An Intelligent Fault Diagnosis Framework for the Smart Grid Using Neuro-Fuzzy Reinforcement Learning

by

Babak Esgandarnejad

B.Sc. (University of Tehran, 2012)

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Applied Science

in the Department of Electrical and Computer Engineering

An Intelligent Fault Diagnosis Framework for the Smart Grid Using Neuro-Fuzzy Reinforcement Learning

by

Babak Esgandarnejad

B.Sc. (University of Tehran, 2012)

Supervisory committee:

Dr. T. Aaron Gulliver, Department of Electrical and Computer Engineering, University of Victoria (Supervisor)

Dr. T. Ilamparithi, Department of Electrical and Computer Engineering, University of Victoria (Departmental member)

Supervisory committee:

Dr. T. Aaron Gulliver, Department of Electrical and Computer Engineering, University of Victoria (Supervisor)

Dr. T. Ilamparithi, Department of Electrical and Computer Engineering, University of Victoria (Departmental member)

Abstract

Accurate and timely diagnosis of faults is essential for the reliability and security of power grid operation and maintenance. The emergence of big data has enabled the incorporation of a vast amount of information in order to create custom fault datasets and improve the diagnostic capabilities of existing frameworks. Intelligent systems have been successful in incorporating big data to improve diagnostic performance using computational intelligence and machine learning based on fault datasets. Among these systems are fuzzy inference systems with the ability to tackle the ambiguities and uncertainties of a variety of input data such as climate data. This makes these systems a good choice for extracting knowledge from energy big data. In this thesis, qualitative climate information is used to construct a fault dataset. A fuzzy inference system is designed whose parameters are optimized using a single layer artificial neural network. This fault diagnosis framework maps the relationship between fault variables in the fault dataset and fault types in real-time to improve the accuracy and cost efficiency of the framework.

Contents

Chapter 1: Introduction

Power networks all over the world are facing a paradigm shift towards higher efficiency and quality [1]. There is a global effort to enhance the intelligence of these networks to create the Smart Grid (SG). The SG will ensure a higher level of reliability in power generation, transmission and distribution compared to current networks. The quality of service, reliability of the grid, and safety of personnel and equipment have to be ensured at a competitive cost. Efficiency is a key issue in the SG as a more efficient power grid can reduce pollution and thus the carbon footprint [2]. SG must fulfill the demand for efficiency in energy management by incorporating advanced information and communication technologies with energy big data [1] [3]. This data is sensitive since it could be used for malicious purposes such as security breaches and cyber attacks. Therefore, security is an important SG issue. SG includes the concept of an interconnected network of semi-autonomous operating units called micro-grids. Each micro-grid is autonomous in terms of the generation, transmission and distribution of electricity in a geographical area and relies on active communication with other units to optimize operations. From a topological point of view, micro-grids in the SG function similar to stations in current networks. However, the autonomy of micro-grids helps the SG operate more efficiently and robustly while reducing the risks of propagating faults [3] [4] [5] [6] [7].

Tackling SG reliability, efficiency and security issues requires improvement of the sensory and supervisory abilities of the grid. This can be achieved by implementing enhanced maintenance procedures including the implementation of improved analytics and diagnosis frameworks to increase the awareness of the grid. The grid needs to be aware of grid operations (or the grid should be visible to the supervisory unit) to adapt to unforeseen changes. For example, major blackouts

have been caused by slow response to initiate necessary actions because of poor awareness of the grid and lack of appropriate analytics [1] [3] [6] [7]. In order to increase the awareness of the grid, the response time of the analytics and diagnostic framework needs to be decreased. This can be achieved by enhancing their accuracy and computational costs. It is the goal of this thesis to design a fault diagnostic framework that can improve the accuracy and computational costs for each micro-grid and consequently, the entire SG.

1.1 Fault Diagnosis

At the core of any grid anomaly is a fault which is a deviation from the acceptable, usual or standard condition. Reducing faults and the effects of faults is an important element of SG reliability, efficiency and security [4]. Therefore, correct and timely analysis of the causes of faults through accurate and timely diagnostics is needed to initiate effective preventive measures and reliable post-fault actions [3]. Faults can be caused by equipment failure, environmental conditions, security breaches or a combination of the above. One categorization of power grid faults focuses on the cause of faults. This divides faults into transient faults caused by natural events (e.g. lightning), persistent faults caused by device failures or outside attacks (e.g. cyber and physical), and anomalies which are usually shorter in duration and disappear once power at the fault location is cut off (e.g. overloading and short circuits) [3] [4].

Fault diagnosis in the smart grid involves many uncertainties and ambiguities. Since, typically, only part of a fault situation is known, it can be difficult to deduce the cause. Therefore, diagnosis becomes an inductive process in which the possibility of equipment malfunction needs to be considered together with other uncertainties and ambiguities. This transforms the inductive problem into a problem of inferring the cause of faults using an information set that may only be

partly related to the fault type. Traditionally, protection and maintenance engineers use Supervisory Control And Data Acquisition (SCADA) systems, digital fault recorders and other data monitoring, gathering and analysis methods for fault diagnostics. Fault diagnosis systems built using these methods mainly operate based on limit checking of process variables to help identify faults. For example, if the voltage between two points surpasses a pre-defined limit, an alarm is set off followed by a corresponding set of actions. The disadvantages of these methods include low flexibility and poor adaptability to unforeseen changes in the grid and the operating environment. They rely on a pre-defined model or expert knowledge in order to make decisions and must be redesigned in case of a change in the grid which makes them inefficient for automation. Another disadvantage is that increasing the accuracy of these systems typically requires more complex models or a larger database of expert knowledge which means higher accuracy with these methods comes at a higher computational cost [8]. Moreover, some of the information in the information set may be qualitative (e.g. climate conditions), whose inherent ambiguities make it difficult to create a database suitable for automated diagnostic frameworks.

1.2 Intelligent Fault Diagnosis

Advances in computational intelligence have enabled the development of advanced fault diagnosis methods with superior accuracy and low computational cost. Among these, methods based on learning with data-driven approaches have shown the greatest potential to satisfy key SG issues. These methods can learn from data which makes them adaptable to changes in the system and suitable for automation. In addition, enhancing the testing accuracy of these methods does not necessarily depend on a more complex model or larger database [8]. The use of historical data for diagnostics has been a trend for decades. Protection engineers are accustomed to extracting

knowledge of the system in the form of parameter estimates, rules and patterns for post-fault data analysis [9]. This knowledge can be used to identify and diagnose faults in the network.

Frameworks that use historical data for tackling power network diagnostic problems include knowledge-based approaches, data-driven approaches, optimization techniques and hybrid systems. Knowledge-based approaches, also known as expert systems, transform knowledge of the problem into a mathematical model using a set of rules with an underlying logic. They employ rule sets that represent the knowledge of the system and an inference engine that acts as the logical and decision making unit. A specific logic is necessary to construct the knowledge base and develop the inference engine. In this regard, Boolean logic has been used in a variety of studies [10 - 22]. However, these methods are susceptible to inaccuracies and the ambiguities of real-world phenomena. Data-driven approaches use data to extract knowledge about the system. They either model the system using signal processing or statistical approaches or operate without a model using machine learning techniques that employ mathematical and statistical concepts as well as those related to philosophy, psychology and neuroscience [23 - 26]. One example is an Artificial Neural Network (ANN) that can be viewed as a parallel distributed signal processor with simple processing units called neurons [27] [28]. Optimization techniques provide an estimate of the fault types through optimization of a grid model. These methods compare the results of a set of simulated fault incidents with actual measurements using parameters such as fault type, distance from a substation, resistance and energy [29 - 35]. Hybrid systems combine two or more of the above, e.g. fuzzy logic, neural networks and multi-agent systems [36] [37].

1.2.1 Fuzzy Logic and ANN Background

Fuzzy logic was introduced in 1965 as a means of handling uncertainties and ambiguities [38]. Unlike Boolean logic which assumes every fact is either true or false, fuzzy logic introduces a degree of membership to elements of a set that allows them to belong to more than one set simultaneously, which results in fuzzy sets. Moreover, fuzzy logic can use fuzzy sets to represent complex decision boundaries by including a set of fuzzy IF-THEN rules to describe the input-output relationship of the system [39 - 41]. For example, this approach was employed to find fault locations in transmission networks and to estimate the locations of faults with the help of a network of cause-effect relationships in [42] and [43], respectively. In this thesis, fuzzy logic is used to analyze qualitative information as well as to build a fuzzy inference system to obtain a fault data set and diagnose SG faults [37] [44] [45].

Neural networks are constructed with layers of highly interconnected processing elements called neurons. The ability to map non-linearities, adaptability, complex input-output relations as well as simple implementation are among the advantages of ANNs which make them very suitable for solving problems that involve learning from big datasets. Algorithms that automatically learn from data in order to map meaningful connections between input and output sets are generally referred to as Machine Learning (ML) algorithms [28] [46 - 50]. ML with ANNs has many advantages among which are the ability to learn using samples, classification, association and pattern recognition, working with insufficient and incomplete data (black box approach) and adaptability to a wide range of applications [45].

Classification problems that use training to map input/output relationships are called supervised learning while classification problems with no prior labels in the corresponding data are called unsupervised learning. Machine learning with ANNs can be used in both of these classification

problems to predict if a sample belongs to one of several classes (or clusters). ANN algorithms for automatic diagnosis of faults on feeders of distribution networks have been shown to be successful. These diagnostic methods benefit from a distributed architecture making them a good choice for SG fault diagnosis. They also benefit from lower computational costs compared to traditional methods which makes them suitable for real-time applications [51 - 53]. For example, a simple ANN was designed in [54] which is based on a radial basis function (a real-valued function whose value depends only on the distance from a point, usually the origin), to classify faults using voltages and currents as inputs. A similar approach was presented in [55] that uses a multilayer perceptron ANN instead of a radial basis function. A discriminative classifier known as a Support Vector Machine (SVM) was used in [56] to classify faults based on regression analysis of a model constructed via supervised ML from data. K-nearest neighbours is another classification method used for identifying a particular type of fault (lightning), making it one of the few examples of unsupervised learning involving ANNs [57]. In this thesis, a single layer ANN is used to optimize the parameters of a Fuzzy Inference System (FIS). The optimized inference system is then used for real-time diagnosis of SG faults.

1.3 Research Question and Contributions

The cause of SG faults has a strong correlation with climate conditions. For example, high winds and precipitation from seasonal storms cause tree limbs to fall on electricity distribution lines resulting in service interruptions to large numbers of customers and major power outages [58] [59]. Lightning is the cause of more than half of the faults in overhead transmission lines [3]. It is estimated that in the U.S. alone, climate-related outages cost between 20 and 55 billion dollars annually, and outages related to climate are increasing [58]. Therefore, utilizing climate

information could improve the ability of SG fault analysis frameworks. Considering that the climate conditions are often expressed qualitatively, this thesis considers the following question.

> How can climate information be incorporated into an intelligent framework to increase the accuracy and lower the computational cost of real-time SG fault diagnosis?

First, a database is constructed that includes measurements of faults in a specific geographical location together with the corresponding fault reasons. The corresponding climate information is added to the database to form the experimental set. The initial database that does not contain climate information serves as the control set. Then, strategies are designed to extract knowledge from these databases to understand the relationships between fault measurements and fault reasons (or types). This helps investigate the effect that different strategies have on the success of including climate information in the diagnostic framework. These strategies must satisfy two objectives. First, the diagnostic framework should be automated, which means it should not require manual adjustment once initialized. The second objective is for the design to operate in real-time. These two conditions can be translated into low learning error and low computational cost, respectively [2]. Therefore, the diagnostic framework is evaluated in terms of accuracy and computational cost.

The rest of this thesis is organized as follows. In the next chapter, an overview of the diagnostic framework is presented followed by a detailed explanation of the database. Then, each component of the framework is introduced followed by an explanation of their mechanism and relation to the other components of the framework. Chapter 3 presents the simulation results obtained using different datasets and rule generation strategies. Results are then compared in terms of accuracy and computational cost to investigate the effect of these variables. Chapter 4 presents some conclusions and future research possibilities.

Chapter 2: Methodology

2.1. Diagnostic Framework Overview

The flowchart in Figure 1 shows the diagnostic framework used in this thesis. It has three main parts which are data preparation, learning and validation. Information regarding grid faults is stored in the fault database. This information, collected from various sources, is cleaned and cross-referenced. Then it is quantized and clustered for use by the learning and testing parts of the diagnostic framework. Then it is divided into training and test sets. The training set is fed into a learning section that optimizes the input-output relationship between the fault measurements and fault types. The test set is used to validate the learning results. When a new fault occurs, its information is prepared and fed into the diagnostic framework for the corresponding output (i.e. fault type) can be inferred. After the actual fault type is determined, the information related to the new fault together with the actual fault type is used to update the fault database which in turn, is used for learning; thus, the term, intelligent framework.

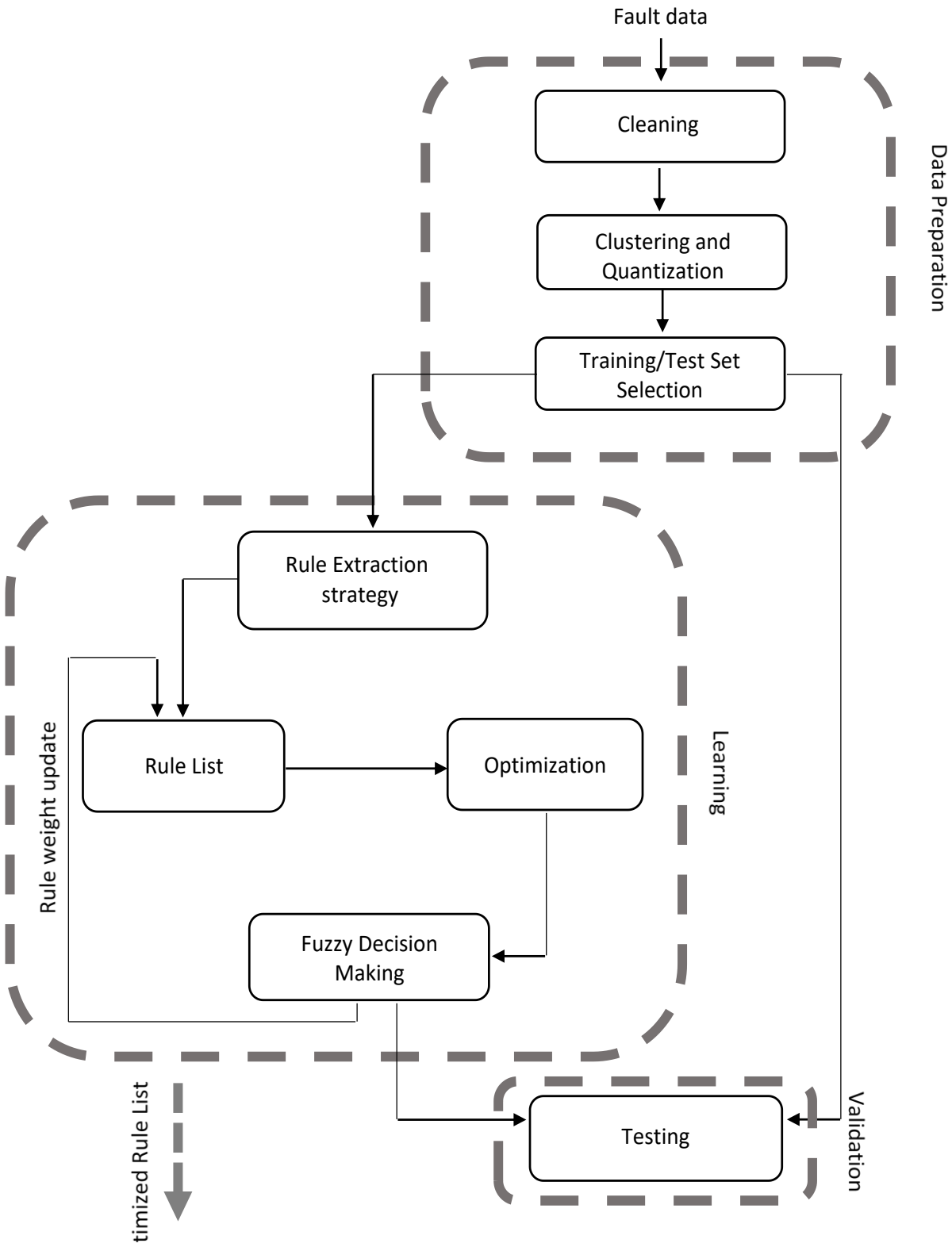

Figure 1. Flowchart of the diagnostic framework.

