, mixed rain and snow, snow, and rain class respectively.

Ramer's algorithm was evaluated on the data of rawinsonde and Surface Airways Observation (SAO) from late November 1992 to early March 1993. There are in total of 2,084 data points including 945 of rain, 58 of freezing rain, 30 of mix of rain and snow, 16 of freezing mix, and 1035 of snow. The achieved CSI values are 0.95 for rain, 0.45 for freezing rain, 0.23 for mix of rain and snow, 0 for freezing mix, and 0.93 for snow.

Precipitation-type algorithms have been evaluated in [32]. The study also evaluated these three algorithms on 1828 observations collected from Canadian and

[5]Note that the freezing rain class in the experiment consists of 57 observations of freezing rain and 144 observations of freezing drizzle

American rawinsonde. Only independent occurring precipitation types of snow, rain, freezing rain and ice pellets were selected. The data set consists of 627 observations of snow, 191 observations of ice pellets, 387 observations of freezing rain, and 623 observations of rain. Only the best achieved performance was reported. The best obtained CSI value for rain class is 0.8 which is achieved by Bourgouin's and Ramer's algorithms. The best CSI for freezing rain class is 0.6 achieved by Ramer's and Baldwin's algorithms and 0.8 for snow class achieved by CSTPS [7] method.

By inspecting the performance information of the proposed method and other algorithms, the quality of the proposed algorithm is comparable. Although there are lower scores on freezing rain class when the under-sampling method is not applied, the performance on rain and snow class is comparable to other methods. The lower scores on the freezing rain class can be explained as the freezing rain data used to train and verify the proposed method are not independent occurring events. The occurring of precipitation with other types of precipitation definitely negatively affects the classifier's capability in discriminating the precipitation type from each other. It is expected that the quality of the proposed method can be improved with independently occurring precipitation.

Chapter 6

Conclusions and future work

The precipitation type classification system is an important part of a system for automatic weather forecasting. We present the use of machine learning techniques in precipitation type classification as well as methods of preparing and representing atmospheric data in a pattern that can be learned by a classifier. We use vertical temperature profile obtained by radiosonde as the parameters in our classification system. The temperature profile from sounding is approximated and represented at a set of meaningful vertical levels. We apply the genetic algorithm method to search for these important levels. Finally, the temperature profiles represented at these levels are used to train the neural network classifier to predict the precipitation types. In our study, we are interested in the prediction of rain, snow, and freezing rain; especially, the prediction on freezing rain type that is of high significance in the study of rare event prediction.

We propose to use genetic algorithms to search for vertical levels where the temperatures values can well represent and approximate the actual temperature profile from soundings. The finite number of vertical levels makes it possible to automate the weather prediction process since the known locations of these levels can be used to acquire the significantly atmospheric parameters for classification task using numerical weather prediction. The vertical temperature profile is of prime importance to identify a precipitation type. Thus, we construct the vertical temperature profile at these levels to represent the actual temperature profile. The profile can be approximated by connecting every two adjacent points of temperature values at required

levels using a linear interpolation method.

The sounding data from St.John's, Newfoundland (YYT), station has been used to train GAs in finding the optimal set of vertical levels. The chromosomes are encoded with real numbers in the range of (0, 1) which represent the selected vertical levels in σ coordinate system. The experiment results demonstrate that the quality of vertical levels obtained from the proposed genetic algorithm method is better than that of standard levels from European Centre for Medium-Range Weather Forecasts (ECMWF). The quality of vertical levels can be directly assessed by determining the error between actual temperature profile and temperature profile constructed using the selected vertical levels. We use the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) to measure this quality.

Furthermore, the quality of the vertical level set is assessed through its application in the precipitation type classification. The temperature and dew point temperature values obtained at the vertical levels are used as the attributes for classification algorithms. We propose the neural network as a precipitation type classifier. A three-layer back-propagation neural network has been designed with all layers fully connected in a feed-forward manner. Inputs of the neural network are vectors of temperature and dew point temperature values at the selected vertical levels. The output layer consists of output neurons corresponding to the predicted classes. Each output neuron then indicates each type of precipitation. We investigate the performance of the neural network with three precipitation type including rain, snow, and freezing rain.

