Intelligent Instability Detection for Islanding Prediction

Zahra Pakdel

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Electrical Engineering

Virgilio A. Centeno
Arun G. Phadke
Jaime De La Ree Lopez
Amos Lynn Abbott
Werner E. Kohler

May 6, 2011
Blacksburg, Virginia U.S.A

Keywords: wide area measurements, neural network, data mining, decision trees.

Intelligent Instability Detection for Islanding Prediction

Zahra Pakdel

Abstract

The goal of the proposed procedure in this dissertation is the implementation of phasor measurement unit (PMU) based instability detection for islanding prediction procedures using decision tree and neural network modeling. The islanding in the power system define as a separation of the coherent group of generators from the rest of the system due to contingencies, in the case that all generators are coherent together after introducing a fault, it is called stable or non-islanding. The main philosophy of islanding detection in the proposed methodology is to use decision trees and neural network data mining algorithms, performed off-line, to determine the PMU locations, detection parameters, and their triggering values for islanding detection. With the information obtained from accurate system models PMUs can be used online to predict system islanding with high reliability.

The proposed approach is proved using a 4000 bus model of the California system. Before data mining was performed, a large number of islanding and non-islanding cases

were created for the California model. PMUs data collection was simulated by collecting the voltage and current information in all 500 kV nodes in the system. More than 3000 cases were collected and classified by visual inspection as islanding and non-islanding cases. The proposed neural network and decision tree procedures captured the knowledge for the correct determination of system islanding with a small number of PMUs.

Acknowledgements

I would like to thank the National Science Foundation for their fellowship award and their support to make my dream come true.

I would like to acknowledge and thank my committee members, Dr. Arun Phadke, Dr. Jaime De La Ree, Dr. Amos Lynn Abbott, and Dr. Werner E. Kohler, for their assistance and encouragement throughout this research. In particular, I would like to thank my advisor, Dr. Virgilio Centeno, for his tremendous help and direction. Indeed, without his support, this research would not have been possible. You pushed me to excel during this process, and you reminded me that the reward was well worth the effort. Thank you.

I would like to extend my sincerest appreciation to my daughter Niyousha. It is her great support that gives me motivation to continue. Niyousha, being your mother is the greatest reward in my life.

Table of Contents

Abstract ii
Acknowledgements iv
List of Figures. vii
List of Tables viii
1. Introduction 1
  1.1. Islanding detection 2
  1.2. Synchronous Generator Concept 2
    1.2.1. Stand Alone Synchronous Generator 4
    1.2.2. Synchronous Generators in Parallel 4
  1.3. Chapters overview 7
2. Literature Review 8
  2.1. Phasor Measurements Applications 10
    2.1.1. Transient Stability and Control 11
    2.1.2. Phasor Measurements and Protection 13
  2.2. Islanding Detection in Power System 14
  2.3. Decision Tree Modeling and Power System 17
  2.4. Neural Networks and Power Systems 18
3. Methodology 25
  3.1. Islanding Detection 26
  3.2. Generating of Islanding and non-Islanding Cases 27
  3.3. Decision Trees 29
    3.3.1. Input Selection 31
    3.3.2. Tree Training Methodology 32
      3.3.2.1. Training Decision Trees for Classification 34
      3.3.2.2. Cost Function 35
    3.3.3. Assessment of Model Performance 37
      3.3.3.1. Cross Validation 37
      3.3.3.2. chi-squared Distribution 41
    3.3.4. Decision optimization 42
      3.3.4.1. The factors affecting behavior of the algorithm 42

3.4. Predictive Modeling: Neural Network 43
3.4.1. Feedforward Neural Network Model 48
3.4.2. Feedback Neural Network Model 49
3.5. Neural Network Modeling 50
3.5.1. Neural Network Training 51
3.5.2. Number of hidden neurons 52
3.5.3. Complexity Optimization 53
4. Simulation Results 54
4.1. Heavy Winter Model: Decision Tree 58
4.1.1. The Optimal Model for Heavy Winter 66
4.1.2. Performance Assessment: Heavy Winter Model 68
4.1.3. Heavy Summer Model: Decision Tree 73
4.1.4. Performance Assessment: Heavy Summer Model 77
4.2. Neural Network Simulation 80
4.2.1. Neural Network Training Results 82
4.2.2. Neural Network Performance Improvement 84
4.2.3. Neural Network Optimal Model 86
5. Conclusions 88
5.1. Contributions 88
5.2. Future Work 90
Appendix A 92
Appendix B 102
Literature Cited 104

LIST OF FIGURES

Figure **Page**

Figure 1: Simple Classification Demonstration 39
Figure 2: Decision Tree Example 40
Figure 3: Simple Neural Network Diagram with Transfer Function and Weight 44
Figure 4. Summary of Simulation Process 55
Figure 5: Generator Rotor Angles of case 1065, Stable Case 59
Figure 6: Generator Rotor Angles of case 151, Unstable Case 60
Figure 7. Decision Tree Result for HW with more Branches 62
Figure 8. Misclassification Results for Higher Number of Branches Tree 64
Figure 9. Heavy Winter Decision Tree Result with more Branches and Higher Misclassification Rate 65
Figure 10. Misclassification Results for Higher Branches Tree 66
Figure 11: Decision Tree, Heavy Winter 67
Figure 12: Misclassification Rate for Optimal Model for Heavy Winter 68
Figure 13: Data Validation for out of Sample 69
Figure 14: Misclassification Rate for HW Second Location 71
Figure 15: Decision Tree with 6% variation in operating condition 72
Figure 16: Misclassification Rate for 6% Variation in out of Sample Data 73
Figure 17: Islanding Case by Bus Fault for Heavy Summer Model 74
Figure 18: Decision Tree, Heavy Summer 75
Figure 19: Data Validation for Heavy Summer Model 75
Figure 20: Simulation Result for Heavy Summer Model with 1044 Cases 76
Figure 21: Misclassification Rate for the Heavy Summer Model with 1044 Cases 77
Figure 22: The Training Data and Validation Misclassification Rate 78
Figure 23: Simple Block Diagram for Neural Network Model 81