2.2 Data Preparation

2.2.1 Fault Database

In this thesis, a set of measurements associated with a fault is referred to as the input part, while the associated fault type is called the output part. Together they form a fault sequence. A series of historical fault sequences used by the diagnostic framework is referred to as the fault database. This includes data from the Greater Tehran Power Network located in the capital region of Iran, and data from the Tehran Province Meteorological Administration. The first source includes measurements related to transmission lines from two power stations gathered independently from one another, while the second provides information about climate conditions. Each fault sequence is specific to the time of a particular fault with one hour precision. These two sources are cross referenced based on their time stamps resulting in two data sets named after their corresponding stations (Shemiran and Azadi). The approximate geographic regions for these stations are shown in Figure 2. These stations have several substations and feeders and are responsible for the generation, transmission and distribution of electrical power in their respective areas. The data obtained from the Greater Tehran Power Network includes measurements of 1354 and 2822 fault incidents for the Shemiran and Azadi stations, respectively.

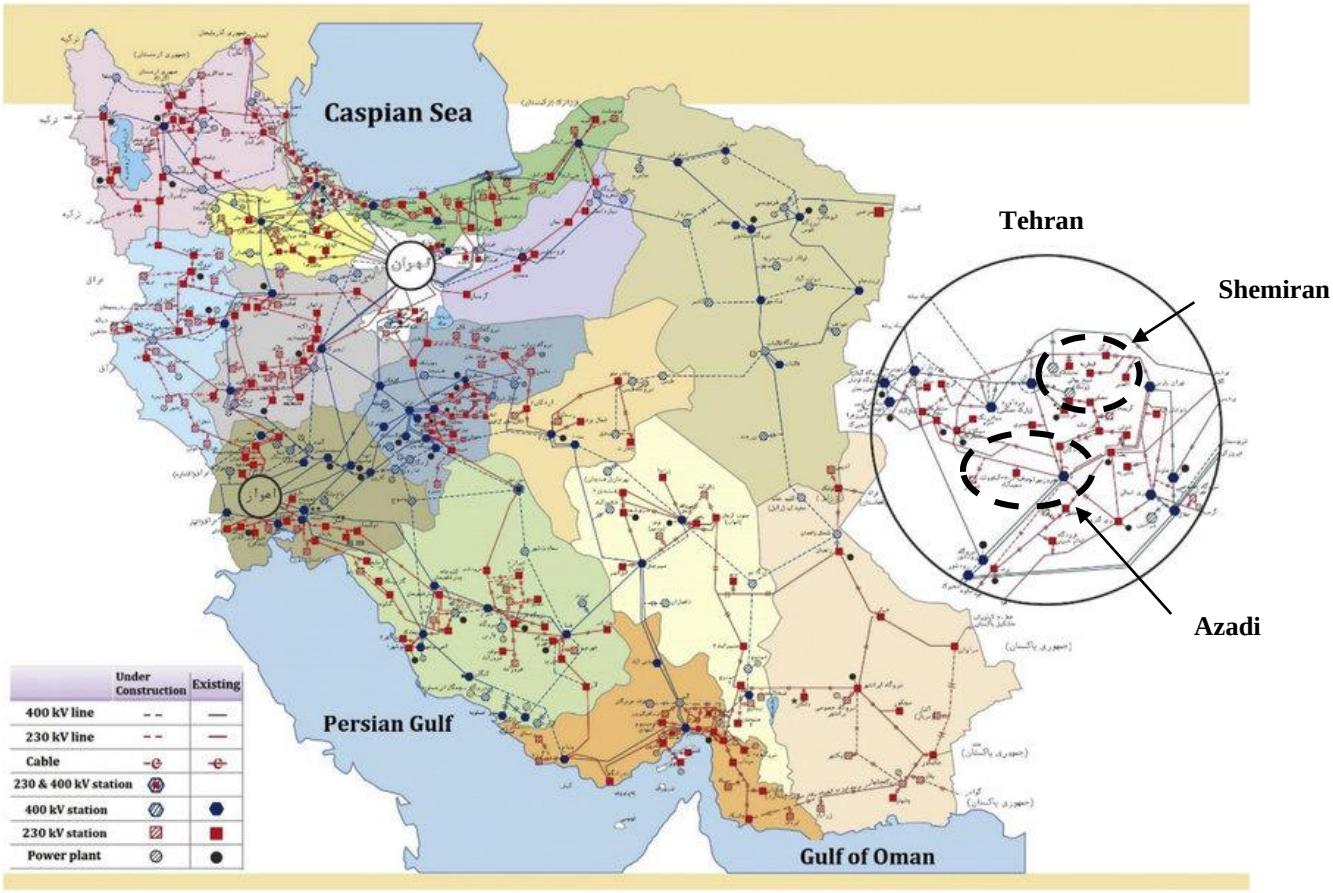

Figure 2. The approximate geographic regions of Azadi and Shemiran stations and their physical connections to the Iran power line transmission network.

The input part of a fault sequence is in the form of an array of length 14. It includes quantitative measurements of grid parameters expressed in numbers, quantitative parameters which are station specific evaluations of the fault environment provided by engineers expressed in numbers, and qualitative evaluations of the fault environment. They are referred to in this thesis as fault parameters or just parameters. It should be noted that some parameters in the fault database (e.g. preventive maintenance, load density and failure rate) were documented in their normalized form based on their station-specific values. The unavailability of the reference value for these parameters makes it impossible to combine station specific data sets to form one large fault

database. Therefore, fault diagnosis is done separately for each station which is advantageous to the design as discussed later in this chapter. A snapshot of the input part of a fault sequence used in this thesis is shown in Figure 3. Descriptions of the fault parameters are given below.

Station	Span	Sag	Tree Triming	PM	Aging	Load Density	Failure Rate	YY	M	D	H	Duration	Current	Energy	Climate Condition	Temperature	Dew point	Wind
Azadi	0	0	0	0.15	0.25	0.04	0.142857	86	1	4	0.465	18	10	0.088	Clear	14	-2	0
Azadi	0	0	0	0.85	1	0.1067	0.238095	86	1	26	0.308	2	35	0.034	Clear	13	1	0
Daneshgah	0	0	0	0.15	0.75	0.13867	0.047619	86	1	12	0.487	10	105	0.515	Dust	8	2	1
Daneshgah	0	0.1	0.1	0.4	0.75	0.2	0.238095	86	1	6	0.587	10	85	0.417	Rain	15	7	0
Daneshgah	0	0	0.2	0.15	0.5	0.168	0.238095	86	1	31	0.222	7	15	0.052	cloudy	19	4	1
Daneshgah	0	0.05	0	0.5	1	1	0.190476	86	1	24	0.604	8	50	0.196	Mostly cloudy	21	5	1
Shemiran	0.1	0.1	0.1	0.7	0.75	0.248889	0.238095	87	8	14	0.288	305	55	0.93	cloudy	8	6	0
Shemiran	0.1	0.1	0.3	0.6	0.75	0.153333	0.142857	88	10	22	0.493	70	130	3.754	cloudy	6	-4	0
Shemiran	0.1	0.1	0.6	0.5	1	0.12	0.904762	86	12	12	0.217	63	20	0.508	Rain	9	-1	0

Figure 3. An example of the fault sequences from the Greater Tehran Power Network.

- Span is the horizontal distance (l), measured in meters, between two electrical supports (poles) in transmission lines. This is a significant factor in determining the strength and size of an electrical support as it determines the maximum bending moment and deflection. Inappropriate span is a contributing factor in some faults [60].

- Sag (δ) is the vertical distance, measured in meters, between the highest point of an electrical support and the lowest point of the conductor between two adjacent electrical supports. This is an important factor in the operation of transmission lines [61]. Figure 4 shows the sag in a freely suspended conductor AOB. Sag for equal level supports is

$$\delta = \frac{wl^2}{8H} \qquad (1)$$

where w is the weight per unit length of the conductor, l is the span length, and H is the horizontal tension in the conductor at the point of maximum deflection (O).

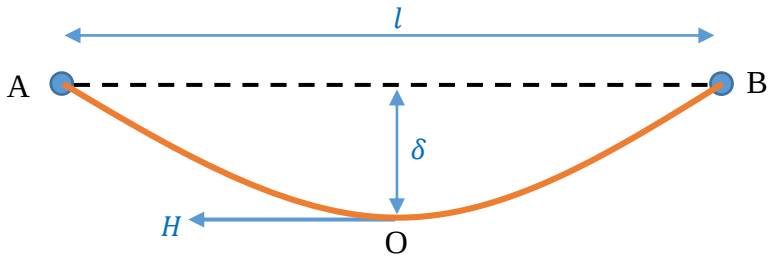

Figure 4. Span (l) and sag (δ) of a transmission line.

- Aging, which indicates how old the grid equipment is and reflects the quality of the equipment [61].

- Load density, which is the density of the electrical load at the point and time of the fault normalized for each feeder within each station. Grid equipment has a maximum current per volume (maximum load density) allowed and surpassing this value can damage equipment and lead to faults [61].

- Failure rate, which is the historical rate of faults of any type at the station normalized for each feeder within each station.

- Fault current, which is the electrical current (in amperes) at the point and time of the fault as recorded in the SCADA.

- Fault energy, which is the energy of the electrical load at the point and time of the fault as recorded in the SCADA.

- Fault duration, which is the duration of the fault as recorded in the SCADA.

- Preventive Maintenance (PM), which indicates the quality of grid equipment maintenance at the fault location. PM is an important factor in the reliability and security of the grid. The PM values are normalized for each feeder within each station [62].

- Tree trimming, which indicates the probability of tree branches touching the grid equipment. Insufficient tree trimming can disrupt grid operation [63].

- Temperature (in degrees Celsius) at the time and point of the fault which is an important factor affecting all grid components and parameters from equipment to current [59] [61].

- Dew point, which is the temperature (in degrees Celsius) to which air must be cooled, at constant pressure and moisture content, in order for saturation to occur. This is a measure of atmospheric moisture. The higher the dew point, the greater the amount of water vapor in the air, generally referred to as humidity. This is an important factor in the operation of grid components as it results in corrosion to metal components [59] [61].

- Climate condition is expressed in natural language and is determined by meteorologists at the time and location of the fault. These conditions from least to most severe [59] are

  Clear

  Mostly Clear

  Partly Cloudy

  Mostly Cloudy

  Cloudy

  Wind(y)

  Dust

  Wind and Dust

  Mist

  Fog

  Rain

  Snow

  Lightning

  Rain and Lightning

- Wind condition is a binary variable indicating the wind severity at the time and place of the fault and is determined by meteorologists. Wind condition 1 indicates that it is severe enough to threaten power network safety at the time and place of the fault while wind condition 0 indicates otherwise.

The fault type is determined qualitatively by engineers present at the fault location and is categorized in the preparation stage as transient, persistent or anomaly. It is a vector that indicates one of the following.

- Transient fault, which is caused by natural phenomena such as lightning.
- Persistent fault, which is caused by device failure or an attack.
- Anomaly, which is a short lived fault that usually disappears after the power is cut off and restored (such as birds interfering with transmission lines or short lived device malfunctions) [4].

Fault type vector $[0, 0,1]$ indicates a transient fault, $[0, 1, 0]$ a persistent fault, and $[1, 0, 0]$ an anomaly. Examples of fault reasons and the associated vectors are shown in Figure 5. This is a snapshot of the output part of some of the fault sequences that were obtained in Farsi (Persian) language from the sources introduced in Section 2.1.1. The faults (failure) reasons were then translated into English and matched with their corresponding types.

Failure Reason (English)	Anomaly	Persistent	Transient
Fault in the client's internal network	1	0	0
Transient Fault	0	0	1
Transient Fault	0	0	1
Transient Fault	0	0	1
Transient Fault	0	0	1
Transient Fault	0	0	1
Fault in the lightning arrestor – Exploded or burnt lightning arrester	1	0	0

Figure 5. An example output part of some fault sequences and associated reasons.

Since the stations whose data is used in this thesis are independent in terms of generation, transmission and distribution of electricity in the corresponding geographical regions, they are assumed to be micro-grids in an SG. After constructing station specific data sets, each variable is

normalized using feature scaling to restrict the values to the range $[0, 1]$. After the fault database is constructed, it must be prepared for the learning stage. To this end, the fault database is clustered and quantized, and then divided into training and test sets.

2.2.2 Clustering and Quantization

In order to simplify the identification of patterns in the input-output space (i.e. map relationships between the input and output parts of a fault sequence), the number of clusters for each parameter in the fault data sets is investigated. Clustering reduces the dimension of the input/output space which results in lower computational cost. Moreover, the number of clusters determines the number of elements in fuzzy linguistic sets related to each parameter. Identifying the number of clusters is done using the jump algorithm. This algorithm uses K-means clustering with a within-cluster dispersion metric called distortion to measure the distance between possible cluster centers. Consider each parameter in the fault database of one of the stations X as a P-dimensional random variable with covariance Γ for each cluster. If $c_1, c_2, \dots, c_k$ is a set of candidate cluster centers with c_x being the closest to X, then the minimum achievable distortion associated with K centers fit to the data is

$$d_K = \frac{1}{P} \min_{c_1,\dots,c_K} E[(X - c_X)^T \Gamma^{-1} (X - c_X)]$$

This can be interpreted as the average Mahalanobis distance between X and c_X per dimension. Since $P = 1$ for every parameter in the fault database used in this thesis, so the covariance matrix Γ is an identity matrix, so the above equation can be written as

$$d_K = \min_{c_1,\dots,c_K} E[(X - c_X)^T (X - c_X)] \qquad (2)$$

It was shown in [8] that plotting d_k^{-Y} versus K (i.e. distortion curve where Y is an appropriate negative number usually set to $\frac{P}{2}$), will show a sharp jump at the true number of clusters. This leads to the jump clustering algorithm used in this thesis which has the following steps.

- Execute the K-means algorithm on each variable in the input part of the fault sequences specific to a station for various numbers of clusters, K, and calculate the corresponding distortion $\hat{d}_K$ using (1).
- Select a transformation power $Y = \frac{P}{2} = 0.5$
- Calculate the jumps in the transformed distortion $J_K = \hat{d}_K^{-Y} - \hat{d}_{K-1}^{-Y}$ (3)
- The number of clusters is $K' = \arg\max_K J_K$ (4)

In some cases, the resulting number is very large, leading to fuzzy linguistic sets with a large number of elements which makes it difficult to design the corresponding membership functions [64]. To solve this problem, a maximum number of clusters is defined for each variable. The jump clustering algorithm results are given in Table 1. An asterisk (*) indicates a large number of clusters reduced to 20. This number was obtained empirically considering the jump results to simplify the design of the fuzzy membership functions which, in turn, reduces the computational cost.

Variable	Jump results	Reduced results
Span	9	9
Sag	10	10
Tree Trimming	10	10
PM	11	11
Aging	12	12
Load Density*	85	20
Failure Rate	16	16
Duration*	208	20
Current*	64	20
Energy*	579	20
Climate Condition	14	14
Temperature*	53	20
Dew Point*	44	20
Wind	2	2

Table 1. Jump and reduced jump results for each variable of a fault sequence.

2.2.3 Training/Test Set Selection

After the data has been clustered and quantized, it is divided into training and test sets. The training set is used to generate the FIS rule set whose weights are optimized by the ANN via reinforcement learning. The test set is used to validate the performance of the learning stage. To avoid bias, these sets are randomly chosen from the original fault database.

2.3 Decision Making Using FIS

The main components of a fuzzy inference system are

- fuzzification,

- fuzzy rules that operate as an inference engine reasoning of outputs based on fuzzy inputs, and
- de-fuzzification that translates outputs of the fuzzy inference system back into non-fuzzy variables.

These components are explained in the following.