The classification performance is measured by probability of detection (POD), false alarm rate (FAR), and Critical Success Index (CSI). The experiment results show that the vertical levels obtained by GAs provide better performance on freezing rain class than the standard vertical levels from ECMWF. We are interested in the classification of the freezing rain class since it is a rare class, and its prediction is important for avoiding and mitigating damages it could cause.

To achieve a better classification performance, we incorporate the under-sampling method to balance class distribution. The experiment of under-sampling involves reduction of majority class examples until its number is equal to the number of minority class examples. We also vary the class distributions by eliminating the majority classes until the numbers of examples are 10%, 20%, 30%, ..., 100% greater than the number of minority class examples. The experiment results of using under-sampling method demonstrate the improvement of classification performance on freezing rain class.

6.1 Future work

We presented a novel approach to the classification of precipitation type. The goals of this work are to obtain the vertical levels which are important to categorize the precipitation type and to build an accuracy precipitation type classification system based on vertical temperature profile. While the experiments show promising results, there are some issues that can be improved in the future work.

GAs to find the optimal number of vertical levels

In our approach, we use GAs to find the optimal locations of vertical levels, while the number of required levels is specified in advance. It will be interesting to employ GA in searching for the optimal number of vertical levels as well. Genetic algorithms are suitable for multi-objective optimization; thus, with an appropriate chromosome encoding and sufficiently computational resources, it is promising that we can operate GA to find both the number of vertical levels and their locations at the same time.

Other atmospheric parameters

We have not used atmospheric parameters other than vertical temperature profile. Although the temperature profile is the most important, ideally other related atmospheric parameters should be considered in diagnosis of precipitation type.

These parameters include pressure, relative humidity, wind speed and direction, etc. An advantage of incorporating other parameters is that they provide more information on weather conditions to the system. It is expected that their use would enhance the performance of the precipitation type classifier.

Other rare event handling methods

Our focus in this thesis is on the representation of temperature profile and the precipitation type classification by using neural network. The simple re-sampling method, random under-sampling, is used to illustrate that neural network classifier performance is affected by the problem of imbalanced class. Applying methods of handling imbalanced class can improve the overall performance of the precipitation type classification. There are other interesting rare event handling methods that can be explored further in future work such as over-sampling by using SMOTE [6], multiple re-sampling method [12], MetaCost method [8], etc.

Bibliography

[1] M. Baldwin, R. Treadon, and S. Contorno. Precipitation type prediction using a decision tree approach with nmc's mesoscale model. *In Preprints, 10th Conference on Numerical Weather Prediction,Portland OR, American Meteorological Society*, pages 30–31, 1994.

[2] R. Barandela, J.S. Snchez, V. Garca, and E. Rangela. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3)(849 851), 2003.

[3] P. Bourgouin. A method to determine precipitation types, american meteorological society. *American Meteorological Society*, 15:583–592, 2000.

[4] S. Boyd and L. Vandenberghe. Convex optimization. *Cambridge University Press. Cambridge*, 2004.

[5] J.A. Bullinaria. Introduction to neural networks. lecture 1, 2, 3, 2004. accessed Apr 2008.

[6] N. Chawla, K. Bowyer, L. Hall, and P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, YEAR = 2002, volume = 16, pages = 341-378,*.

[7] R.R. Czys, R.W. Scott, K.C. Tang, R.W. Przybylinski, and M. E. Sabones. A physically based, nondimensional parameter for discriminating between locations of freezing rain and ice pellets. *Weather and Forecasting, vol. 11*, 11(4):591–598, 1996.

[8] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[9] C. Drummond and R.C. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. *Proceedings of the International Conference on Machine Learning (ICML 2003) Workshop on Learning from Imbalanced Data Sets II. Washington, DC, USA.*, July 2003.

[10] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* New York: Wiley, 2nd 2001.

[11] R.C. Eberhart and Y. Shi. *Computational Intelligence Concepts to Implementations.* Elsevier Inc., 2007.

[12] A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36, 2004.

[13] M.K. Fleury. 1998 north america big ice storm. http://naturaldisasters.suite101.com/article.cfm/the_great_ice_storm_of_1998, 2007. accessed May 2008.

[14] The European Virtual Organisation for Meteorological Training. Critical success index (csi) or threat score (ts), and equitable threat score (ets). http://euromet.meteo.fr/resources/ukmeteocal/verification/www/english/msg/ver_categ_forec/uos2/uos2_ko4.htm, accessed May 2008.