LIST OF TABLES

Table **Page**

Table 1: Sample collected cases data for heavy summer model 61
Table 2: Secondary Split to Increase Reliability 70
Table 3: Splitting Rules 71
Table 4: Logworth Values for Diablo and Midway 79
Table 5: Splitting Rule for Diablo 79
Table 6: Weight Model for 500 KV Bus 1 through 31 84
Table 7: Misclassification Rate for Different Layers 85
Table 8: Number of Neurons for each Hidden Layer 86
Table 9: PMU locations based on the Neural Network Results 87
Table 10: Weight Model for 500 KV Buses 1 through 31 102
Table 11: Weight Model for 500 KV Buses 103

1. INTRODUCTION

The growth of system loading requires utilities to operate close to the maximum capabilities of their transmission systems, increasing the possibility of unstable swings among generators. When severe faults occur in a system, loss of synchronism between groups of generators may take place after single or multiple swings, and if appropriate action is not taken on time, the loss of synchronism may cascade to the rest of the system and cause large blackouts.

Usually, islanding in the power system occurs after some transmission lines are tripped by local relays, which results in electrically disconnected areas with different degrees of unbalance between load and generation. Studies of past blackouts [1] show that incorrect operation of local relays contribute to cascading events that lead to areas with excessive electrical unbalance and if proper system islanding was performed in time, many blackouts could have been avoided.

Phasor measurement units have become an essential part of the power system monitoring, improving real-time observation capabilities. These devices can capture real time data to be used by an offline-trained model to make decisions for islanding detection. The methodologies proposed in this work contribute to develop off line-trained models for islanding detection in power system based on data mining approaches. More specifically, the proposed methodology is applied to train a decision tree model and a neural network model, and subsequently compares the performance of the two models.

1.1. Islanding detection

Intelligent islanding is a critical way of preventing large disturbances from propagating into the rest of the system and causing a severe system breakup. The initial disturbances may be caused by a loss of a major transmission line or loss of major generation in the system. In such applications, it is essential to detect islanding conditions quickly to increase the possibility of implementing intelligent islanding control procedures. If intelligent islanding is performed quick it is easier to maintain suitable island frequency and voltage, both transiently and in post disturbance steady state, allowing faster system recovery.

A power system dynamic study is necessary to evaluate and determine islanding conditions. The cases for this work are collected by dynamic simulation for various reasonable conditions in a highly accurate model of the California system. In this model large industrial motors are modeled in detail, and smaller motor loads are representing by equivalent lumped motors. The turbine-governors, and generators with exciters, are also modeled in detail.

1.2. Synchronous Generator Concept

Power plants convert potential, thermal and other forms of energy into electrical energy. There are three main components in a conventional power plant: a turbine, a generator, and a transformer. The turbine provides the mechanical energy to the generator. The generator converts this mechanical energy into electrical energy and a transformer raises the voltage levels to allow efficient transportation of the energy. This section describes some of the basic concepts of synchronous generators for stand-alone and parallel operation. Knowledge of these concepts helps understand the islanding operation.

The basic principle of power generation is that a magnetic field changing across a stationary conductor induces a voltage in the conductor. In simple terms generators are formed by an electrical circuit and a magnetic circuit interacting with each other. In today's generators, the electrical circuit is stationary and it is made up of armature windings embedded in stationary stator core. When the generator is running, voltage is induced in these windings by the rotating magnetic fields in the rotor. When a load is connected, current flows through the stator windings. The frequency of the electricity produced by a synchronous generator is related to the speed of the rotor and the number of poles in the stator as shown in equation (1) [2].

$$f_e = \frac{n_m P}{120} \tag{1}$$

Where f_e is electric frequency (Hz), n_m is rotor speed (Rev/Min), and P is number of poles. The output capacity of the generator is expressed in MVA, or Mega-Volt-Amperes. Typical values for the rating of a generator range from about 2 MVA, for a small turbine-generator intended for Navy applications, to 1,560 MVA for large nuclear power stations. The range are for most of generators connected to the grid the same, but some of generators at the distribution level like powering a home may be as low as 3 KVA.

1.2.1. Stand Alone Synchronous Generator

In the case of a single synchronous generator connected to a load, any change in the load causes a change in the terminal voltage. The details and reasons are discussed by Chapman [2], but as a summary an increase of inductive reactive power load decrease the terminal voltage. On the other hand, an increase of capacitive reactive power load causes the terminal voltage to increase. The change in the real power load causes the terminal voltage to change as well, which is complicated and highly dependent on the generators, load distribution, and other factors in the power network.

Generally, in case of islanding, the machines are operated as speed control or isochronous speed control to maintain a constant speed by changing the fuel. This type of control provides automatic reset (adds the time integral of speed error to running speed error) to the running speed, as determined by the speed set point, following load changes during isolated operation. Therefore, isochronous speed control gradually eliminates all speed error to maintain a constant steady state speed. Because of this action isochronous speed control is sometimes called a speed corrector.

1.2.2. Synchronous Generators in Parallel

All generators in the grid power systems are connected in parallel. When a new generator is added to the system there is a precise process to connect it to the grid, which is called synchronization. Basically in order to connect a generator to the grid, it needs to have the same electrical terminal voltage, phase, and phase rotation as the grid. The frequency of the generator that is going to be synchronized to the grid needs to be slightly higher than the frequency of the grid in order to not act as a motor [2].