2.3.1 Fuzzification

Fuzzification translates crisp variables (i.e. variables that take precise numerical values) into fuzzy inputs to be processed by the FIS. Fuzzification uses a fuzzy membership function for a crisp input variable to translate it to a fuzzy variable. A fuzzy membership function associated with a given fuzzy set maps an input value to an appropriate membership value. This is done by assigning a degree of membership by which the input value belongs to a fuzzy set. This is a number between zero and one. Input and output membership functions enable the fuzzification (and de-fuzzification) of the input (and output) space. The degree of membership to a fuzzy set is given by

$$\mu_A : X \rightarrow [0,1] \qquad (5)$$

where μ_A is the membership function of fuzzy set A and X is a fuzzy linguistic set. Table 2 shows the fuzzy linguistic sets associated with each of the fault sequence variables used in this thesis in which L stands for $Level$. Membership functions can be designed manually or automatically from data.

Variable	Number of Clusters	Linguistic Set
Span	9	$\{Appropriate, Inappropriate\ L1, Inappropriate\ L2, \dots, Inappropriate\ L8\}$
Sag	10	$\{Appropriate, Inappropriate\ L1, Inappropriate\ L2, \dots, Inappropriate\ L9\}$
Tree Trimming	10	$\{Sufficient, Insufficient\ L1, Insufficient\ L2, \dots, Insufficient\ L9\}$
PM	11	$\{Proper, Improper\ L1, Improper\ L2, \dots, Improper\ L10\}$
Aging	12	$\{Fresh, Aged\ L1, Aged\ L2, \dots, Aged\ L11\}$
Load Density	20	$\{Light\ L2, Light\ L1, Light, Average, Dense\ L1, Dense\ L2, \dots, Dense\ L17\}$
Failure Rate	16	$\{Low, Average, High\ L1, High\ L2, \dots, High\ L14\}$
Duration	20	$\{Short, Medium, Long\ L1, Long\ L2, \dots, Long\ L18\}$
Current	20	$\{Low, Medium, High\ L1, High\ L2, \dots, High\ L18\}$
Energy	20	$\{Low, Medium, High\ L1, High\ L2, \dots, High\ L18\}$
Climate Condition	14	$\{Clear, Mostly\ Clear, Partly\ Cloudy, Mostly\ Cloudy, Cloudy, Windy, Dust, Wind\ and\ Dust, Mist, Fog, Rain, Snow, Lightning, Rain\ and\ Lightning\}$
Temperature	20	$\{Low, Medium, High\ L1, High\ L2, \dots, High\ L18\}$
Dew Point	20	$\{Low, Medium, High\ L1, High\ L2, \dots, High\ L18\}$
Wind	2	$\{Low, High\}$

Table 2. Fuzzy linguistic sets associated with fault sequence variables.

Based on (5), the degree of a fuzzy input value belonging to an element of the corresponding linguistic set is assigned by the corresponding membership function. For simplicity, the fuzzy membership functions used in this thesis for fuzzifying FIS input variables are chosen as Gaussian. This is further elaborated in Section 2.5.3.

2.3.2 Rule Generation

Fuzzy rules are a set of linguistic IF-THEN rules which connect outputs of the fuzzification stage (antecedents) to the fuzzy outputs (consequences) with the corresponding degree of membership

function. Multiple antecedents can be connected via the logical operators AND or OR to obtain an output. Principles by which fuzzy rules are derived are considered the defining feature of the corresponding FIS. Therefore, to avoid bias in the FIS design, four strategies are used to generate fuzzy rules in this thesis based on the approaches introduced in Chapter 1. These are described below.

- Automatic rule generation in which the frequency of each fault sequence is calculated in the training set and normalized to $[0,1]$. Then, all fault sequences with frequencies equal to or greater than 0.5 are selected for the FIS rule list and the corresponding normalized frequencies are used as initial weights. Fault sequences whose frequencies are lower than 0.5 are discarded.

- Expert rule generation which relies only on expert knowledge to construct the rule list. Rules are generated through investigation of the power network. The process of using the expert rule generation strategy to construct the corresponding rule set used in this thesis is elaborated later in this chapter.

- Large strategy, which is similar to the automatic strategy but only those sequences selected for the FIS rule list with normalized frequency equal or larger than 0.8 are used. This is done based on the assumption that selecting rules with higher frequencies results in faster learning (this is validated in Chapter 3). Comparing the accuracy and computational cost of this strategy with other rule generation strategies can provide an understanding of the effects of rule list size on the performance of the diagnostic framework.

- Hybrid strategy, which combines expert and large rule generation strategies. This is used to examine the performance of the diagnostic framework with a rule list which is a combination of machine intelligence and expert knowledge.

In these strategies, the rule matrix has $m + n + 2$ columns. The first m columns are the system inputs (antecedents) and the next n columns are the outputs of the system (consequences). Column $m + n + 1$ contains the weight that is applied to the rule which is a number between zero and one. Column $m + n + 2$ determines the operator used for combining the antecedents. This is 1 if the rule operator is AND and 2 if it is OR.

Since the process of selecting the training/test set involves a random sampling of the original fault database, rule sets generated using automatic, large-only and hybrid strategies will have different sizes and elements each time the diagnostic algorithm is initialized. However, the expert strategy has a fixed number of elements in the rule list. Using the knowledge of electric power system dynamics and Table 2, the following fuzzy clauses are used as the expert rules [6] [59].

- Clause 1

  If $Climate\ Condition$ is Rain and Lightning, AND

  $Temperature$ is High L18, AND

  $Dew\ Point$ is High L18, AND

  $Wind$ is High,

  then,

  $Fault\ Type$ is Transient.

- Clause 2

If $Climate\ Condition$ is Rain and Lightning, AND

$Temperature$ is Low, AND

$Dew\ Point$ is Low, AND

$Wind$ is Low,

then

$Fault\ Type$ is Transient.

- Clause 3

If $Tree\ Trimming$ is Insufficient L9, AND

$Aging$ is Aged L11, AND

$Duration$ is Long L18, AND

$Current$ is Low,

then

$Fault\ Type$ is Persistent.

- Clause 4

If $Load\ Denisity$ is Dense L17, AND

$Current$ is High L18, AND

$Energy$ is High L18,

then

$Fault\ Type$ is Anomaly.

Based on Table 2 and the information in Section 2.3.2, these clauses form the following expert rule matrix (or expert rule list) after quantization

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 14 & 20 & 20 & 2 & 2 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 14 & 1 & 1 & 1 & 2 & 0 & 0 & 1 & 1 \\ 0 & 0 & 10 & 0 & 12 & 0 & 0 & 20 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 20 & 0 & 0 & 20 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 \end{bmatrix}$$

The number of elements in the expert rule list can be increased to improve the performance of the corresponding framework. However, for simplicity and computational cost reasons, 4 elements are used in the expert rule list used here.

Unlike the expert strategy, automatic, large and hybrid strategies do not have a fixed number of elements in their rule list. Although this number changes because of the data randomization before the learning stage, the automatic strategy results in the largest number of elements on average. The number of elements in the rule list affects the accuracy and computational cost of the diagnostic framework which can be seen in the results of the automatic and large strategies in Chapter 3.

2.3.3 De-fuzzification

The process of translating fuzzy sets and corresponding membership functions into a crisp set is called de-fuzzification. In fuzzy control systems (such as the FIS used in this thesis), the result of de-fuzzification is a non-fuzzy control action. Here, the inputs for de-fuzzification are fuzzified fault sequences together with the fuzzy rule list of choice. The output of de-fuzzification is a length 3 vector from which the associated fault type can be inferred. The de-fuzzification method used in this thesis is the center of gravity (centroid) method which can be expressed as

$$Q_d^T = \frac{\int_q \mu_{X\prime}(q) q dq}{\int_q \mu_{X\prime}(q) dq} \quad (6)$$

where $\mu_{X\prime}$ is the degree of the membership function of output fuzzy set X', obtained by the FIS [65].

Fuzzy inference is typically either Mamdani or Takagi-Sugeno. In the first type, the output is a fuzzy set with corresponding membership function and is suitable for Multi Input Multi Output (MIMO) systems. In the second, the output is either a crisp value or a weighted average of the rule consequences which is only suitable for Multi Input Single Output (MISO) system. Since the SG diagnostic problem is MIMO, Mamdani inference is selected for building the FIS. Mamdani inference also benefits from better interpretability of the rule consequences compared to Takagi-Sugeno [64]. To compute the output of a Mamdani FIS using fuzzy inputs, the following steps are taken.

- Construct a fuzzy set of rules
- Combine the fuzzified inputs based on the fuzzy rules to determine the rule weights
- Combine rule weights and output membership functions to determine the consequences of the rules
- Combine the consequences to find the output distribution
- De-fuzzify the output distribution [44].

In Figure 6, the fuzzy input variable wind is shown with a Gaussian fuzzy membership function and a fuzzy linguistic set $X\{Low, High\}$. Fuzzification takes the crisp value for wind calculated from measurements (Figure 2) and returns a membership function (μ_A) according to (5). A Gaussian distribution is used here for all membership functions of the input sets. The reason for this choice is simplicity and speed in design. For the output sets, a triangle membership function is employed since it is assumed that outputs (fault types) cannot overlap with each other [64] [66].

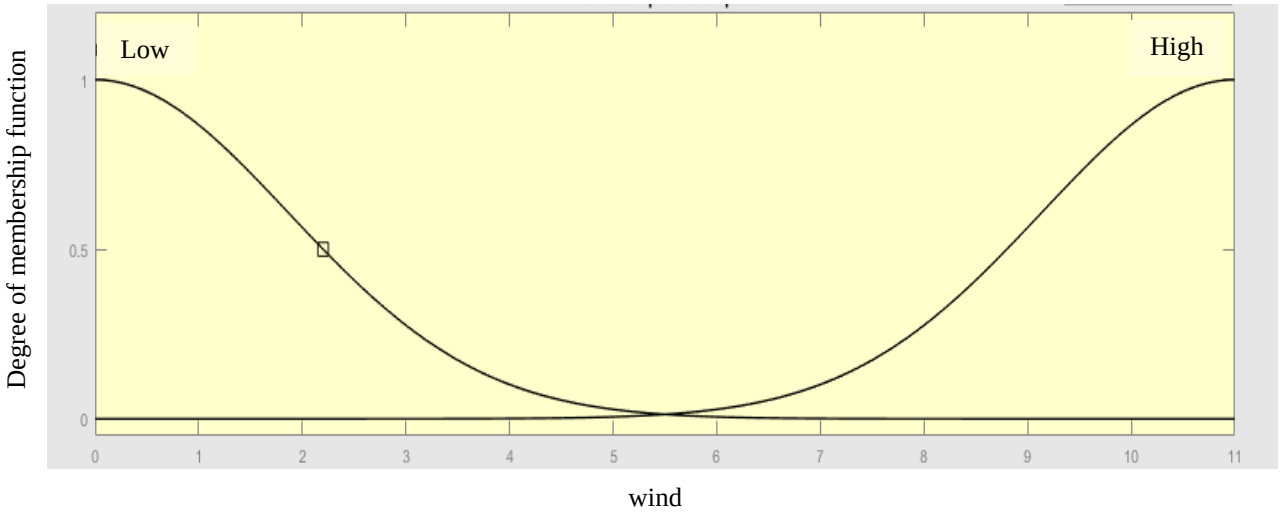

Figure 6. Fuzzy membership function of the variable wind, with two fuzzy linguistic variables which are Gaussian distributed.

2.4 Learning

In the context of the SG fault diagnosis problem, learning can be viewed as optimizing the weights of the fuzzy rule list. Once the fuzzy rule set is constructed, the training set is used to optimize its weights. This is done via an ANN that uses reinforcement learning in each iteration to update rule weights thorough the following steps. Note that steps three to five comprise the reward scheme which is explained in the following section.

1. The samples of the training set are chosen randomly to avoid bias.
2. Weights are initiated to the frequency ($\in [0,$ size of the training set]) of the associated fault sequence in the rule generation set mapped to $[0, 1]$.
3. The input part of the first fault sequence of the training set is fed to the FIS and the corresponding output (inferred fault type) is obtained.
4. The output obtained in step 3 together with the actual output from the training set are used to calculate the reward.
5. The reward is used to update the rule weights, and the updated rule weights are fed back (reinforced) to the FIS.
6. Steps 3 to 5 are repeated with the next fault sequence in the training set.

The process continues until all fault sequences in the training set have been processed.

2.4.1 Reward Scheme

The reward scheme reinforces learning to the rule weights and is done in three steps. First, the difference (distance) between the FIS output and the actual output is calculated. This is used to compute the reward associated with each rule. In the third step, rule weights are updated using the computed reward. These steps are described below.

2.4.1.1 The Difference Between Outputs

Let p and q be the n-dimensional vectors associated with the actual and simulated (FIS) outputs of a fault sequence, respectively. The Mahalanobis (Euclidean) distance between them from (2) is

$$D = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \quad (7)$$

In this thesis, $n = 3$ since there are three types of faults, therefore

$$D = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + (q_3 - p_3)^2} \quad (8)$$

2.4.1.2 The Reward

For simplicity, the reward is proportional to the distance between the outputs and is given by

$$R(k, i) = w_k(i) \times \frac{(D_M - D_O(i))}{D_M} \qquad 1 \leq k \leq r \quad (9)$$

where k is the rule number in the rule list and $w_k(i)$ is the ANN learning rate of rule k in the ith iteration of the training. The learning rate should be set to a sufficiently small number for the ANN to converge. Firing strength is a variable in the MATLAB Fuzzy Logic Toolbox™ (the software used for building the FIS and simulating the SG in this thesis), and is used as the ANN learning rate. The firing strength of rule k at iteration i reflects the degree by which rule k has been utilized in producing the FIS output in iteration $i - 1$ of learning. Therefore, $w_k(i)$ regulates the reward in the ith iteration based on the relevance of rule k on the simulated output at the $(i - 1)$th iteration. It has a value between 0 and 1, with 0 indicating no relevance and 1 indicating the highest relevance to $D_o(i)$. In (9), $D_O(i)$ is the Mahalanobis distance between the actual and simulated outputs associated with the ith training iteration, D_M is the maximum possible distance between the simulated and actual outputs, and r is the number of rules.

Since the output is a 3-dimensional vector belonging to the set $\{(0,0,1), (0,1,0), (1,0,0)\}$, and the largest simulated output possible is also a 3-dimensional vector of the form $(1,1,1)$, D_M can be calculated as

$$D_M = \sqrt{(1-0)^2 + (1-0)^2 + (0-0)^2} = \sqrt{2} \qquad (10)$$

so (9) becomes

$$R(k,i) = w_k(i) \times (1 - \frac{D_O(i)}{\sqrt{2}}) \qquad (11).$$

Thus, the reward at each iteration is only a function of the firing strength of rule k and the distance between the inferred and actual outputs associated with fault sequence i. The reward is larger when $D_O(i)$ is small as a smaller distance between the inferred and actual outputs indicates a more accurate weight assignment and so is reinforced.

2.4.1.3 Updating the Rule Weights

The calculated rewards are used to update the corresponding weights. The updated rule matrix is then used in the next iteration of training. The rule matrix has dimensions $k \times (m+n+2)$ and the weight update at each iteration can be formulated as

$$Rule_{new}(k) = [Rule_{old}(k,1), Rule_{old}(k,2), \dots, Rule_{old}(k,m+n), Rule_{old}(k,m+n+1) + R(k), Rule_{old}(k,m+n+2)] \qquad (12)$$

where $R(k)$ is the reward associated with rule k after the last iteration of training, m is the number of inputs and n is the number of outputs associated with the fault sequence. Learning continues with the input part of the next fault sequence in the training set. A flowchart of the learning algorithm is given in Figure 7.