[15] Z. Fuqing. Nwp model notes - vertical resolution and coordinates. http://www.met.tamu.edu/class/metr452/models/2001/vertres.html, 2002. accessed June 2008.

[16] M.T. Goodrich. Efficient piecewise-linear function approximation using the uniform metric. *In proceeding of the symposium on computational geometry,ACM press*, pages 322–331, 1995.

[17] M. Govett and B. Moninger. Fsl output format description(fsl radiosonde database). http://raob.fsl.noaa.gov/intl/fsl_format-new.cgi.

[18] M. Govett and B. Moninger. Radiosonde database access. http://raob.fsl.noaa.gov/, August 2007.

[19] J. Haby. Precipitation types. www.theweatherprediction.com/preciptypes, accessed June 2008.

[20] E.J. Hopkins. Radiosondes – an upper air probe. http://www.aos.wisc.edu/~hopkins/wx-inst/wxi-raob.htm, 1996. accessed May 2008.

[21] N. Japkowicz. The class imbalance problem: Significance and strategies. *In Pro-ceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning Las Vegas, Nevada*, 2000.

[22] N. Japkowicz. Learning from imbalanced data sets: a comparison of various strategies, 2000.

[23] I.T. Jolliffe and D.B. Stephenson. *Forecast Verification: A Practitioner's Guide in Atmospheric Science*. John Wiley and Sons, 2003.

[24] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. *In proceeding of the IEEE International Conference on Data Mining*, pages 289–296, 2001.

[25] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30:25–36, 2006.

[26] A. Magnani and S.P. Boyd. Convex piecewise-linear fitting. *Optimization and Engineering, Springer Netherlands*, 2008.

[27] D.D. Massie. Neural network fundamentals for scientists and engineers. *CIBSE National Conference,Regents College*, 2001.

[28] W. McCulloch and W. Pitts. A logical calculus of ideas imminent in nervous activity. *Bull. Math. Biophys.*, pages 115–133, 1943.

[29] NASA. Weather forecasting through the ages, goddard space flight center greenbelt, maryland, 20771. `http://www.gsfc.nasa.gov`, 2002. accessed May 2008.

[30] Department of Atmospheric Science. University of wyoming. `http://weather.uwyo.edu/upperair/sounding.html`, accessed June 2008.

[31] Meteorological Service of Canada's national and regional offices. Climate data online. `http://www.climate.weatheroffice.ec.gc.ca/climateData/canada_e.html`, accessed May 2008.

[32] University of Oklahoma. Creation, evaluation, and implementation of precipitation-type forecasting system. `http://www.comet.ucar.edu/outreach/abstract_final/0019128.htm`, accessed May 2008.

[33] J.E. Passner, Army research LAB white sands NM computational, and information science directorate. An evaluation of three-dimentional weather hazards using sounding data and model output data. August 2000.

[34] M. Pazzani, C. Merz, P. Murphy, K. Ali, T. Hume, and C. Brunk. Reducing misclassification costs. *In Proceedings of the Eleventh International Conference on Machine Learning*, 1994.

[35] W. Pedrycz, A. Breuera, and N.J. Pizzi. Genetic design of feature spaces for pattern classifiers. *Artificial Intelligence in Medicine*, 32:115–125, 2004.

[36] J. Pittman and C.A. Murthy. Fitting optimal piecewise linear functions using genetic algorithms. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, pages 701–718, 2000.

[37] F. Provost. Machine learning from imbalanced data sets. *In AAAI Workshop on Learning from Imbalanced Data Sets. Tech Rep*, 2000.

[38] J. Ramer. An empirical technique for diagnosing precipitation type from model output. *Preprints, 5th Conf. On Aviation Weather Systems, Vienna, Virginia, AMS.*, pages 227–230, 1993.

[39] B.E. Schwartz and M. Govett. *A hydrostatically consistent North American Radiosonde Data Base at the forecast Systems Laboratory, 1946-present.* NOAA Technical Memorandum ERL FSL-4, Available from NOAA/ERL/FSL 325 Broadway, Boulder, CO 80303, 1992.

[40] P.J. Sousounis and T.A. Hutchinson. Wsi realtime winter precipitation forecasting using wrf. *21st Conference on Weather Analysis and Forecasting/17th Conference on Numerical Weather Prediction,Session 8A, Mesoscale Observations and Modeling of Winter Weather*, August 2005.