The frequency of the generator changes as the load increases as shown in equation (2) as Speed Droop (*SD*) equation.

$$SD = \frac{n_{nl} - n_{fl}}{n_{fl}} * 100\ \% \tag{2}$$

The decrease in frequency from no load to full load in most generators is between 2 to 4% [2].

Gas turbine generator shaft speed and power output are regulated together by a droop governor response to help maintain the frequency constant with the grid. Generally, the droop governor response increases power output for low generator frequency operation and decreases power output for high generator frequency operation.

A drop in electrical grid frequency indicates that the power generation capability of the grid is less than the load demand on the grid. Conversely, if the electrical grid frequency is above nominal, the power generation capability supplied to the grid is greater than the load demanded. The governor response attempts to correct these situations by changing the power output of a turbine inversely proportional to the electrical grid frequency departure from nominal. If the grid frequency drops below rated frequency, the turbine will be commanded to increase its power output. If the grid frequency increases above the rated frequency, the turbine will be commanded to reduce its power output.

Chapman [2] describes the relationship between real power and frequency as shown in equation (3):

$$P = s_p (f_{nl} - f_{sys}) \tag{3}$$

Where P is real power coming out of generator, f_{nl} is generator frequency at no load, f_{sys} is frequency of the system, and S_p is slope of the curve (MW/Hz).

As the load real power increases the frequency of the system decreases, and as the load real power decreases the frequency of the system increases. A severe fault can cause a large variation in power flow and bus bar voltage, which in turn can cause the outage of generation units or transmission lines, and losing the balance between the load and power generation. These disturbances may propagate in an uncontrolled fashion and cause blackout in the power system. For example, loss of synchronism between groups of generators may take place after single or multiple swings and lead to instability of the system. Islanding detection, enabled by PMU measurements and data mining models is the first step in preventing a severe fault from propagating into the rest of the system and causing a system breakup.

1.3. Chapters overview

The next chapter discusses the literature reviews of islanding in power system, decision tree modeling and power system, neural network modeling and power system and some of the existing applications for phasor measurements. It also contains a comprehensive review of controlled islanding and neural network modeling for power systems.

Chapter 3 describes the proposed methodology for islanding detection with both decision trees and neural network modeling. The methodology is based on Wide Area Measurements (WAMs) and data mining. The models are trained offline and use real data online for predictions. The concepts of data mining and its modeling techniques as decision trees and neural network are thoroughly described in this Chapter.

Chapter 4 shows the simulation results for both decision trees and neural network models. The simulation results are based on the two models from the California system: heavy winter model and heavy summer model. This chapter also shows the results for decision trees and neural network model comparisons. Chapter 5 presents the conclusion and provides discussion for future work.

2. LITERATURE REVIEW

Current literature classifies islands in the power system as intentional or unintentional [3],[4]. In the intentional islanding case, the power system is designed to deal with the situation and survive since the case has been planned. Unintentional islanding means that there was no design and islanding occurs unplanned and may lead additional disturbances in the power system.

Self-adaptive islanding, also known as controlled islanding [3], would occur when the whole transmission system is actively divided into several islands. After controlled islanding, the whole power system is under intentional islanding operation. In each island the load and generation remains in balance usually by shedding the extra load in the island [5]. One of the problems for this method is implementing the islands in a reasonable time. The literature reviews suggest that this has not received sufficient study in the past.

Any transient disturbances that cause instability in the system may lead to uncontrolled islanding in the power system. For example, if more power is generated than required by the load, the grid frequency increases; on the other hand, if the load demand is larger than the generated power, the grid frequency is decreased. Moreover, the voltage stability of the gird is important to maintain a stable grid. The voltage stability is defined as the voltage of the system maintain as acceptable value [6]. The changes in the magnetization current in the generator affect the voltage. This affects the reactive output and hence the voltage level. More details may be reviewed in [6].

An islanding detection system has to distinguish between islanding and other events in the power system. It is vital for such a system to be reliable. A reliable system is defined by its dependability and security. Dependable islanding detection design requires a methodology that identifies all islanding in the system. A secure design may be defined as an islanding detection procedure that detects only to islanding event and ignores any other type of disturbance.

There are many different methods used for islanding detection. All have their own positive and negative sides. These methods have traditionally been divided into two subgroups: passive and active methods. Passive islanding detection methods use quantities such as voltage or frequency to monitor system changes. One of the advantages of passive methods is that they do not contribute to power quality issues such as voltage dips. Voltage relays measure the voltage magnitude and respond to both under- and over-voltage situations. One of the disadvantages voltage controlled passive methods is that the generating unit does not trip unless the voltage is outside predefined limitations. Frequency based passive methods are based on the knowledge that the speed of a synchronous generator is proportional to the average frequency. For these methods a relay makes its decision based on the frequency deviation from nominal. If the frequency increases or decreases from a predetermined value for a certain time, then under or over frequency relays take action based on the situation [7].

Active islanding detection methods are also based on the voltage or the frequency. Reference [8] shows a method based on the supervisory control and data acquisition (SCADA) system. The proposed system collects the information on the states of the circuit breakers in the grid and shows that the information in the SCADA system is sufficient to determine if islanding has occurred in the power system. The investigation performed in [8] shows that these methods are very slow, and have been mostly used for distributed resources.

2.1. Phasor Measurements Applications

Phasor measurement units are quickly becoming a vital part of the power system and may greatly help to prevent blackouts. This rapid increase has been partially motivated by the final reports on the UCTE split of 2006 [9], and U.S/Canada blackout of 2003 [1], which indicate that the lack of wide area awareness was a critical cause of the blackouts. These reports recommend using PMUs to improve the real-time system observation.

PMUs devices provide synchronized phasor measurements, thanks to modern developments in computer-based measurements and the synchronization service available from GPS satellite transmissions. PMUs are also having a large impact on state estimation, security monitoring, predictive controls, and relaying. These measurements enhance the power system protection, control and monitoring [10], [11], [12], [13], [14], [15], [16], [17].