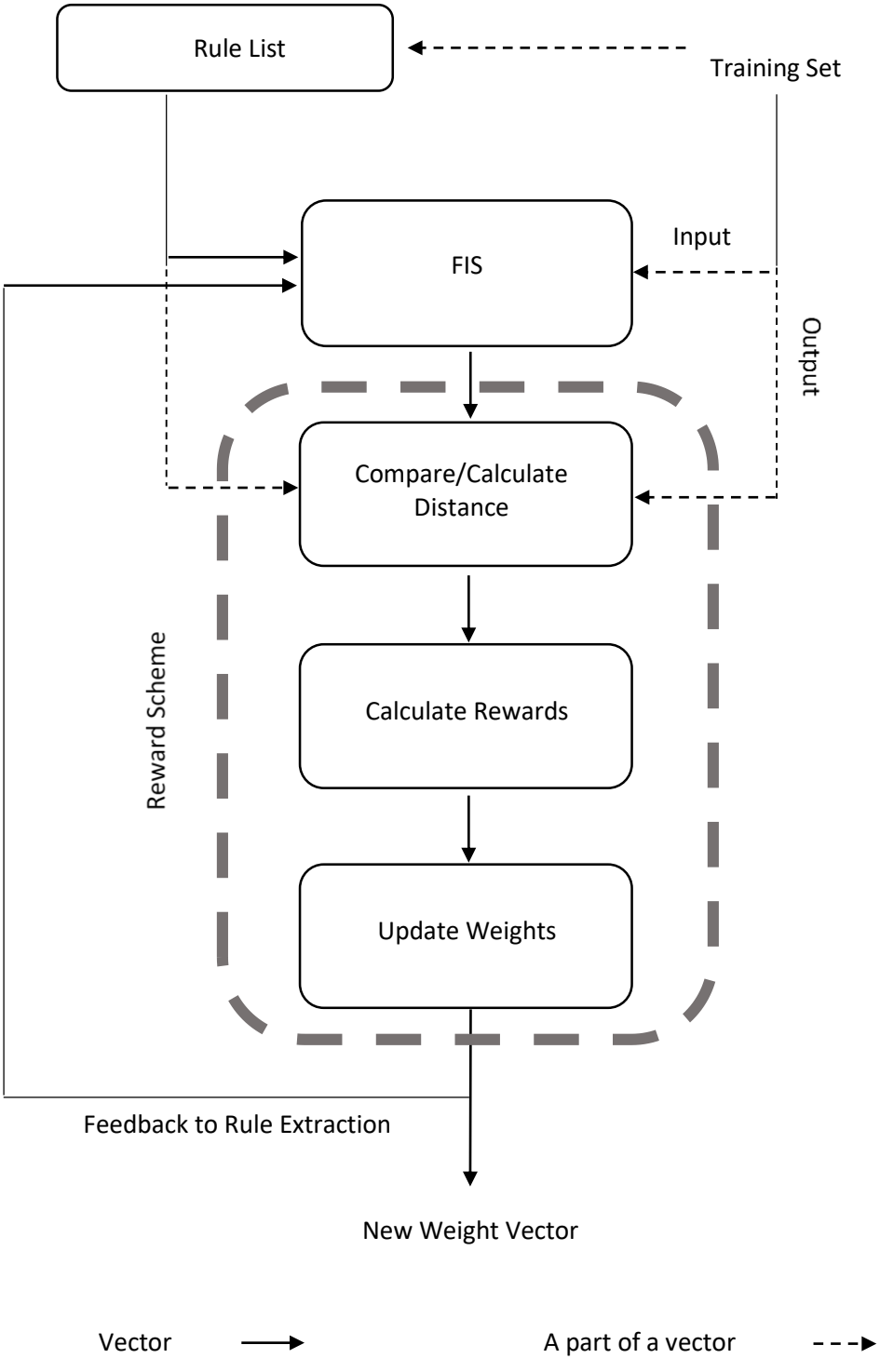

Figure 7. Reinforcement learning flowchart.

The designed FIS operates as an inference engine for the diagnostic framework when a new fault sequence enters the fault database. The new fault sequence is first fed into the diagnostic framework that has been trained as explained above. The output of the framework is the fault type associated with the new fault sequence in the form of a vector. The new fault sequence is then used for real-time training of the FIS.

Chapter 3: Results and Discussion

In this chapter, the effects of several design parameters on the performance of the diagnostic framework are investigated. As shown in the flowchart shown in Figure 1, these parameters are

- training to test set ratio, and
- rule generation strategy.

The other design parameters that can play a role in the performance of the diagnostic framework are left for the next phases of the research and do not fit in the scope of this thesis. These parameters are

- reward schemes
- cluster numbers, and
- shape of the fuzzy membership functions

To investigate the effect of changing each of these parameters on the performance of the diagnostic framework, the learning error, computational cost and testing accuracy of each of the designs are measured and compared. In the following, definitions of these performance measures are given.

Learning Error

In the learning stage shown in Figure 1, a fault sequence from the training set is used whose output is inferred by the FIS. This is then compared to the actual output from the training set using the Mahalanobis distance (D_o) which indicates the error between them. Therefore, D_o can be used as a measure of the error of the corresponding learning. A small value of D_O indicates a low learning

error and a large D_O indicates a high learning error. Minimum learning error is achieved when $D_O = 0$ which is the minimum distance possible. The maximum learning error is achieved when $D_O = D_M$. To compare the learning errors across the different learning strategies used in this thesis, the diagnosis distances are normalized using

$$[0, D_M] \rightarrow [0, 100] \quad \forall D_O \qquad (13)$$

Computational Cost

The overall computational cost of the diagnostic framework consists of two parts

- the computational cost of the learning process (performed on the training set), and
- the computational cost of the diagnosis process (performed on the test set).

However, the bottleneck for computational cost is the learning process since it requires running the FIS as many times as the number of fault sequences in the training set while in the diagnosis process this is done only once on the test set. Therefore, only the learning process is considered to compare the computational costs. Computational cost is measured as the iteration number after which the learning error remains within a desirable range. In this thesis, this range is defined as 10%. This was obtained empirically based on the observation that after a certain point, changes in the learning error were very small (Figures 8 to 15). In this regard, various ranges were evaluated to obtain the best range.

The computational cost of the diagnosis process was measured in MATLAB in seconds. Results on the order of seconds or less indicate that the expert strategy has the best performance followed by the large, hybrid and automatic strategies. Moreover, lower training to test set ratios resulted in

lower computational cost. Further, this cost is correlated to the number of the rules in the rule lists as well as the size of the test set.

Accuracy (Testing Accuracy)

After rule weight optimization, the resulting rule list is ready for use in the FIS to diagnose fault types. To evaluate the performance of this FIS, the test set constructed in Section 2.4 is used. The input parts of the fault sequences in this set are fed into the optimized framework. The accuracy of this framework is measured by comparing the inferred outputs with the actual outputs in the test set (i.e. how accurately it diagnoses fault types). The accuracy is given by

$$AC = 100 \times \left(1 - \frac{\sum_{i=1}^{n} D_i}{n \times D_M}\right)\% \qquad (15)$$

where n is the number of fault sequences in the test set and D_i is the Euclidean distance between the generated and the actual outputs of the ith element of the test set. D_M is the maximum possible distance between the inferred and actual outputs given in (10) and is used as a normalizing factor to map the value of $\frac{\sum_{i=1}^{n} D_i}{n \times D_M}$ to [0, 1]. $D_i = 0$ for $1 \leq i \leq n$, results in $AC = 100\%$ and $D_i = D_M$ for $1 \leq i \leq n$ results in $AC = 0\%$.

The following steps were used to simulate the diagnostic framework in MATLAB according to the flowcharts in Figures 1 and 7.

- Data cleaning was performed manually which consisted of removing bad and incomplete data. This was a very small percentage of the entries (approximately 1%).
- Data clustering and quantization were done using the jump algorithm (Section 2.2.2) which was coded manually in MATLAB.

- Train/test set selection and rule generation were done using the four strategies, each of which was coded separately in MATLAB.
- As explained in Section 2.4, a single layer ANN with backpropagation was used for learning the optimized rule weights in the constructed rule list which was manually coded in MATLAB. The ANN initial weights were set to the values corresponding to each rule list, explained in Section 2.3.3 and the learning rate was set to the firing strength (Section 2.4.1.2).
- Fuzzy membership functions were designed using the schematics option of the Fuzzy Logic Toolbox in MATLAB (Figure 6). The FIS was coded manually in MATLAB.
- Testing was done in MATLAB using the FIS constructed in the previous stage.

3.1 Training to Test Set Ratio

In this section, the diagnostic performance of three different training to test set ratios is investigated. These ratios are $50/50$, $25/75$ and $10/90$. The first number refers to the percentage of the training set elements selected from the original fault database while the second number refers to the percentage of test set elements.

3.1.1 Learning Error

Figures 8 to 15 show the effect of changing the training to test set ratio on the learning error. These give the normalized D_o obtained from (13) for each iteration of the learning stage. In these figures, the horizontal axis shows the iteration number and the vertical axis is the associated D_o as the distance (or error) between the actual and FIS outputs at each iteration. The blue lines indicate no climate data is included while the red lines indicate climate data is included.

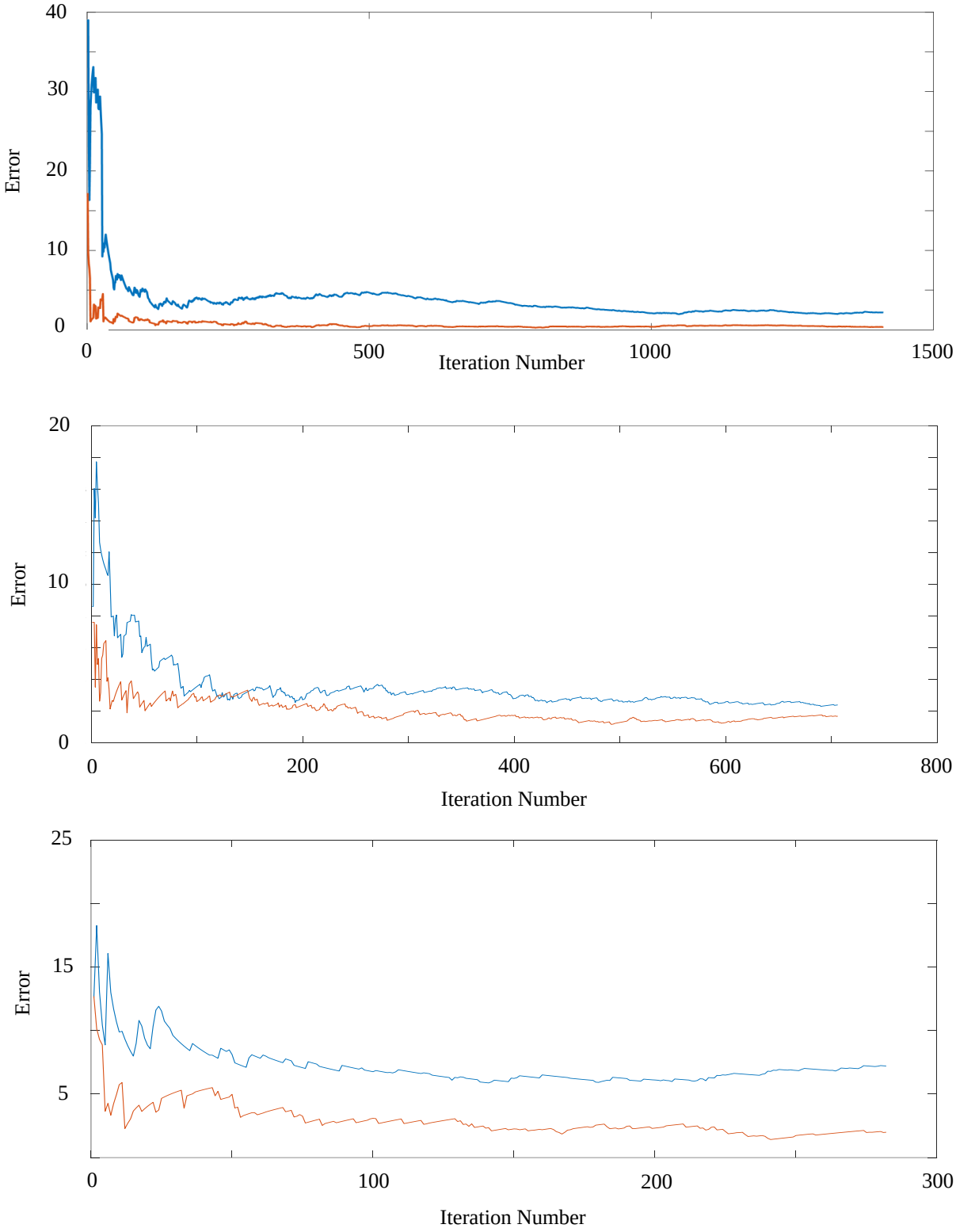

Figure 8. Learning error for the Azadi station with a) 50/50 (top), b) 25/75 (middle) and c) 10/90 (middle) training to test set ratios and automatic rule generation. The blue lines indicate no climate data is included while the red lines indicate climate data is included.

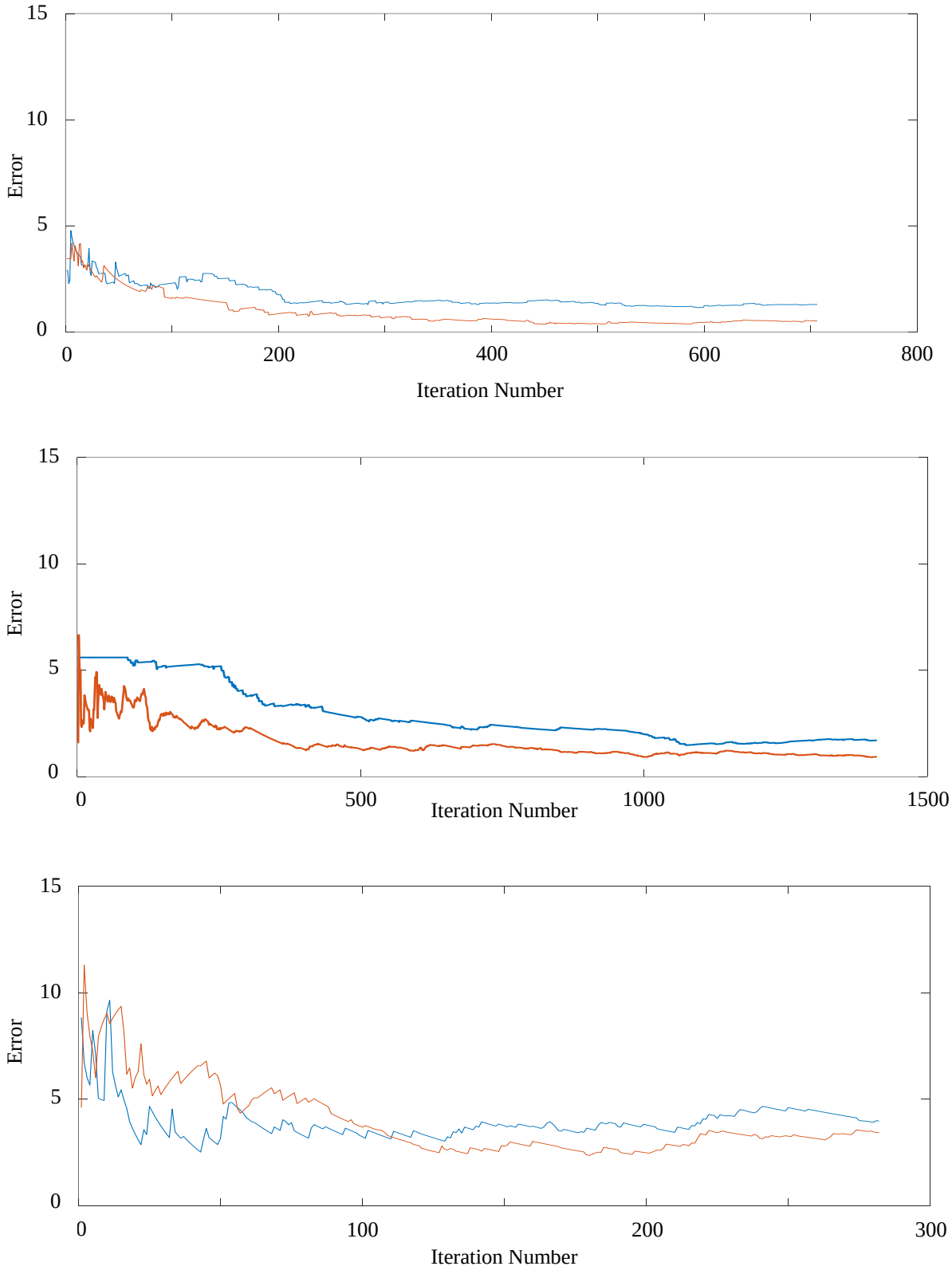

Figure 9. Learning error for the Azadi station with a) 25/75 (top), b) 50/50 (middle) and c) 10/90 (bottom) training to test set ratios and large rule generation. The blue lines indicate no climate data is included while the red lines indicate climate data is included.