[41] S. Visa. and A. Ralescu. Issues in mining imbalanced data sets - a review paper. *in Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference*, 2005.

[42] G.M. Weiss. *Data Mining and Knowledge Discovery Handbook*, chapter Mining with Rare Cases, pages 765–776. Springer US, 2005.

[43] S. Yen and Y. Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. *Lecture notes in control and information sciences, ISSN 0170-8643,CODEN LCISDU*, 34:731–740, August 2006.

Appendix A

A.1 Experiment results using 3 vertical level models

Hidden NN 10		HSS	CSI	POD	FAR	BIAS
	FZRA	0.49±0.07	0.39±0.06	0.51±0.07	0.37±0.08	0.82±0.13
	Rain	0.89±0.02	0.89±0.02	0.95±0.02	0.06±0.01	1.01±0.03
	Snow	0.75±0.05	0.72±0.05	0.87±0.05	0.19±0.03	1.07±0.03
Hidden NN 15		HSS	CSI	POD	FAR	BIAS
	FZRA	0.49±0.07	0.41±0.05	0.58±0.09	0.40±0.11	1.00±0.25
	Rain	0.49±0.07	0.89±0.04	0.94±0.03	0.06±0.02	1.00±0.03
	Snow	0.72±0.05	0.69±0.05	0.82±0.08	0.18±0.03	1.01±0.12
Hidden NN 20		HSS	CSI	POD	FAR	BIAS
	FZRA	0.43±0.04	0.36±0.03	0.51±0.04	0.44±0.08	0.93±0.16
	Rain	0.87±0.01	0.88±0.01	0.95±0.02	0.07±0.02	1.02±0.04
	Snow	0.69±0.05	0.66±0.05	0.80±0.05	0.20±0.02	1.01±0.05
Hidden NN 25		HSS	CSI	POD	FAR	BIAS
	FZRA	0.47±0.10	0.39±0.09	0.58±0.13	0.44±0.10	1.05±0.27
	Rain	0.87±0.03	0.88±0.03	0.92±0.03	0.05±0.03	0.98±0.05
	Snow	0.67±0.08	0.65±0.08	0.79±0.11	0.22±0.03	1.01±0.13
Hidden NN 30		HSS	CSI	POD	FAR	BIAS
	FZRA	0.40±0.09	0.34±0.07	0.50±0.09	0.48±0.11	1.00±0.24
	Rain	0.88±0.02	0.88±0.02	0.92±0.03	0.05±0.02	0.98±0.04
	Snow	0.70±0.09	0.67±0.08	0.82±0.09	0.21±0.06	1.03±0.12
Hidden NN 35		HSS	CSI	POD	FAR	BIAS
	FZRA	0.38±0.10	0.31±0.09	0.46±0.18	0.48±0.06	0.88±0.36
	Rain	0.86±0.04	0.87±0.04	0.92±0.05	0.07±0.01	0.99±0.06
	Snow	0.71±0.04	0.68±0.04	0.84±0.05	0.21±0.04	1.07±0.11

Table A.1: Experiment results using ECMWF levels.