2.1.1. Transient Stability and Control

Any transient disturbances that cause instability in the system are defined as transient instability. The initial disturbances can be a loss of major transmission lines or a loss of major load or generation in the system. The instability is detected by monitoring one or more of the system quantities like rate of change of power, bus voltage angle, or sudden change in power flow. After instability is detected in the system, controlled Islanding is recommended in the predetermined areas before cascading outages occur.

PMUs have great advantages in transient stability monitoring. Reference [18] develops a placement scheme based on the coherent machine clusters as identified by the techniques described in the publication. The results are the development of procedures to identify weak branches in the system when reduced to the internal generator nodes. When a power system has large disturbances, groups of generators will have coherent motion. Using coherent techniques for PMUs placement have some advantages like the use of a single PMU for each cluster.

Two definitions of voltage coherency are proposed by Phadke and Begovic in [19]. These definitions are based on the different properties, and two different methods for identifying coherent regions are presented. They demonstrated that a system could be divided into a few voltage control areas. When the loading increases beyond a threshold value and the limit of reactive power generation are reached, this may lead to voltage collapse. The control of voltage level is obtained by controlling the absorption, production, and flow of reactive power at all levels in the system. The devices used to control reactive power are classified as regulating transformer, line reactance compensator, and sources and sinks of reactive power.

The transient stability phenomenon has been discussed by Kundur [6] and can be followed by representing the generators electrical output by equation (4):

$$P_e = \frac{E_G E_M}{X_T} \sin \partial = P_{\max} \sin \partial \tag{4}$$

Where

P_e, electrical power

E_G, voltage at generator terminal

E_M, voltage at machine terminal

X_T, impedance between two machines

∂, angular separation between the rotors of two machines

P_{max}, maximum electrical power output

$$P_{\max} = \frac{E_G E_M}{X_T} \tag{5}$$

In equation (5) the stator resistance has been neglected. The equation of motion or swing equation is:

$$\frac{2H}{\omega_{\circ}}\frac{d^2\partial}{dt^2} = P_m - P_{\max}\sin\partial \tag{6}$$

Where

t= time in second

H = inertia constant

P_m= mechanical power input (7)

$P_{\max}$ = maximum electrical power output

When the mechanical power increases the amount of electrical power, it generates an accelerating torque, which causes the rotor angle acceleration.

2.1.2. Phasor Measurements and Protection

One of the innovations in a modern protection system is a system integrity protection schemes. These schemes identify system status and actions to prevent major system outages [10]. According to the author recent events have shown that phasor measurement units are no longer optional but they are essential for maintaining the reliability of the power system, and they are needed to predict major events in the gird. On September 1, 2008 hurricane Gustav, a category 2 storm hit Cocodrie, Louisiana. This caused outages of 354 substation and 241 transmission lines [10].

This hurricane destroyed or damaged 4349 transformers, and it was the second most destructive storm in 95 years in the history of the Entergy system. More detailed information in regards to island creation and precautionary steps that Entergy took in order to protect their system may be reviewed at [20]. A PMU was located inside the island, Waterford, during the event. This PMU was used as the temporary reference frequency for the Entergy system during the event. This event proved that the location of this PMU was essential in detecting and managing the island. The authors concluded that PMUs taking measurements at 30 samples per second provide a great advantage that was not possible with other system [20].

2.2. Islanding Detection in Power System

The main cause of major blackouts is cascading failures. Tripping a transmission line or a generating unit can be the beginning of a cascading sequence that may lead to blackouts and/or uncontrollable islanding. For economic reasons, most power systems operate close to their stability limits. A variety of emergency controls are available for power system protection- generator tripping, load shedding, excitations control and system islanding- but so far system islanding is the last line of defense.

Uncontrolled islanding is the process by which an interconnected power system separates into unplanned islands as a result of a severe disturbance. With the formation of uncontrolled islanding, power system recovery is more complex and more time consuming. It is desirable detect islanding conditions as soon as possible to control the formation of islands, by containing the impact of a disturbance to a particular region of the network "as early as possible". The goal is to preserve stable areas of the power system. In [4] an algorithm based on dynamic performance and slow coherency of the islands is explored. The authors also consider the type and location of the extreme faults. The developed algorithm is applied to IEEE 118 bus system. The results show the evaluation between the proposed islanding and an islanding scheme based on the physical allotment of the synchronous generator in the power system. It is extracted from their work that controlled islanding preserves more stable areas, and optimizes the load shedding. Also, the results of this work show that controlled islanding not only helps to preserve more stable areas in the system; it also optimizes the power requirement for generation and load in each island. They system voltage stay between limits and line flow match the loading limit [4].

Reference [21] discusses decision tree controlled islanding for preventing cascading events in the power system. Their work uses PMUs and presents a decision tress scheme to decide the time of control islanding. From their analysis, the accurately trained decision trees are able to give a correct prediction. The proposed scheme is tested on the Entergy system. Their work is based on training one decision tree for each critical contingency lead to blackout in the system. The transient simulations are performed for N-2 contingencies. It is concluded that controlled islanding method can help to make the power system stable faster after occurring major faults. Also control islanding prevent to tripping load more than the amount that is required for stability of the power system. Training one decision tree for each contingency seems inefficient and fails to cover unexpected contingencies.

Reference [22] is a major study on forcing islands to form at predetermined system frequency before its natural formation. This is one good example of many recent discussions on the implementation of under frequency detection controlled islanding in the world.