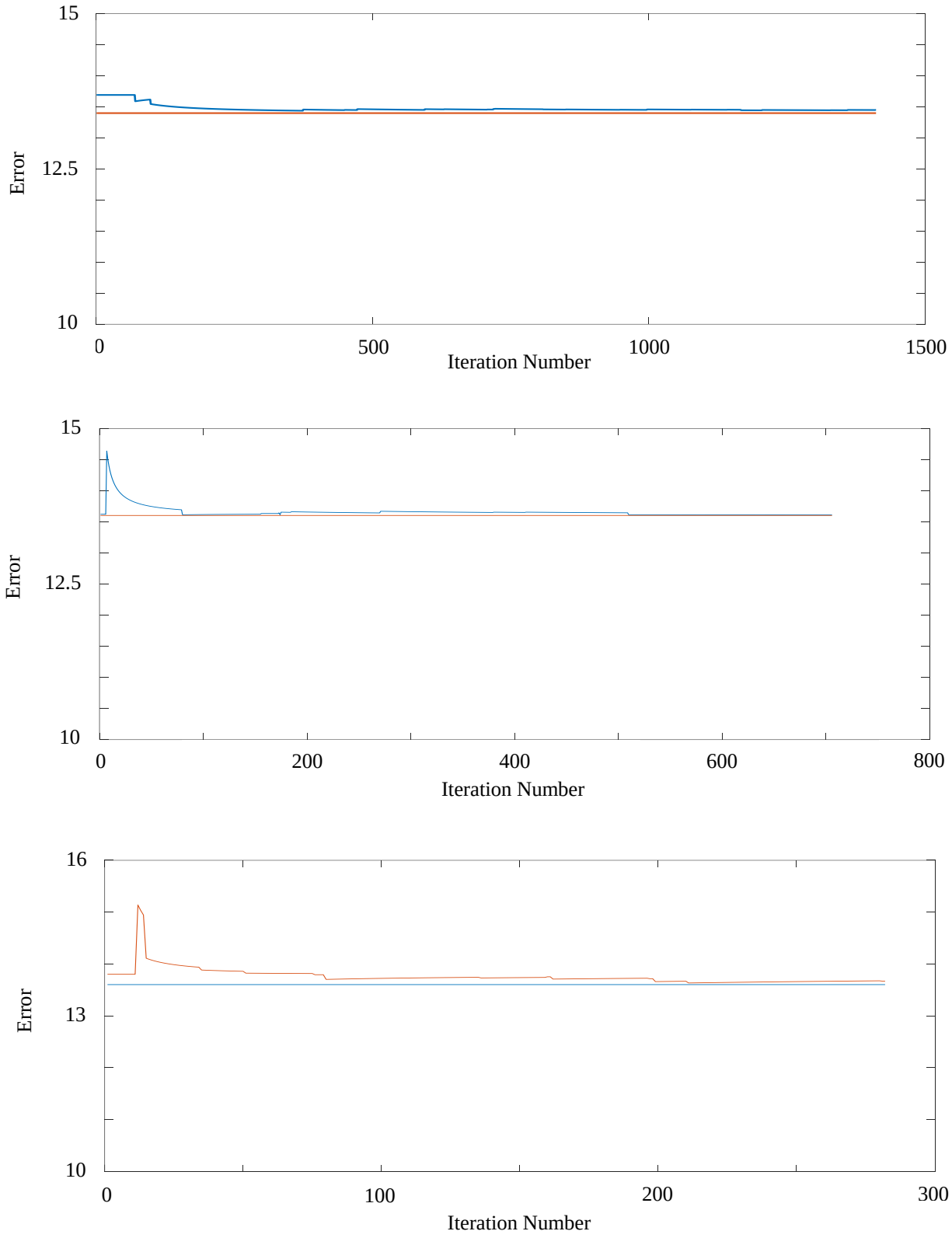

Figure 10. Learning error for the Azadi station with a) $50/50$ (top), b) $25/75$ (middle) and c) $10/90$ (bottom) training to test set ratios and expert rule generation. The blue lines indicate no climate data is included while the red lines indicate climate data is included.

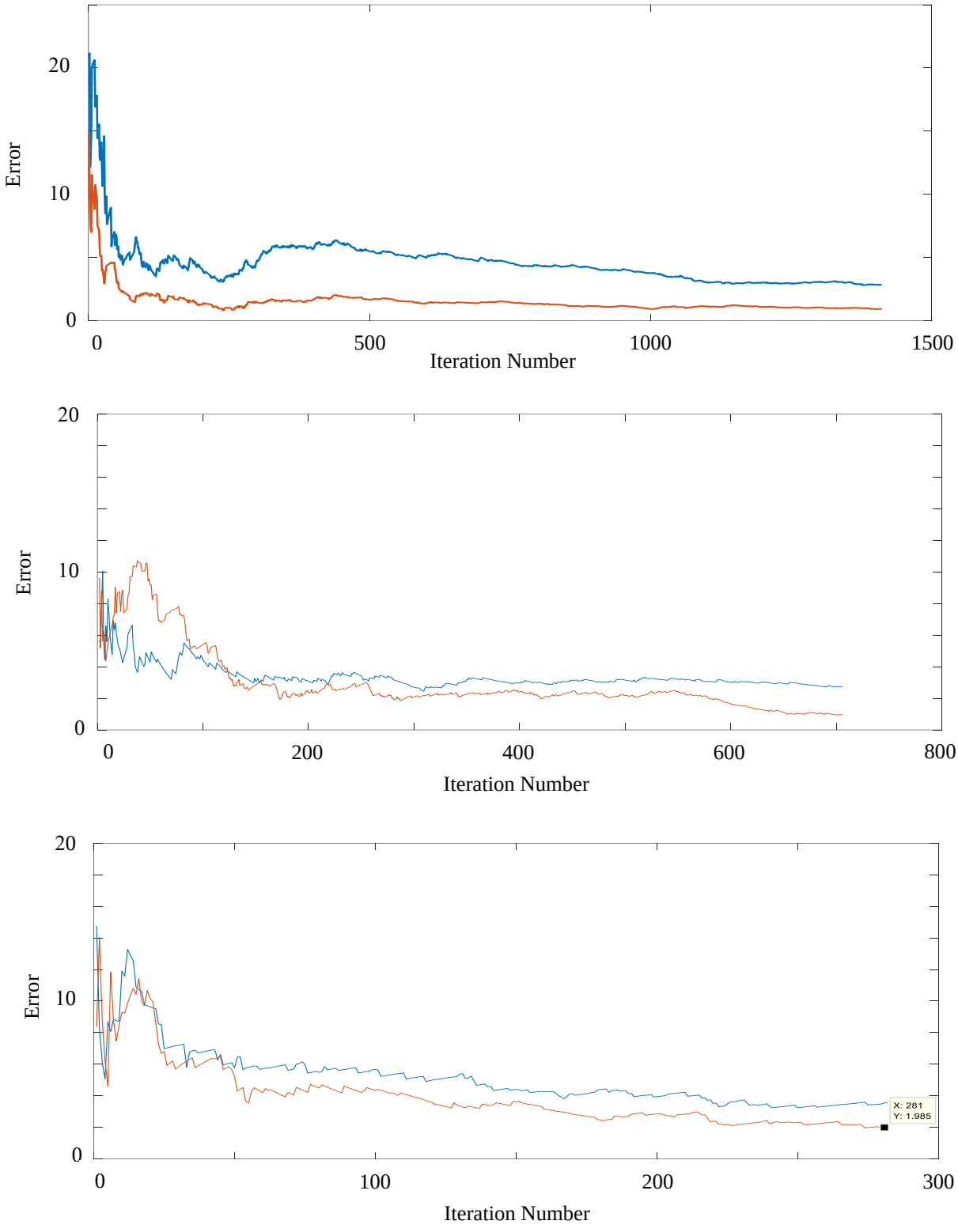

Figure 11. Learning error for the Azadi station with a) 50/50 (top), b) 25/75 (middle) and c) 10/90 (bottom) training to test set ratios and hybrid rule generation. The blue lines indicate no climate data is included while the red lines indicate climate data is included.

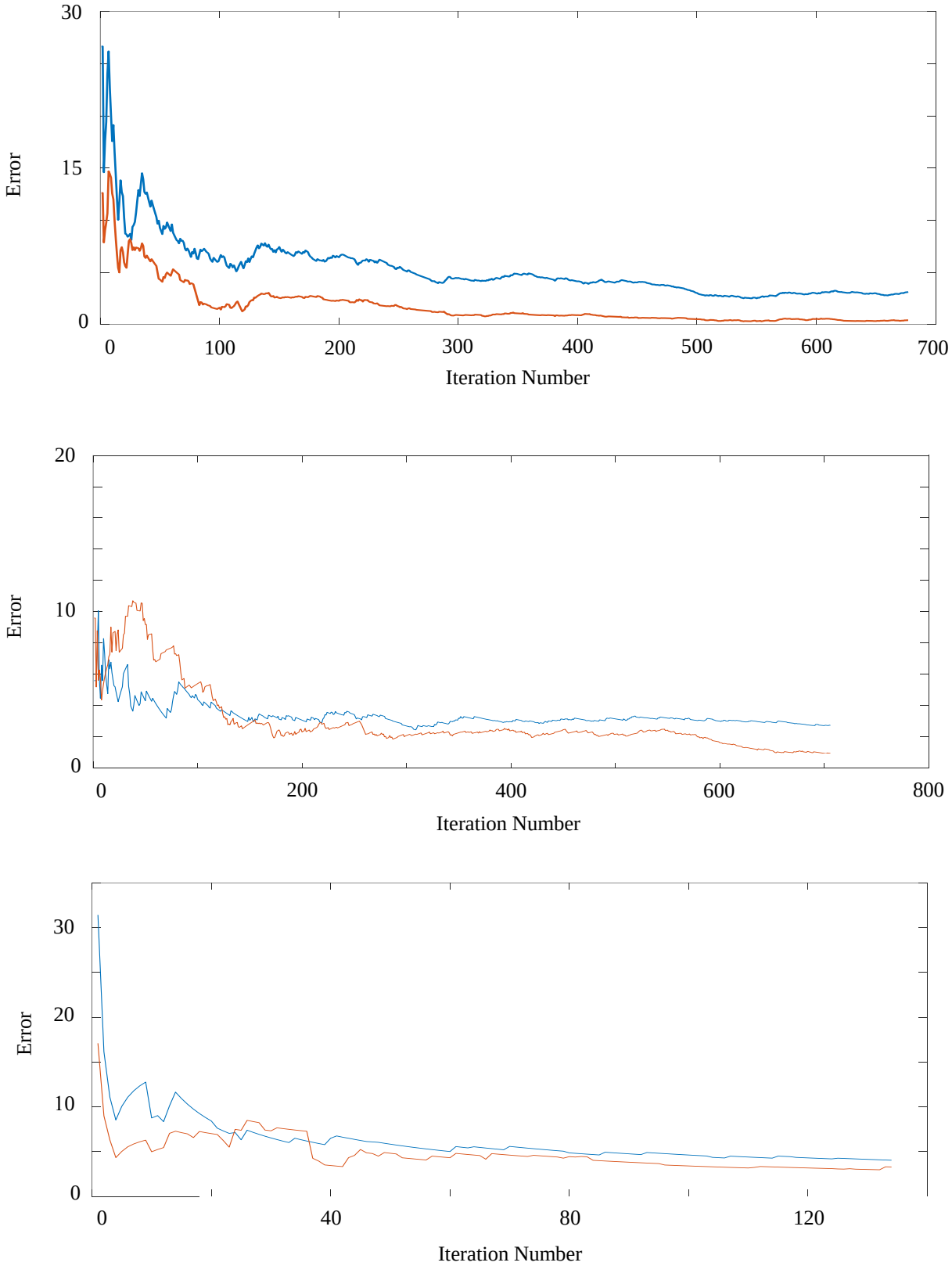

Figure 12. Learning error for the Shemiran station with a) $50/50$ (top), b) $25/75$ (middle) and c) $10/90$ (bottom) training to test set ratios and automatic rule generation. The blue lines indicate no climate data is included while the red lines indicate climate data is included.

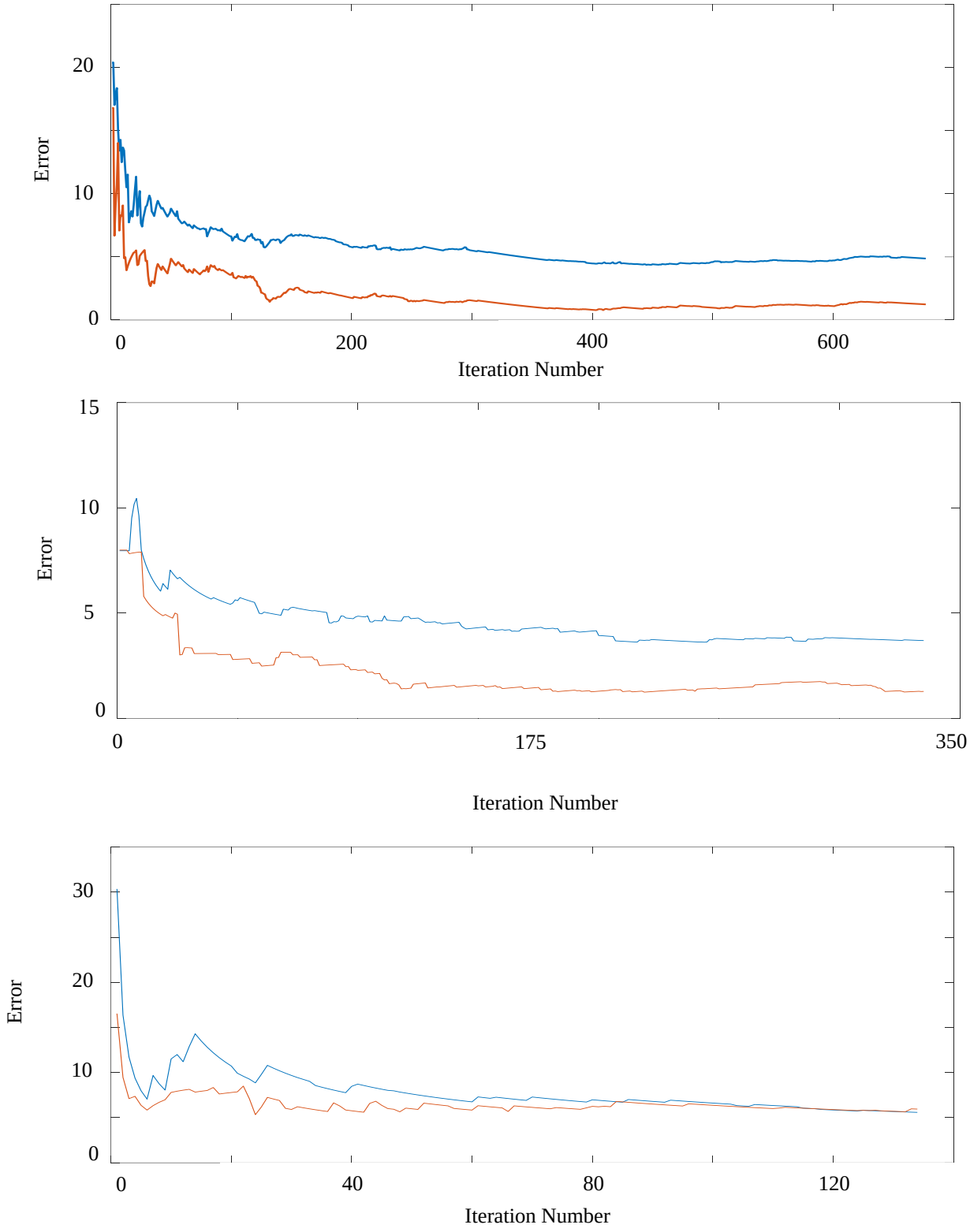

Figure 13. Learning error for the Shemiran station with a) 50/50 (top), b) 25/75 (middle) and c) 10/90 (bottom) training to test set ratios and large rule generation. The blue lines indicate no climate data is included while the red lines indicate climate data is included.