Hidden NN 10		HSS	CSI	POD	FAR	BIAS
	FZRA	0.44±0.07	0.36±0.05	0.48±0.09	0.40±0.10	0.81±0.22
	Rain	0.87±0.04	0.87±0.04	0.94±0.03	0.08±0.02	1.02±0.04
	Snow	0.74±0.04	0.72±0.04	0.86±0.06	0.19±0.01	1.06±0.07
Hidden NN 15		HSS	CSI	POD	FAR	BIAS
	FZRA	0.44±0.05	0.37±0.05	0.56±0.12	0.47±0.06	1.08±0.31
	Rain	0.83±0.05	0.84±0.05	0.88±0.06	0.06±0.04	0.94±0.09
	Snow	0.69±0.06	0.67±0.05	0.82±0.02	0.22±0.07	1.05±0.12
Hidden NN 20		HSS	CSI	POD	FAR	BIAS
	FZRA	0.46±0.08	0.38±0.07	0.58±0.14	0.45±0.07	1.08±0.29
	Rain	0.84±0.04	0.85±0.04	0.89±0.04	0.06±0.04	0.95±0.08
	Snow	0.71±0.07	0.68±0.07	0.82±0.10	0.20±0.03	1.02±0.13
Hidden NN 25		HSS	CSI	POD	FAR	BIAS
	FZRA	0.41±0.04	0.34±0.03	0.52±0.09	0.48±0.06	1.03±0.32
	Rain	0.82±0.05	0.84±0.04	0.91±0.04	0.09±0.04	1.01±0.06
	Snow	0.67±0.07	0.64±0.07	0.77±0.08	0.21±0.03	0.97±0.09
Hidden NN 30		HSS	CSI	POD	FAR	BIAS
	FZRA	0.32±0.06	0.28±0.06	0.45±0.18	0.55±0.09	1.04±0.51
	Rain	0.81±0.03	0.82±0.03	0.89±0.06	0.08±0.04	0.97±0.09
	Snow	0.67±0.07	0.65±0.07	0.79±0.09	0.22±0.05	1.02±0.15
Hidden NN 35		HSS	CSI	POD	FAR	BIAS
	FZRA	0.38±0.10	0.32±0.07	0.49±0.08	0.52±0.09	1.03±0.14
	Rain	0.83±0.03	0.84±0.03	0.91±0.04	0.08±0.03	0.99±0.07
	Snow	0.67±0.07	0.65±0.06	0.79±0.05	0.22±0.06	1.01±0.10

Table A.2: Experiment results using equal step levels.

Hidden NN 10		HSS	CSI	POD	FAR	BIAS
	FZRA	0.50±0.03	0.41±0.02	0.60±0.05	0.43±0.04	1.04±0.12
	Rain	0.87±0.03	0.87±0.04	0.90±0.05	0.04±0.02	0.93±0.07
	Snow	0.74±0.03	0.72±0.03	0.86±0.05	0.19±0.02	1.07±0.07
Hidden NN 15		HSS	CSI	POD	FAR	BIAS
	FZRA	0.46±0.04	0.38±0.04	0.53±0.07	0.43±0.05	0.94±0.15
	Rain	0.84±0.05	0.85±0.05	0.91±0.07	0.07±0.04	0.98±0.10
	Snow	0.72±0.04	0.69±0.03	0.84±0.03	0.20±0.04	1.06±0.09
Hidden NN 20		HSS	CSI	POD	FAR	BIAS
	FZRA	0.42±0.07	0.35±0.06	0.50±0.08	0.46±0.05	0.93±0.08
	Rain	0.87±0.02	0.88±0.01	0.94±0.02	0.07±0.02	1.01±0.05
	Snow	0.71±0.03	0.68±0.03	0.82±0.03	0.20±0.03	1.02±0.05
Hidden NN 25		HSS	CSI	POD	FAR	BIAS
	FZRA	0.40±0.07	0.34±0.05	0.54±0.14	0.50±0.09	1.12±0.42
	Rain	0.84±0.07	0.84±0.07	0.89±0.07	0.06±0.02	0.95±0.08
	Snow	0.66±0.10	0.64±0.10	0.78±0.16	0.22±0.03	1.01±0.22
Hidden NN 30		HSS	CSI	POD	FAR	BIAS
	FZRA	0.43±0.07	0.36±0.07	0.55±0.13	0.48±0.06	1.07±0.31
	Rain	0.86±0.04	0.86±0.04	0.92±0.03	0.07±0.04	1.00±0.06
	Snow	0.66±0.03	0.64±0.03	0.77±0.07	0.21±0.03	0.97±0.12
Hidden NN 35		HSS	CSI	POD	FAR	BIAS
	FZRA	0.39±0.05	0.33±0.04	0.53±0.06	0.53±0.05	1.13±0.14
	Rain	0.83±0.05	0.83±0.05	0.88±0.05	0.06±0.02	0.93±0.05
	Snow	0.66±0.06	0.64±0.06	0.79±0.08	0.23±0.02	1.03±0.10

Table A.3: Experiment results using GA levels

A.2 Experiment results using under-sampling method to handle rare class problem

	HSS	CSI	POD	FAR	BIAS
FR	0.47±0.11	0.47±0.09	0.61±0.13	0.31±0.11	0.91±0.24
Rain	0.80±0.06	0.76±0.06	0.85±0.09	0.10±0.10	0.96±0.22
Snow	0.58±0.13	0.58±0.11	0.78±0.08	0.31±0.10	1.14±0.10

Table A.4: Performance results on equally distributed class data.