One of the major drawbacks of waiting for formation of natural islands for partial system preservation to aid in overall system recovery is the severity of the under voltages experienced in the island formed. In some study cases under voltages resulted in activation of existing protection relays, producing island disintegrating almost immediately after its formation. In this study performed for Manitoba Hydro it was observed that an uniform frequency exits at all points of separation during natural island formation when under frequency relays at strategic locations are used for island formation. After a major outage on the Manitoba Hydro power plan, the system form a natural island. The under frequency relays were set to 59 *Hz,* in order to generate islands at the natural points of separation. This is because the forced islands formed by under frequency relays are more stable but they need to be form between 10 and 15 cycles after under frequency detection [22].

One major application to form controlled islanding would be to form islanding before under frequency relays react and shed the load in the system, which usually results in shedding more load than is required to stabilize the system.

2.3. Decision Tree Modeling and Power System

The classification approach based on decision trees has a long history of success in power system applications mainly due to their simplicity [23]. Decision tree models are built from collected data and are used to classify occurrences of events. Classification modeling starts with a training data set. The observations in a training data set are known as training cases. The input variables for decision tree model are called predictors, or parameters. The target is known as the response.

Reference [24] presents a security-dependability adaptive protection scheme for the California power system. The optimal PMUs locations for the adaptive scheme are found by using a decision tree model. The results show that decision trees are capable of being trained from the training samples and representing a simple decision rule to adjust the reliability balance of the protection scheme.

Reference [25] discusses the application of decision trees for dynamic security enhancement in the power system. The decision tree model determines the security regions and the boundaries to predict the status of the power system. The Entergy power system model is used to test the topology presented in this study. The CART program is used to train the decision tree model in this study. For the preventive and corrective control methods proposed in this study most of the computation time is used for the generation of knowledge bases to train the decision trees. Two types of decision trees orthogonal and oblique model are used to determine the transient stability related to security regions and their boundaries. It is argued that propose method has successful results for Entergy power system [25].

2.4. Neural Networks and Power Systems

In recent years, the neural network methodology has shown promise in a lot of scientific research areas including power systems.

One of the important characteristics of neural network modeling is pattern recognition in a collaborative manner [26]. The neural network demonstrates a network with a defined number of layers and elements that are called neurons. These neurons may have different types of interactions between layers. The number of layers or neurons heavily depended on the optimization process for the model. It is also important to know that the goal is to minimize these layers in the modeling process to achieve minimum complexity for implementation and processing time [26].

The high adaptation of the model in neural networks has attracted recent researchers in the field of power systems. This modeling method seems to be able to perform complex mapping and to demonstrate which combination of values lead to a single system response. The structures of the network are not complicated to communicate the performance of the transients in the power system. The following sections present some of the examples of neural network modeling use in the power systems.

Damborg et al. [27] explores the potential of neural networks for the power system operation specially for aiding dispatchers with decision making. Moreover, it is shown that neural network with appropriate training is capable of determining the security status. The test system is made of 14 lines, 4 generators, and 8 buses. The neural network model contains 3 input neurons, one hidden layers with 10 neurons, and one neuron for output. The neural network model is trained by error back propagation method. The dynamic security of a test system is evaluated with a high accuracy but there are some problems with extending the results for a larger system.

Reference [28] discusses the implementation of a protection model for transmission lines based on neural network. It is argued that the neural network can learn different fault conditions and network changes which result in faster operating time with accurate results. Oleskovicz et al. [28] explore the neural network as a pattern classifier for operation of distance relay. The voltage and current after occurrence of fault are inputs to the neural network model. The output response of the neural network is a trip or no trip decision for the protection system. The neural network model is trained and tested only for forward and reverse single line to ground fault condition for different locations of the system. The data set for the model training contains total of 975 cases, which are all phase to earth faults in the different locations of the system. The 400 KV transmission systems are used to train and test the neural network. The neural network model has a strong performance to classify forward and reverse fault condition but a weak performance to classify whether the fault occurs in the relay primary protection zone. The best neural network model for this work has two hidden layers, with 10 nodes in the input layer, and 25 nodes in the first hidden layer, and 20 nodes in the second hidden layer.

Reference [29] discusses a method based on neural network for reliability analysis of power systems. Since the time required by neural network to generate response is short enough, the proposed method can be used for both on line and off line evaluation of reliability for generation and transmission systems. The results are obtained for the "IEEE reliability test system". The reliability evaluation of the generation system has an accuracy of 99%. The reliability evaluation for the transmission system has higher accuracy compare to the reliability evaluation for the generation system and the training times for this system are less than the generation system. It is discussed that the system adaptation for change of generation or transmission is easy due to the short time required for training. The proposed method requires small number of training samples. The quality of the training samples has direct effect on the accuracy of the results. The other factor for the proposed method performance is the quality of the neural network training algorithm. The neural network that is trained better will have improved performance for prediction.

Delimar et al. [30] discuss the neural network for power system topology recognition, in particular for finding irregularities in the power system topology. The "IEEE 24 node network", which contains 34 lines, is used as model for the study. A multi layers perceptron with 3 layers is developed. The input to the model is power line flow, and the output presents the bus switch state. The best model has four neurons in the hidden layer. The topology achieve by the neural network training can be match with the switch status that is reported to determine the accuracy of the power system topology.

Bretas et al. [31] discusses neural network for power system restoration. The proposed methodology is tested in a 162 bus transmission system. It is argued that neural network need small quantity of memory and processing time. The proposed restoration method contains few island restoration scheme (IRS). When the system is recovering from a disturbance each IRS is developing an island restoration plan. The off line study determine the number of IRS needed for this method. It is argued that the neural network is a successful method for power system restoration especially for its high processing speed.

Reference [32] explores the identification of nonparametric equivalents of dynamic power system with neural network. The method uses measurements at points where internal and external systems are interfaced. Moreover, the knowledge of parameters and topology of the subsystems are not necessary for this proposed method. The obtained equivalent is planned to be used for on line applications. The procedure is shown on a test model derived from the Western Systems Coordinating Council (WSCC). The system contains 46 nodes, and 19 generators.