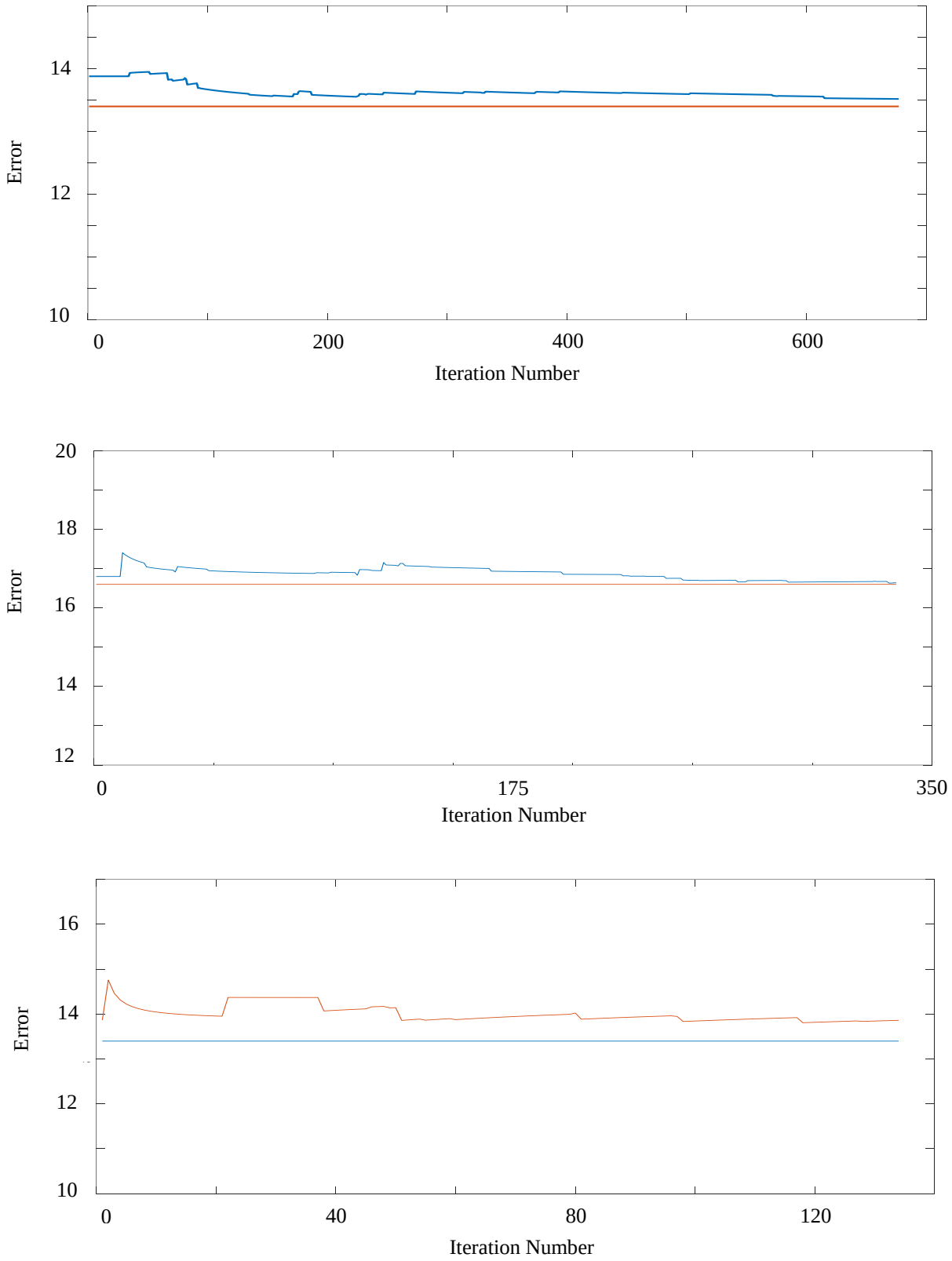

Figure 14. Learning error for the Shemiran station with a) $50/50$ (top), b) $25/75$ (middle) and c) $10/90$ (bottom) training to test set ratios and expert rule generation. The blue lines indicate no climate data is included while the red lines indicate climate data is included.

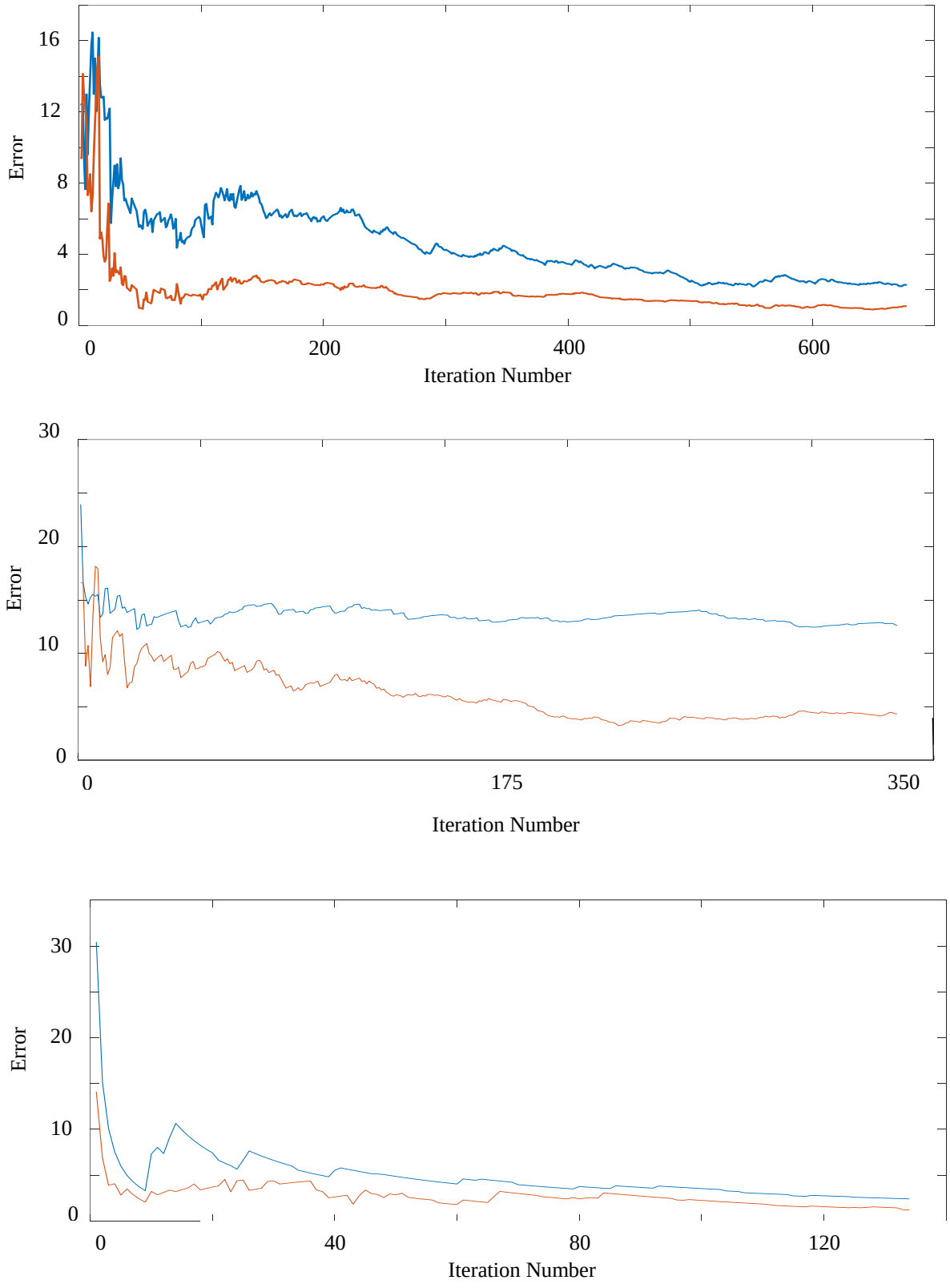

Figure 15. Learning error for the Shemiran station with a) $50/50$ (top), b) $25/75$ (middle) and c) $10/90$ (bottom) training to test set ratios and hybrid rule generation. The blue lines indicate no climate data is included while the red lines indicate climate data is included.

Figures 8 to 11 show the learning error for the Azadi station with varying training to test ratios and automatic, large, expert and hybrid rule generation strategies, respectively. The first diagram in each of these figures (a) is associated with $50/50$, the second diagram (b) with $25/75$ and the third diagram (c) with $10/90$ training to test ratios respectively. These results show that the learning error decreases as the training to test ratio increases regardless of the rule generation strategy. This can be seen in the final learning error. Figures 12 to 15 show the learning error for Shemiran station with varying training to test ratios and automatic, large, expert and hybrid rule generation strategies, respectively. Moreover, Figures 8 to 15 clearly show that including climate information in the fault database decreases the learning error regardless of the training to test ratio. The effect of the training/test ratio on the average learning error is summarized in Table 3. To investigate this effect only, the averaging is performed over all four learning strategies.

Station	Training/test ratio	Final error (Climate)	Final Error (No Climate)
Azadi	50/50	4.5	5.3
	25/75	4.8	6.4
	10/90	5.5	7.2
Shemiran	50/50	4.5	7.0
	25/75	6.2	8.3
	10/90	6.8	9.6

Table 3. Average learning error for the three training/test ratios.

Table 3 shows that increasing the training/test ratio results in a lower final error for both stations and all the rule generation strategies which is consistent with Figures 8 to 15. Table 3 also shows that including climate information decreases the final error. This is most significant in the Shemiran station with a $50/50$ training/test ratio where the final error is 1.6 times lower when climate data is included.

3.1.2 Computational Cost

In this section, the computational cost of various training to test set ratios for each of the stations is evaluated. Since the size of the training sets for each of the learning strategies for each station is fixed, averaging is performed across all the learning strategies for fixed training to test ratios. The results are shown in Table 4. To show the effect of climate information, averaging is performed separately for climate and no climate conditions. However, the comparison is done based on the overall averages. In some cases, training did not improve the error (such as the expert rule generation strategies), so the computational cost is not defined and therefore the corresponding entry is excluded from the averaging. These cases are marked with an asterisk in Table 4. This table shows the computational cost results for each strategy followed by Table 5 which shows the averaging results.

50/50 Ratio	Automatic	Expert	Large	Hybrid
Shemiran	250	-	157	224
Shemiran (No Climate)	262	596	195	408
Azadi	56	-	745	63
Azadi (No Climate)	67	797	1014	742

25/75 Ratio	Automatic	Expert	Large	Hybrid
Shemiran	166	246	132	262
Shemiran (No Climate)	188	327	207	188
Azadi	114	66	454	310
Azadi (No Climate)	350	618	287	589

10/90 Ratio	Automatic	Expert	Large	Hybrid
Shemiran	29	126	46	46
Shemiran (No Climate)	84	118	85	95
Azadi	207	263	243	183
Azadi (No Climate)	211	80	206	160

Table 4. Computational cost of the learning stages of the diagnostic framework with a) $50/50$, b) $25/75$ and c) $10/90$ training to test ratios.

Station	Training to Test Ratio	Average Computational Cost (No Climate)	Average Computational Cost (Climate Included)	Average Computational Cost (Overall)
Azadi	50/50	288	655	498
	25/75	236	461	348
	10/90	224	164	194
Shemiran	50/50	210	340	298
	25/75	201	227	214
	10/90	62	95	79

Table 5. Average computational cost of the learning based on station, training to test ratio, and inclusion of climate information.

These results show that the average computational cost decreases as the training to test ratio decreases. This is consistent with the fact that training a small set requires fewer iterations compared to a larger one [67]. This relationship is depicted in Figure 16 in which the blue and red lines correspond to the Azadi and Shemiran stations, respectively. Moreover, comparing Table 5 with Figures 8 to 15 reveals that even though a smaller training set results in a lower computational cost, it also results in a higher learning error. Therefore, there is a training size learning accuracy tradeoff.

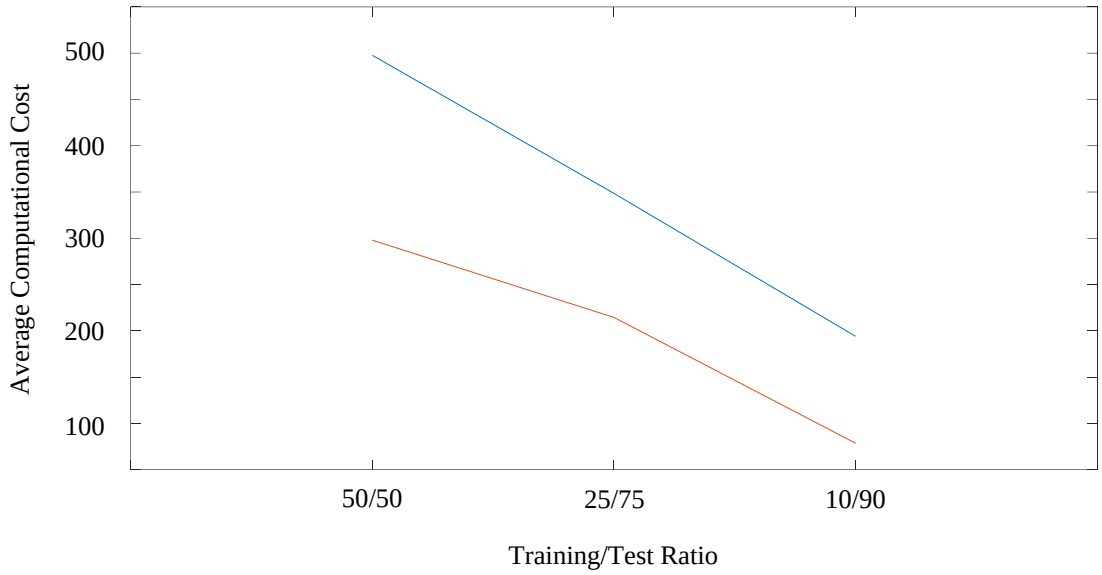

Figure 16. Average computational cost for three training to test ratios for the Azadi (blue) and Shemiran (red) stations.

3.1.3 Accuracy

In order to investigate the effect of varying the training to test ratios on the accuracy of the framework, the average testing accuracy for each of the ratios is evaluated. The averaging is performed for each of the various rule generation strategies. To show the effect of climate information, the averaging is done with and without climate information. Table 6 shows the testing accuracy results with the three training to test ratios for the Azadi and Shemiran stations for each of the rule generation methods. Table 7 shows the average accuracies for both stations and includes both climate and no climate information.

50/50 Ratio	Automatic	Expert	Large	Hybrid
Shemiran	87%	65%	86%	86%
Shemiran (No Climate)	63%	34%	59%	68%
Azadi	92%	65%	89%	89%
Azadi (No Climate)	46%	35%	80%	68%

a) 50/50

25/75 Ratio	Automatic	Expert	Large	Hybrid
Shemiran	77%	65%	68%	80%
Shemiran (No Climate)	61%	32%	54%	66%
Azadi	64%	65%	81%	78%
Azadi (No Climate)	41%	34%	73%	64%

b) 25/75

10/90 Ratio	Automatic	Expert	Large	Hybrid
Shemiran	65%	57%	53%	56%
Shemiran (No Climate)	57%	18%	43%	48%
Azadi	33%	64%	73%	72%
Azadi (No Climate)	19%	20%	69%	54%

c) 10/90

Table 6. Accuracy for the three training to test ratios.

Training/test ratio	Automatic	Expert	Large	Hybrid	Average
50/50	72%	50%	79%	78%	70%
25/75	61%	49%	69%	72%	63%
10/90	44%	40%	60%	58%	50%

Table 7. Average testing accuracy for the rule generation strategies.

Table 7 shows that the accuracy decreases as the training set size decreases. This is because a smaller training set contains less system information which results in a higher learning error and consequently, a worse rule list. A worse rule list results in lower testing accuracy which can be seen in Table 7. The relation between the training set size and the testing accuracy is shown in Figure 17. This figure depicts the accuracy versus the training to test ratio. This shows that the accuracy decreases with a steeper slope as the training size decreases. The effect of climate information on testing accuracy can be seen in Table 5. It shows that the testing accuracy increases when climate information is included. This is most significant with the expert method for the Azadi station where climate information increases the accuracy by 320% when a 10/90 training to test ratio is used. This is followed by the expert method for the Shemiran station with a 10/90 training to test ratio where the increase is 317%. Next are the expert strategy for the Shemiran and Azadi stations with a 25/75 training to test ratio where climate information increases the accuracy by 230% and 190%, respectively. The smallest increase in accuracy is 106% with the large strategy for the Azadi station with a 10/90 training to test ratio, followed by 110% with the large strategy for the Azadi station with a 25/75 training to test ratio and 111% with the large strategy for the Shemiran station with a 50/50 training to test ratio. These results suggest that the accuracy of the expert method benefits the most from the inclusion of climate information while the large method

benefits the least. This may be because while the other methods are capable of mapping input/output relations through learning, the expert method relies more on the initial information to accurately map the input data to fault types. Therefore, a small change in the a priori information results in a significant change in the performance of the expert method (hence, it is considered less intelligent).