	HSS	CSI	POD	FAR	BIAS
FR	0.46±0.12	0.45±0.10	0.61±0.16	0.35±0.07	0.94±0.25
Rain	0.84±0.06	0.81±0.06	0.91±0.06	0.10±0.10	1.02±0.16
Snow	0.60±0.07	0.59±0.05	0.75±0.08	0.26±0.09	1.03±0.22

Table A.5: Performance results on rain and snow data size is 10% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.43±0.06	0.41±0.07	0.57±0.16	0.38±0.06	0.94±0.32
Rain	0.81±0.05	0.79±0.05	0.88±0.05	0.12±0.06	1.01±0.10
Snow	0.60±0.05	0.59±0.05	0.76±0.11	0.26±0.05	1.04±0.20

Table A.6: Performance results on rain and snow data size is 20% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.42±0.09	0.42±0.08	0.65±0.14	0.45±0.04	1.18±0.24
Rain	0.81±0.06	0.78±0.07	0.86±0.07	0.10±0.07	0.96±0.13
Snow	0.53±0.08	0.53±0.04	0.65±0.06	0.25±0.11	0.90±0.24

Table A.7: Performance results on rain and snow data size is 30% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.42±0.07	0.40±0.04	0.58±0.07	0.41±0.10	1.02±0.27
Rain	0.79±0.04	0.76±0.04	0.84±0.06	0.11±0.06	0.95±0.11
Snow	0.63±0.07	0.62±0.07	0.78±0.12	0.24±0.02	1.03±0.17

Table A.8: Performance results on rain and snow data size is 40% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.41±0.18	0.39±0.13	0.57±0.17	0.47±0.13	1.07±0.19
Rain	0.83±0.05	0.81±0.05	0.87±0.05	0.08±0.06	0.95±0.10
Snow	0.62±0.09	0.62±0.08	0.77±0.12	0.23±0.08	1.01±0.19

Table A.9: Performance results on rain and snow data size is 50% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.43±0.06	0.39±0.06	0.53±0.14	0.38±0.09	0.89±0.37
Rain	0.82±0.03	0.80±0.03	0.87±0.06	0.09±0.06	0.96±0.12
Snow	0.68±0.03	0.68±0.04	0.85±0.12	0.21±0.09	1.10±0.24

Table A.10: Performance results on rain and snow data size is 60% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.44±0.04	0.40±0.04	0.60±0.12	0.44±0.05	1.08±0.29
Rain	0.82±0.06	0.79±0.07	0.84±0.09	0.06±0.03	0.90±0.12
Snow	0.66±0.03	0.66±0.05	0.81±0.13	0.22±0.05	1.05±0.23

Table A.11: Performance results on rain and snow data size is 70% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.45±0.02	0.39±0.02	0.55±0.07	0.41±0.08	0.95±0.22
Rain	0.84±0.04	0.82±0.04	0.87±0.05	0.07±0.03	0.93±0.07
Snow	0.67±0.03	0.67±0.03	0.84±0.06	0.23±0.03	1.09±0.11

Table A.12: Performance results on rain and snow data size is 80% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.47±0.07	0.41±0.06	0.57±0.09	0.40±0.05	0.96±0.18
Rain	0.85±0.05	0.83±0.06	0.87±0.05	0.06±0.03	0.92±0.06
Snow	0.71±0.05	0.71±0.05	0.87±0.05	0.21±0.02	1.09±0.05

Table A.13: Performance results on rain and snow data size is 90% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.45±0.05	0.38±0.03	0.52±0.10	0.36±0.16	0.89±0.34
Rain	0.82±0.04	0.80±0.06	0.87±0.11	0.07±0.07	0.95±0.18
Snow	0.69±0.08	0.70±0.07	0.87±0.07	0.21±0.06	1.11±0.13

Table A.14: Performance results on rain and snow data size is 100% greater than freezing rain class.

	HSS	CSI	POD	FAR	BIAS
FR	0.50±0.03	0.41±0.02	0.60±0.05	0.43±0.04	1.04±0.12
Rain	0.87±0.03	0.87±0.04	0.90±0.05	0.04±0.02	0.93±0.07
Snow	0.74±0.03	0.72±0.03	0.86±0.05	0.19±0.02	1.07±0.07

Table A.15: Performance results using the original data set.