Reference [33] investigate the optimization of the locations, sizes, and number of the reactive power control equipment by neural network modeling. The goal for this study is to increase the power system steady state security and stability performance. The neural network has four output variables which are defined as best, good, average, and poor. Two neural network methods used for training are multilayer perceptron and self-organizing feature map. A 39 bus power system is used for training and testing the neural network in this study. The neural network is trained to determine “Shunt Capacitor Banks (SCBs) installation plans”. The inputs to the neural network are the maximum power load, and maximum reactive power generation reserve. The results show that the neural network is capable of making optimal decision for reactive power control equipment.

Wenjin et al. [34] explore application of neural network for load forecasting in the power system. This study shows the short term load forecasting using multi layered feed forward neural network. The factors that affect the load set are working schedules, weather, or other elements like political or cultural activities. The performance of the network is tested by the hourly load data. The results for 24 hour load forecasting in advance for Jiangxi Province power system for one day winter and summer is presented in this study. Moreover, it is shown that the comparison between the results obtained by the neural network and the qualitative forecasting method are acceptable. It is argued that neural network has high speed for processing and high accuracy results for short term load forecasting.

Critical clearing time as explained in [35] is a complicated function of the pre fault, post fault, and fault condition structures that are related to protective relay policies. Based on their investigations, conventional decision tree techniques are not capable of defining this highly complex relationship.

The work in [28] shows an adaptive data mining approach based on implementing neural networks. One of the important characteristics of neural network modeling for them is the pattern recognition in a collaborative manner. Moreover, the high adaptation capabilities of the neural network for the complex mappings that relate the inputs into the single value describe a great advantage for the security assessment of power systems.

Reference [36] discusses neural networks as a classifier for security assessment of power systems. The multi-layer perceptron with back propagation method is used to train the networks due to its effectiveness to solve many practical problems. The correlation between the angles of the generators and the values of the sensitivity of generators in power generation is investigated. This work was performed for seven operating conditions with total constant load. The generation for each area was changing. The “IEEE 50 generator test system” was used as a model. The input for the neural network model is generator angle. The vulnerable case is presented as one, and not vulnerable case is presented as zero. Nine faults are used for training and testing of neural network. The neural network predicts the system vulnerability with high accuracy.

The neural network application discussed in [37] shows success in its performance in a fast and precise security system prediction. The correct selection of the training data set plays a crucial role in the prediction by neural network in this application. The above researchers performed a case study on the IEEE 50-generator system to demonstrate the effectiveness of their techniques. The technique is based on the use of Fisher's linear discriminant function, coupled with input choice method in order to select neural network training input for the power system security assessment. Fisher's linear discriminant is a method used in pattern recognition to search for a linear combination of input or data that separates two or more classes of the target data. This function is associated with the analysis of variance and regression analysis. The difference is that for the linear discriminant analysis the dependent variable is categorical while for the variance and regression analysis the dependent variable is numerical [37].

3. METHODOLOGY

The methodology used for islanding detection is based on data mining. The underlying hypothesis is that with the information provided by wide area measurements it is possible to predict islanding before it happens. A data mining approach is used to build the decision tree model and neural network model that are trained on a series of off-line dynamic simulations. An optimal decision, relevant to the prevailing system condition, is computed in real-time based on the models decision.

In recent years, critical infrastructures like power systems have been stressed by natural hazards, aging, and human errors that have led to many malfunctioning and misoperations. Recent blackouts are evidence of the vulnerability of this critical network system. Wide area measurement could help greatly to prevent these blackouts as mentioned in the final reports on the UCTE split of 2006 [9], and the U.S/Canada blackout of 2003 [1] that point to the lack of wide area awareness as a critical contributing factor of the blackouts. They recommend using PMUs to improve the real-time system observation.

System islanding is a feasible operation mode for a power system that is experiencing severe disturbances; it would help preserve as many areas in the system as possible. This work explores a new islanding detection scheme based on wide-area measurement and data mining. System monitoring locations are selected based on the wide area measurements required to achieve a feasible islanding prediction scheme that insures minimal power lost in the system.

The proposed methodology was designed and tested using highly detailed 4000 bus models of the California system. The system in these models consists of more than 3,888 transmission lines, and 1,124 generators. Two different models: heavy winter and heavy summer are used for this islanding study. There are four major control areas in California: Southern California Edison (SCE), Pacific Gas and Electric (PG&E), Los Angeles Department of Water and Power (LADW&P), and San Diego Gas and Electric (SDG&E). About 70% of the electric energy consumed in the state is generated in the state, about 22% is imported from Southwest, and about 8% is imported from northwest.

3.1. Islanding Detection

The underlying hypothesis is that phasor measurements at strategic buses provide enough information to predict islanding in the system. The methodology to implement the islanding protection scheme is concerned with:

- Where PMUs should be placed?
- How should an islanding state be defined?
- What attributes are sufficient for the purpose of classifying the system state?

The proposed methodology is based on data mining with most of the effort focused on generating the database of islanding cases and non-islanding cases, and developing the training process for classification.

3.2. Generating of Islanding and non-Islanding Cases

For the derivation of islanding and non-islanding cases PMUs are assumed to be installed at all 500 kV buses in the system models. Usually, attributes included as potential predictors are MVar flows, current magnitudes, voltage magnitudes, and angle differences [38]. For this research the following attributes were considered as potential predictors: rotor angles, voltage magnitudes, current magnitudes, current angles and voltage angles.