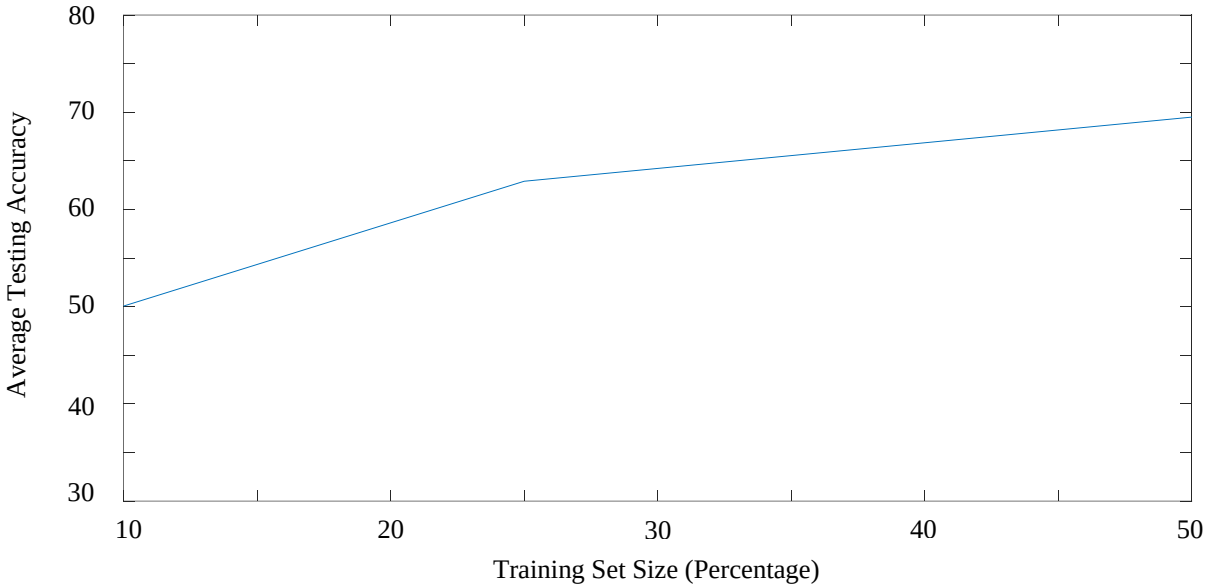

Figure 17. Average testing accuracy versus the training set size.

3.2 Rule Generation Strategy

In this section, the effect of the rule generation strategies on the performance is investigated. To this end, a rule list is generated using a given strategy and is used in the learning stage. In each learning iteration, the output of a fault sequence belonging to the training set is inferred by the FIS. The difference between the inferred and actual outputs from the training set is used to update the weights in the rule list. The updated rule list is then used in the next learning iteration with the next fault sequence. This process continues until all the fault sequences in the training set are processed

for learning. The same procedure is performed for the other rule generation strategies to optimize the corresponding rule lists.

3.2.1 Learning Error

In this section, the effect of the rule generation strategies on the learning error is investigated. Figures 8 to 15 show the effect of the rule generation strategies on the learning error. The climate included curves (red lines) in Figures 8.a, 9.a, 10.a and 11.a (Azadi station data) are presented in Figure 18. This shows the error for the four rule generation strategies when climate information is included and a training to test ratio of $50/50$. This figure suggests that the hybrid strategy results in a lower learning error, followed by the automatic, large and expert strategies.

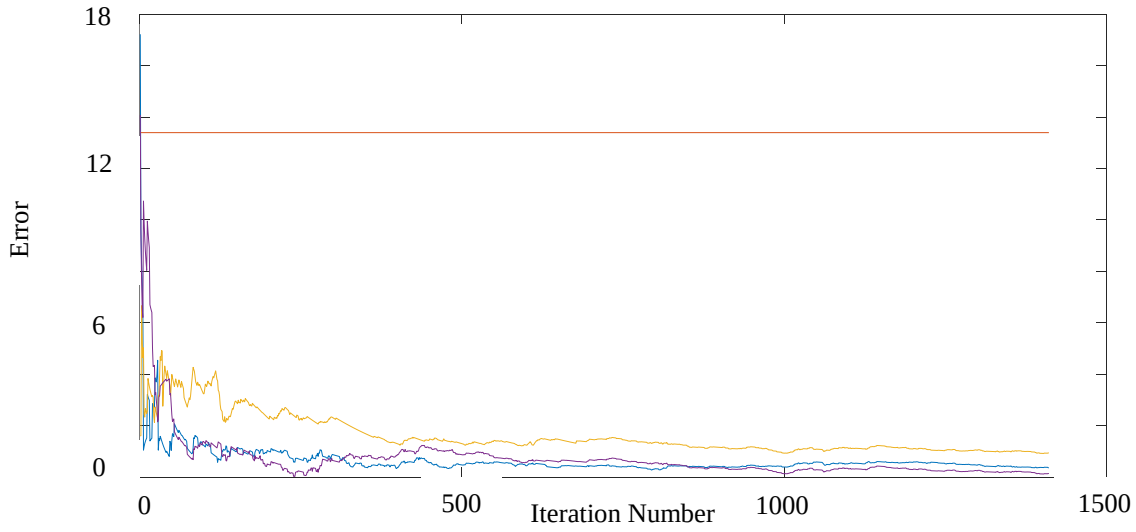

Figure 18. Learning error for automatic (blue), expert (red), large (yellow) and hybrid (purple) rule generation strategies for the Azadi station with a training to test ratio of 50/50.

For the four rule generation strategies, the average learning error at the iteration in which the computational cost is measured is summarized in Table 8. The averaging is performed over the

three training to test ratios. This helps identify the strategies that lead to better rule weights by minimizing the learning error.

Station	Rule Generation Strategy	Error (Climate)	Error (No Climate)
Azadi	Automatic	1.95	4.67
	Expert	-	13.46
	Large	2.26	5.67
	Hybrid	1.49	3.90
Shemiran	Automatic	2.36	4.96
	Expert	-	13.56
	Large	2.40	5.98
	Hybrid	1.76	3.62

Table 8. Average learning error for four rule generation strategies.

Table 8 shows that the hybrid strategy results in the lowest learning error for both stations, followed by the automatic, large and expert strategies. The reasons are as follows. Starting with the expert strategy, Figures 8 to 15 show that learning error is less sensitive to training. This can be seen in these figures where the learning error does not improve for the expert strategy with more iterations. The reason for this is the fixed number of rules in the corresponding rule list as well as the fixed rule weights. This results in a limited capacity to capture the dynamics of the input/output space. The learning error of this strategy may decrease if more rules are used. However, including more rules will result in a higher computational cost as discussed earlier. The learning error for the automatic strategy changes significantly in the initial iterations before converging. The learning error of the large strategy has a slower convergence rate which leads to a larger final error compared to the automatic strategy (Table 8). This means that optimizing the rule weights requires more iterations with the large strategy. This is because eliminating the generated rules with lower frequencies means this strategy starts with less information about the input/output space compared

to the automatic strategy. The hybrid learning strategy slightly improves the final error of the learning stage compared to the other strategies. The reason is the combining of expert rules with large rules. The expert part provides learning with a priori knowledge of the system so the framework is better initialized, and the large part provides enough sensitivity for the framework to learn. Figures 8 to 15 and Table 8 also show that regardless of the rule generation strategy, the learning error is considerably lower when climate information is included. This difference is greatest with the large strategy as the average final learning error for both stations is 2.5 times lower with climate information. Next is the hybrid strategy whose average final learning error is 2.3 times lower, followed by the automatic strategy at 2.2 times lower.

3.2.2 Computational Cost

In this section, the effect of the rule generation strategy on the computational cost of the framework is investigated. Moreover, the effect of including climate information in the fault databases on the computational cost is given. This is done by comparing the average computational cost of learning for the rule generation strategies. The averaging is done over the three training to test ratios. To show the effect of climate information, the averaging is done separately for each case. The results are summarized in Table 9.

Learning Strategy	Automatic	Expert	Large	Hybrid
Shemiran	148	186	112	177
Shemiran (No Climate)	178	347	162	230
Azadi	126	164	481	185
Azadi (No Climate)	209	498	502	497

Table 9. Learning computational cost for each of the rule generation strategies.

To evaluate the effect of the rule generation strategy on the computational cost in the multi criteria environment (station, climate inclusion and rule generation strategy), a Multi Criteria Decision Making (MCDM) approach is used. MCDM is effective for decision making based on multiple criteria [13]. For each station, the learning computational cost with each of the rule generation strategies shown in Table 8 is scored from 1 to 4 (strategies that did not converge received 0). Then, averaging is performed for each rule generation strategy to rank them [66]. The results of applying MCDM to evaluate the computational costs of learning with each of the rule generation strategies are given in Table 10. This shows that learning with the automatic rule generation strategy has the highest score and therefore, the lowest computational cost among the four strategies. This is followed by the hybrid and large strategies. Expert rule generation is the costliest of the strategies. The reason for this is the low sensitivity of this strategy as discussed in Section 3.1.1.

Strategy	Automatic	Expert	Large	Hybrid
Shemiran	2	1	4	3
Shemiran (No Climate)	3	1	4	2
Azadi	4	2	1	3
Azadi (No Climate)	4	2	1	3
Average	3.25	1.5	2.5	2.75
Rank	1	4	3	2

Table 10. MCDM scoring for the rule generation strategies based on the computational cost.

Table 10 shows that the automatic strategy has the lowest computational cost. The reason for this is the high number of rules employed in the initial rule list so learning requires fewer iterations to converge. This is followed by the hybrid strategy which benefits from both large and expert rules. Therefore, it is expected that it outperforms either of them in terms of computational cost. It may be that adding more expert rules in the expert rule list will result in the hybrid strategy having a lower cost than the automatic strategy. Table 10 also shows that including climate information decreases the learning cost. This confirms the significance of including climate information in learning as although including this information increases the ANN training space, the additional information it provides results in faster learning. Except for the expert strategy in which training appears to be insignificant to learning, this difference is most significant in the case of the hybrid strategy for the Azadi data set where the learning cost is 2.7 times lower. This is followed by the automatic strategy for the Shemiran dataset where the learning cost is 1.7 times lower. This cost is 1.5 and 1.3 lower with the large and hybrid rule generation strategy for the Shemiran station, respectively. With the automatic strategy for the Shemiran station it is 1.2 times lower, while it is unchanged with the large strategy for the Azadi station.

3.2.3 Accuracy

In order to investigate the effect of the rule generation strategies on the accuracy of the framework, the average testing accuracy for each of the strategies is evaluated. The averaging is performed over the three training to test ratios. To determine the effect of including climate information, the averaging is done with and without the climate information included in the fault database. The results are summarized in Table 11. This shows that the framework with the hybrid rule generation strategy has the highest accuracy followed by the automatic and large strategies. The framework with the expert rule generating strategy has the worst accuracy. Table 11 also shows that including climate information in the fault database increases the accuracy of the framework significantly. The respective increase in accuracy for the Azadi and Shemiran test sets when the climate information is included is

- 31.8% and 27.7% with the automatic rule generation strategy,
- 11.4% and 45.5% with the large-only rule generation strategy,
- 87.8% and 94.2% with the expert rule generation strategy, and
- 100.8% and 39.8% with the hybrid rule generation strategy.

Rule Generation Strategy	Automatic	Expert	Large	Hybrid
Shemiran	87.4%	65.3%	86.4%	86.3%
Shemiran (No Climate)	62.6%	33.6%	59.4%	67.6%
Azadi	92.4%	65.3%	88.6%	89.2%
Azadi (No Climate)	46%	34.7%	79.6%	67.7%
Average	81.6%	71.4%	81.6%	87.2%

Table 11. Accuracy of the framework for each of the rule generating strategies.

To better illustrate the performance of each of the diagnostic frameworks with a focus on the effect of climate information on their accuracy, the results are presented in Figure 19.

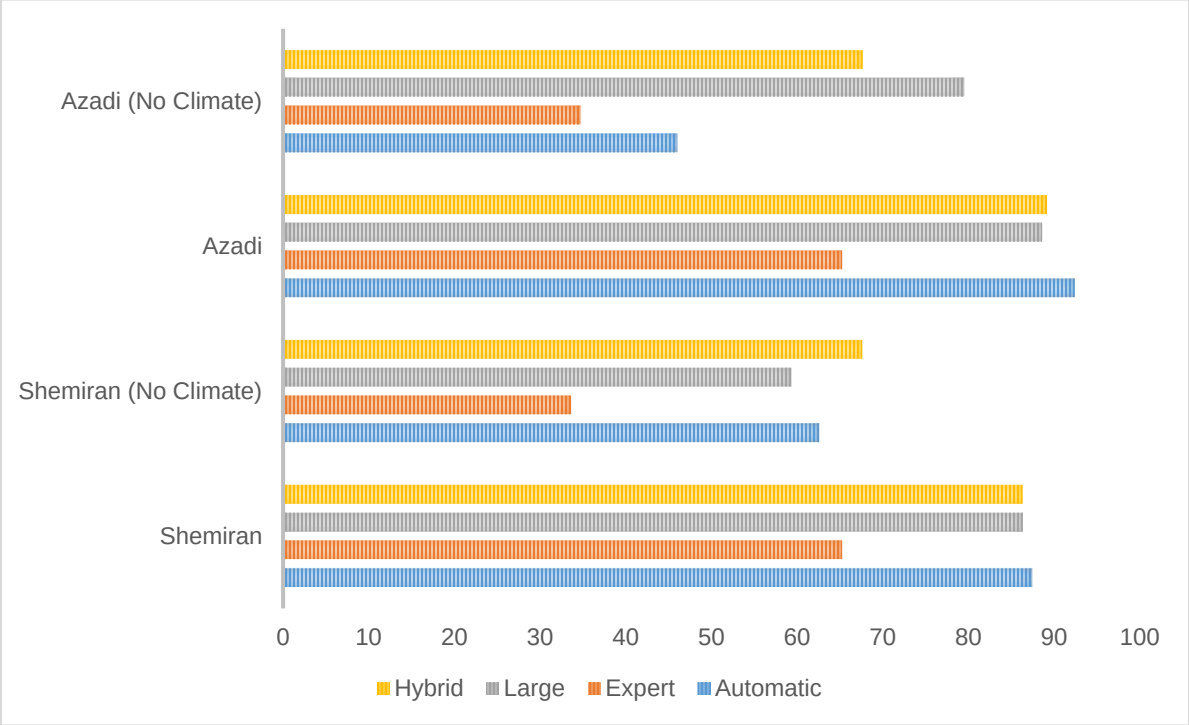

Figure 19. Testing accuracy of the framework with the rule generating strategies with and without climate information.

Chapter 4: Conclusions and Future Research

4.1 Conclusions

A fault diagnosis framework for the smart grid was designed using neuro-fuzzy reinforcement learning. Climate information was incorporated into this framework. The effects of including climate information and varying the training to test ratio on the performance of the framework were evaluated in a simulated SG environment. The results obtained show that incorporating climate information considerably improves the accuracy of the framework. Further, incorporating climate information decreases the computational cost of the framework. The results are summarized below.

- The training to test ratio is a contributing factor in the learning error, computational cost and accuracy of the diagnostic framework.
- Increasing the training/test ratio decreases the learning error for all stations and rule generation strategies. However, the hybrid strategy has a lower learning error for both stations, followed by the automatic, large and expert strategies. The reason is the extent by which these strategies can capture the input/output relations in the fault database. The hybrid strategy benefits from expert rules (which exploit prior knowledge of the input/output space) as well as large rules (which provide sufficient sensitivity to learn from the training data).
- Increasing the training/test ratio increases the computational cost. This is expected since a larger training set requires more iterations for training. However, this relation is not linear.
- There is a training set size and learning error tradeoff.