Of special interest for this research are voltage phase angles that can reveal the generator swing patterns at the early point of the swings, especially at the locations near a cluster of generators [21]. Data mining method needs a large sample of cases to generate accurate results. For this research it is assumed that PMUs monitor the state of all 500 KV buses and transmission lines by measuring magnitude and angle of voltages and currents. Contingencies and collection of voltage angles are created through dynamic simulation with in PSLF, the details of the developed program are shown in Appendix A. The cases where contingencies cause the formation of major coherent groups of the generators in the system are considered certain islanding cases. On the other hand some contingencies may cause some generators to form coherent group, but not unstable system, these cases are considered islanding cases as well. These cases may not be a major separation, but it is argued that the likelihood of hidden failures with creation of these coherent group of generators may cause the power system unstable as well. The reason is that these cases may not be major separation, but other factors like hidden failures in the system will eventually generate unstable system. It is argued that the likelihood of hidden failures with creation of these coherent group of generators may cause the power system unstable as well. A hidden failure is defined as a permanent defect in the system that will cause the incorrect removal of a circuit element as a direct consequence of another event [24]. Hidden failures stay inactive until a certain event causes their related relay not to operate in the way that is assumed in the protection design [39], [40], [24]. The cases where contingencies do not form groups of the coherent generators or separation of the system are considered non islanding cases.

To apply data mining algorithms a large amount of cases must first be produced. For this purpose different combinations of changing load for one area or more than one area at the time are considered. This methodology generates some impractical cases, which account for unpredicted events in the system. The type of contingency used for most cases is a 500 KV bus fault, followed by single line tripping. The fault is cleared after 5 cycles and the simulation runs for ten seconds to determine if islanding occurs.

The islanding cases are limited to generate islanding for lines greater than 230 KV. To guarantee sufficient number of islanding cases N-2 and N-3 contingencies are considered on 500 KV and 230 KV lines. The tripped lines are selected close to each other to increase the possibility of islanding in the system.

3.3. Decision Trees

Data mining is also defined as knowledge management, knowledge discovery, and sense making. Data mining is “an information extraction activity whose goal is to discover hidden facts contained in databases.” [41]. Moreover, data mining involves the systematic analysis of large data sets. Today, with help of Synchrophasors, power system companies can have real-time system observations resulting in massive amounts of data available every cycle. However, extracting useful knowledge from this source of potential intelligence is a difficult challenge, which is defined as a challenge in processing recognizable patterns in the data [42]. In the general algorithm the first step is to collect the data. The second step is to identify a target variable and a set of input variables. Each input variable is then examined one at a time. The aim is to create two or more groupings of the values of the input variable, and measure their similarity and differences. The groupings are selected in the way that maximizes similarity within groupings and minimizes differences between groupings. Once the groupings have been calculated for each input variable, the single input variable is selected in such a manner that maximizes similarity within groupings and minimizes differences between groupings. Decision trees or classification modeling has a long history of success in Data mining [43]. Models are built from collected data and are used to classify occurrences of events. Classification modeling starts with a training data set, the observations in a training data set are known as training cases, the variables are called predictors, or parameters, and the target is known as response.

A model must perform the following tasks:

- Provide a rule to transform a measurement into classifier
- Select a useful inputs
- Adjust the complexity to compensate for noisy training data

The training data is used to construct a model that relates the input variable to the target variable. The classification can be categorized into the following types: decision, ranking, and estimation. For the purpose of this research the decision type is used due to its simplicity for power system data type.

There is a need to emphasize that accuracy of the trained tree is directly dependent on the provided model.

What makes decision trees distinguishable from parametric models is that decision trees do not assume a particular structure to make a relation between the inputs and the target. This capability is important to detect complex input and target relationships missed by inflexible parametric models. The tree classifies observations in a top-down manner, starting from the root and working down according to the outcomes of the tests at the internal nodes, until a leaf node has been reached and a class label has been assigned [44]. One of the most popular decision tree algorithms is the one described by Quinlan [45]. The algorithm induces decision trees based on information theoretic concepts.

Adjustments and rules are defined by the user to limit growth of the tree and are based on engineering judgment. After the splitting stops because one of the stopping criteria has been met, the next step is pruning which may be summarized as follow:

- Prune the tree by cutting off weak branches. A weak branch is one with high misclassification rate, as determined from validation data. Pruning the full tree will increase the overall error rate for the training set, but the reduced tree will generally provide better predictive power for the unseen records.
- Prune the branches that provide the least additional predictive power.

3.3.1. Input Selection

Assessing the power of a classification technique is a significant practice. One could use the same data for both training and estimating the accuracy of a classifier. However, this will often result in an overly optimistic accuracy estimate since a trained classifier is typically somewhat biased towards the training data. This is especially the case for classification techniques with many parameters, like the ones used in power system.

The dimension of a problem refers to the number of the input variables. Data mining problems sometimes are massive in dimensions. One of the common techniques to decrease the complexity of the model would be assortment of the input data. This technique is commonly used in power system data modeling to remove irrelevant or redundant parameters from the classifier. The removal of inputs causes a faster training and faster evaluation of the classifier. Furthermore, classification models with fewer predictors are more attractive for humans since they are less complex and thus easier to comprehend. This predictor pruning may also augment the power of the classifier [46].

A redundant input is the one that does not give any new information that was not already explained by other inputs. It is not an obvious process to choose the best data set for a classifier. The only way an optimal input set can be obtained is when all the input data are exhaustively investigated and evaluated [47].

A backward selection algorithm begins with the whole data collected and prunes the inputs that are not desirable by steps. On the other hand, a forward selection algorithm begins with the vacant input set and adds input variables that are desirable in each step. For the purpose of this work, a backward selection scheme is used because the collected data from simulation has all the desired parameters to start with.

3.3.2. Tree Training Methodology

In a training data set, the essential idea is to choose each split so that a subset is more pure than the parent set. This can be explained with the following example:

Consider there is a mix of data for real estate houses that classifies them in 8 classes denoted by L_1 , $L_{2,}$ …. $L_{8.}$ Each class has its own profile for selected criteria for a range of prices. The data consisted of a number of profiles for each of the different house classes. The goal is to construct a classifier that can take a profile as input and generate a dependable prediction of class association.