- Increasing the training/test ratio increases the accuracy of the diagnostic framework. This is consistent for both stations and all of the rule generation strategies. However, the relation between training set size and accuracy is not linear.
- The rule generation strategy is a contributing factor in the learning error, computational cost and accuracy of the diagnostic framework.
- The hybrid strategy had the lowest learning error, followed by the automatic, large and expert strategies.
- The automatic strategy had the lowest computational cost, followed by the hybrid, large and expert strategies.
- The hybrid strategy had the highest accuracy, followed by the automatic, large and expert strategies.

The following contributions are made in this thesis in addition to answering the research question posed earlier.

- Incorporating power system data for smart grid analysis.
- Providing a framework for improving the accuracy of real-time learning and diagnosis of smart grid faults by incorporating qualitative data alongside quantitative measurements.
- Employing a big data approach for fault diagnosis which is not limited to the generation, transmission, distribution or any other part of the grid.

4.2 Future Research

Implementing an intelligent framework in an SG supervisory and control system requires energy big data [1]. The high accuracy and extremely low computational cost of these frameworks, as

well as their inherent adaptability to unforeseen fault causes (intelligence), make them primary candidates for the future SG [2].

The following topics can be investigated in the future to improve the performance of the framework in SG environments.

1. Optimize the design parameters including fuzzy membership functions, rule list elements and reward function. Fuzzy membership functions can be learned directly from data and via a learning section specific to each variable. This could improve the testing accuracy as well as lower the computational cost.
2. Include rule generation in the weight optimization process. New rules could be generated from the training data set and added to the existing rule list to further capture information about the input/output relationship of the fault data set.
3. Different reward schemes can be used to increase the speed of the learning stage. The reward scheme used in this thesis is a first order algorithm. The speed can potentially be improved by using a gradient based or second order algorithm.
4. Investigate the effect of incorporating more information to construct the fault database. Among this information is load forecasts, and smart meter and generator-specific data.

References

[1] J. Hu, A. V. Vasilakos, “Energy big data analytics and security: Challenges and opportunities”, IEEE Transactions on Smart Grid, vol. 7, no. 5, pp. 2423-2436, Sep. 2016.

[2] S. M. Kaplan, F. Sissine, A. Abel, J. Wellinghoff, S. G. Kelly and J. J. Hoecker, “Smart grid: Modernizing electric power transmission and distribution; energy independence, storage and security; energy independence and security act of 2007 (EISA); improving electrical grid efficiency, communication, reliability, and resiliency; integrating new and renewable energy sources”, US Federal Government Report, Capitol.Net, 2009.

[3] S. Ntalampiras, “Fault diagnosis for smart grids in pragmatic conditions”, IEEE Transactions on Smart Grid, vol. 9, no. 3, pp. 1964-1971, May 2018.

[4] R. Isermann, “Fault-diagnosis systems: An introduction from fault detection to fault tolerance”, Springer-Verlag, 2006.

[5] P. Gopakumar, “Adaptive fault identification and classification methodology for smart power grids using synchronous phasor angle measurements”, IET Generation, Transmission & Distribution, vol. 9, no. 2, pp. 133-145, Feb. 2015.

[6] Y. Yu, “Electric Power System Dynamics”, New York: Academic Press, 1983.

[7] D. Bailey, E. Wright, “Practical SCADA for Industry”, Elsevier Science and Technology, 2003.

[8] V. H. Ferreira, R. Zanghi, M. Z. Fortes, “A survey on intelligent system application to fault diagnosis in electric power system transmission lines”, Electric Power System Research, vol. 136, pp. 135-153, Jul. 2016.

[9] S. D. J. McArthur, E. M. Davidson, J. A. Hossack, J. R. McDonald, “Automating power system fault diagnosis through multi-agent system technology”, Hawaii International Conference on System Sciences, pp. 8, Jan. 2004.

[10] C. C. Liu, T. Dillon, “State-of-the-art of expert system applications to power systems”, International Journal of Electrical Power Energy Systems, vol. 14, pp. 86-96, Apr-Jun 1992.

[11] Y. Sekine, Y. Akimoto, M. Kunugi, C. Fukui, “Fault diagnosis of power systems”, Proceedings of the IEEE, vol. 80, no. 5, pp. 673-683, May 1992.

[12] A. Girgis, M. B. Johns, “A hybrid expert system for faulted section identification, fault type classification and selection of fault location algorithms”, IEEE Transaction on Power Delivery, vol. 4, no. 2, pp. 987-995, Apr. 1989.

[13] V. F. Yu, K. Hu, “An integrated fuzzy multi-criteria approach for the performance evaluation of multiple manufacturing plants”, Journal of Computers & Industrial Engineering, vol. 58, no. 2, pp. 269-277, Mar. 2010.

[14] A. Hertz, P. Fauquembergue, “Fault diagnosis at substation based on sequential event recorders”, Proceedings of the IEEE, vol. 80, no. 5, pp. 684-688, May 1992.

[15] J. R. McDonald, G. M. Burt, D. J. Young, “Alarm processing and fault diagnosis using knowledge based systems for transmission and distribution network control”, IEEE Transactions on Power Systems, vol. 7, no. 3, pp. 1292-1298, Aug. 1992.

[16] M. Kezunovic, P. Spasojevic, C. W. Fromen, D. R. Sevcik, “An expert system for transmission substation event analysis”, IEEE Transactions on Power Delivery, vol. 8, no. 4, pp. 1942-1949, Oct. 1993.

[17] Z. Yongli, Y. Yang, B. Hogg, W. Zhang, S. Gao, “An expert system for power systems fault analysis”, IEEE Transactions on Power Systems, vol. 9, no. 1, Feb. 1994.

[18] M. E. Vazquez, L. Oscar, M. Chacon, J. Hector, F. Altuve, “A knowledge-based system for on-line diagnosis for power system fault allocation”, IEEE International Conference on Systems, Man and Cybernetics, pp. 1148-1153, Oct. 1994.

[19] H. Miao, M. Sforna, C. C. Liu, “A new logic-based alarm analyzer for on-line operational environment”, IEEE Transactions on Power Systems, vol. 11, no. 3, pp. 1600-1606, Aug. 1996.

[20] C. Chang, S. G. Kerk, “Fault signal filtering for improving fault section estimation”, IEEE Power Engineering Society Winter Meeting, pp. 2545-2550, Jan. 2000.

[21] J. Jung, C. C. Liu, M. Hong, M. Gallanti, G. Tornielli, “Multiple hypothesis and their credibility in on-line fault diagnosis”, IEEE Transactions on Power Delivery, vol. 26, no. 2, pp. 225-230, Apr. 2001.

[22] W. H. Chen, S. H. Tsai, H. I. Lin, “Fault section estimation for power networks using logic cause-effect models”, IEEE Transactions on Power Delivery, vol. 22, no. 3, pp. 963-971, Apr. 2011.

[23] Y. Zhu, H. Limin, J. Lu, “Bayesian network-based approach for power systems fault diagnosis”, IEEE Transactions on Power Delivery, vol. 21, no. 2, pp. 634-639, Apr. 2006.

[24] A. T. Ghainani, A. A. M. Zin, N. M. Ismail, “Fuzzy timing Petri net for fault diagnosis in power systems”, Mathematical Problems in Engineering, Jun. 2012.

[25] J. Sun, S. Y. Qin, Y. H. Song, “Fault diagnosis for electric power systems based on fuzzy Petri nets”, IEEE Transactions on Power Systems, vol. 19, no. 4, pp. 2053-2059, Nov. 2004.

[26] X. Lou, M. Kezunovic, “Implementing fuzzy reasoning Petri-nets for fault section estimation”, IEEE Transactions on Power Delivery, vol. 23, no. 2, pp. 676-685, Apr. 2008.

[27] V. Cherkassky, F. M. Mulier, “Learning from Data: Concepts, Theory, and Methods”, 2nd Ed., IEEE Wiley Press, 2007.

[28] S. O. Haykin, “Neural Networks and Learning Machines”, 3rd Ed. Patience Hall, 2008.

[29] C. E. Pereira, L. C. Zanetta Jr. “An optimization approach for fault location in transmission lines using one terminal data”, International Journal of Electrical Power Energy Systems, vol. 29, no. 4, pp. 290-296, May 2007.

[30] V. Miranda, D. Srinivasan, L. Proenca, “Evolutionary computation in power systems”, International Journal of Electrical Power Energy Systems, vol. 20, no. 2, pp. 89-98, Feb. 1998.

[31] L. Wei, W. Guo, F. Wen, G. Ledwich, Z. Liao, J. Xin, “Waveform matching approach for fault diagnosis of a high voltage transmission line employing harmony search algorithm”, IET Generation, Transmission & Distribution, vol. 4, no. 7, pp. 801-809, Jun. 2010.

[32] S. Luo, M. Kezunovic, D. R. Sevick, “Locating faults in the transmission network using sparse field measurements, simulation data and genetic algorithm”, Electric Power System Research, vol. 71, no. 2, pp. 169-177, Oct. 2004.

[33] Y.-X. Wu X.-n. Lin, S.-h. Miao, P. Liu, D.-q. Wang, D.-b. Chen, “Application of family eugenics based evolution algorithms to electric power system fault section estimation”, IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific, pp. 1-55, Aug. 2005.

[34] W. Guo, F. Wen, G. Ledwich, Z. Liao, X. He, J. Huang, “A new analytic approach for power system fault diagnosis employing the temporal information of alarm messages”, International Journal of Electrical Power Energy Systems, vol. 43, no. 1, pp. 1204-1212, Dec. 2012.

[35] B. Hu, J. She, R. Yokoyama, “Hierarchical fault diagnosis for power systems based on equivalent-input-disturbance approach”, IEEE Transactions on Industrial Electronics, vol. 60, no. 8, pp. 3529-3538, Aug. 2013.

[36] A. Ferrero, S. Sangiovani, E. Zapitelli, “A fuzzy-set approach to fault-type identification in digital relaying”, IEEE Transactions on Power Delivery, vol. 10, no. 1, pp. 169-175, Jan. 1995.

[37] H. J. Cho, J. K. Park, “An expert system for fault section diagnosis of power systems using fuzzy relations”, IEEE Transactions on Power Systems, vol. 12, no. 1, pp. 342-348, Feb. 1997.

[38] L. Zadeh, “Fuzzy sets”, Information and Control, vol. 8, no. 3, pp. 338-353, Jun. 1965.

[39] B. Das, J. Reddy, “Fuzzy-logic-based fault classification scheme for digital distance protection”, IEEE Transactions on Power Delivery, vol. 20, no. 2, pp. 609-616, 2005.

[40] H. Monsef, A. M. Ranjbar, S. Jadid, “Fuzzy rule-based expert system for the power system fault diagnosis”, IEE Proceedings – Generation, Transmission & Distribution, vol. 144, no. 2, pp. 186-192, Mar. 1997.

[41] H. J. Lee, D. Y. Park, B. S. Ahn, Y. M. Park, J. K. Park, S. Venkata, “A fuzzy expert system for the integrated fault diagnosis”, IEEE Transactions on Power Delivery, vol. 15, no. 2, pp. 833-838, Apr. 2000.

[42] E. Meza, J. C. de Souza, M. T. Schilling, M. do Coutto Filho, “Exploring fuzzy relations for alarm processing and fault location in electrical power systems”, IEEE Porto Power Techology Conference, pp. 6, Sep. 2001.

[43] H. C. Chin, “Fault diagnosis of power system using fuzzy logic”, IEEE Transactions on Power Systems, vol. 18, no. 1, pp. 245-250, Feb. 2003.

[44] H. Singh, “ANFIS: Adaptive neuro-fuzzy inference system – A survey”, International Journal of Computer Applications, vol. 123, no. 13, pp. 32-38, Aug. 2015.

[45] D. Karaboga, E. Kaya, “Adaptive network-based fuzzy inference system (ANFIS) training approaches: A comprehensive survey”, The Artificial Intelligence Review, pp. 1-31, Jan. 2018.

[46] V. H. Ferreira, A. P. A. da Silva, “Toward estimating autonomous neural network-based electric load forecasters”, IEEE Transactions on Power Systems, vol. 22, no. 4, pp. 1554-1562, Nov. 2007.

[47] A. P. A da Silva, V. H. Ferreira, R. M. Velasquez, “Input space to neural network based load forecasters”, International Journal of Forecast, vol. 24, No. 4, pp. 616-629., Dec. 2008.

[48] V. H. Ferreira, A. P. A da Silva, “Chaos theory applied to input space representation of autonomous neural network-based short term load forecasting models”, SBA Control & Automation, vol. 22, no. 6, Nov. 2011.

[49] V. H. Ferreira, A. P. A da Silva, “Autonomous kernel based models for short-term load forecasting”, Journal of Energy Power Engineering, vol. 6, pp. 1984-1993, May 2012.

[50] V. H. Ferreira, A. R. Aoki, S. M. de Rocco, “Demand forecasting for control of the use of transmission system for electric distribution utilities”, International Conference on Intelligent Systems Applications to Power Systems, Dec 2009.

[51] A. E. Lazzaretti, V. H. Ferreira, H. V. Neto, R. J. Riella, J. S. Omori, “Classification of events in distribution networks using autonomous neural models”, International Conference in Intelligent System Applications to Power Systems”, vol. 9, no. 1, pp. 1-6, Nov. 2009.

[52] A. E. Lazzaretti, V. H. Ferreira, H. V. Neto, R. J. Riella, J. S. Omori, “Autonomous neural models for the classification of events in power distribution networks”, Journal of Control and Automation of Electrical Systems, vol. 24, no. 5, pp. 612-622, Oct. 2013.

[53] A. E. Lazzaretti, V. H. Ferreira, H. V. Neto, “An accurate approach for automatic segmentation of power distribution voltage waveforms”, IEEE Transaction on Power Delivery, vol. 30, no. 3, pp. 1611-1619, Jun. 2015.

[54] W. M. Lin, C. D. Yang, J. H. Lin, M. T. Tsay, “A fault classification method by RBF neural network with OLS learning procedure”, IEEE Transactions on Power Delivery, vol. 16, no. 4, pp. 471-477, Oct. 2001.

[55] T. Dalstein, B. Kulicke, “Neural network approach to fault classification for high speed protective relaying”, IEEE Transaction on Power Delivery, vol. 10, no. 2, pp. 1002-1011, Apr. 1995.

[56] U. B. Parikh, B. Das, R. Maheshwari, “Fault classification technique for series compensated transmission line using support vector machine”, International Journal of Electrical Power Energy Systems, vol. 32, no. 10, pp. 629-636, Jul. 2010.

[57] J. Morales, E. Orduna, C. Rehtanz, “Classification of lightning stroke on transmission line using multi-resolution analysis and machine learning”, International Journal of Electrical Power Energy Systems, vol. 58, pp. 19-31, Jun. 2014.

[58] R. J. Campbell, “Weather-related power outages and electric resiliency”, Congressional Research Service, Library of Congress, Washington DC, USA, Tech. Rep. R42696, Aug. 2012.

[59] K. L. Anya, M.G. Pollitt, “Does weather have an impact on electricity distribution efficiency? Evidence from South America”, [Online]. Available: https://doi.org/10.17863/CAM.5857

[60] J. M. Gere, B. J. Goodno, “Mechanics of Materials, 8th ed.”, Cengage Learning, 2009.

[61] D.G. Fink, H. W. Beaty, “Standard Handbook for Electrical Engineers, 11th ed.”, McGraw-Hill, New York, 1978.

[62] C. I. Hubert, “Preventive Maintenance of Electrical Equipment”, McGraw-Hill, New York, 1969.

[63] S. D. Guekima, R. A. Davidson, H. Liu, “Statistical models of the effects of tree trimming on power system outages”, IEEE Transaction on Power Delivery, vol. 21, no. 3, pp. 1549-1557, Jul. 2006.

[64] K. Shimojima, T. Fukuda, Y. Hasegawa, “Self-tuning fuzzy modeling with adaptive membership function, rules, and hierarchical structure based on genetic algorithm”, Fuzzy Sets and Systems, vol. 71, pp. 295-309, May 1995.

[65] W. V. Leekwijck, E. E. Kerre, “Defuzzification: Criteria and classification”, Fuzzy Sets and Systems, vol. 108, no. 2, pp. 159-178, Dec. 1999.

[66] Kwangsun, Y. C. Hwang, “Multiple Attribute Decision Making” Springer-Verlag, Berlin, 1981.

[67] S. Raschka, V. Mirjalili, “Python Machine Learning”, Packt Publishing Ltd, 2015.