One of the initial problems is that some of the information in any profile is unnecessary or redundant. In the collected data set (e.g., like power system or the example above) one of the solutions for classification is the structure of the tree where the decision tree classifiers are generated by repeated splits of subset L into two more pure subsets.

The subsets that do not split anymore are terminal subsets. Each terminal subset presents a class, but more than one terminal node could have the same class.

The splits are formed by conditions on the subset. For example, the first split of L in L_1 , and $L_{2,}$ May be as follows:

$$L_2 = [l : L_5 \leq a] \tag{8}$$

$$L_3 = [l : L_5 > a] \tag{9}$$

The whole process is based on how to choose the most efficient split at the node. The node that is end of division for data is defined as terminal node. The main philosophy that needs to be maintained in decision trees is that the selection works to make each subset or child more pure than the parent or previous subset.

The main question that may arise is how to terminate the development of a tree. The answer is the tree would stop growing when a node reaches a state where no substantial reduction in impurity is obtained by more splitting. The prune process will then begin for the large tree going upward and achieve the reduction in the sequence of sub trees.

3.3.2.1. Training Decision Trees for Classification

In this work the SAS Enterprise Minor algorithm is used to grow decision trees. In the proposed method, the decision trees are trained offline to be used for online application. The accuracy of the results has a direct relationship with the accuracy of the models and decision trees need to be updated any time a significant change occurs in the system models. Tow system models are used in this work: The California Heavy Winter Model and the California Heavy Summer model. As described by [21] the amount of imported power, peak load, maintenance schedules, and generation dispatched are different for each season, and that is the main reason to have a different model and develop different trees for each season. The California power system has an additional model for light summer with minor differences from the heavy summer case, this model was not used in this research.

Decision trees usually relate to a categorical target variable, which can be binary, nominal or ordinal. Different variable selection methods are used to identify the best-input variable for classifications as described in the previous sections. The numbers of input variables are also refer to as dimensions of the problem or degrees of freedom. The dimensionality is defined as an exponential increase in the required data to populate space as the dimension increases. The practical ability to fit a flexible model to noisy data is limited by the dimensionality. In this research the cases that are collected for the heavy winter and heavy summer models are used to train different trees and select the best trees with minimum cost function or misclassification rate.

3.3.2.2. Cost Function

The cost function is a measure of the model's quality. It is a function of the error between the model output and the system output. Raol [48] approached a model base, and defined a mathematical model of the dynamic system as $z = H\beta + v$, and $y = H\beta$ Where y is $(m \times 1)$ vector of true outputs, z is $(m \times 1)$ vector that denotes the measurement of the unknown parameter through H, and v is the error between true output and output through the unknown parameter. The chosen estimator β should minimize the sum of the squares of the error.

$$J \cong \sum_{k=1}^{N} v_k^2 = (z - H\beta)^T (z - H\beta) \tag{10}$$

J is a cost function, and v is residual errors at time k. Superscript T is for the vector transposition. The detail description is explained in the following sections. The cost function is a measure of the model's quality. It is a function of the error between the model output and the system output. Raol [48] approached a model base, and defined a mathematical model of the dynamic system as $z = H\beta + v$, and $y = H\beta$

Where y is $(m \times 1)$ vector of true outputs, z is $(m \times 1)$ vector that denotes the measurement of the unknown parameter through H, and v is the error between true output and output through the unknown parameter. The chosen estimator β should minimize the sum of the squares of the error.

$$J \cong \sum_{k=1}^{N} v_k^2 = (z - H\beta)^T (z - H\beta) \tag{11}$$

J is a cost function, and v is residual errors at time k. Superscript T is for the vector transposition. Cost function is a measure of the model's quality. It is a function of the error between the model output and the system output. Superscript T indicates vector transposition. The minimization of J with respect to β yields:

$$\frac{\partial J}{\partial \beta} = -(z - H\beta_{LS})^T H = 0 \qquad \text{or} \qquad H^T (z - H\beta_{LS}) = 0 \tag{12}$$

After simplification, we get:

$$H^T z - (H^T H)\beta_{LS} = 0 \quad \text{or} \qquad \beta_{LS} = (H^T H)^{-1} H^T z \tag{13}$$

Equation (13) is defined as the best parameter estimation. We can obtain β using the pseudo-inverse of H, $\beta_{LS} = (H^T H)^{-1} H^T z$. This shows that the estimate can be obtained from the known data.

3.3.3. Assessment of Model Performance

In SAS Enterprise Minor, part of the collected data, which is called the training data set, is used for fitting the model. The data is also collected for empirical validation or out of sampling test set. The tuning procedure improves and optimizes the selected model on validation data. The out of sample data with variation in the operating points up to 6% for each area in the model is used for a final unbiased assessment of the classification. As part of this variation, the load is changed for one area at a time or for a combination of two areas at once.

3.3.3.1. Cross Validation

The sample test is used to estimate the accuracy of the classification. This indicates the need to use one set of sample data for training, and a disjointed data set for validation [43].

The sample test is used for collection of data limited to a few hundred cases, but since the cases for this study are several thousands, cross validation is considered.

As described by Walpole et al. [49] cross validation is used to make decisions for selecting the optimize model between all models that are created with the same data set. The model that best predicts or estimates mean response is desired for this work. It is also important to consider how well the model predicts response values that were not used in the selected models, this requires cross validation error. These errors in prediction, also known as press residuals, are defined as follow

$$\delta_i = y_i - \bar{y}_{i,-i,} \qquad i = 1,2,3,\ldots.n, \tag{14}$$

Where $\bar{y}_{i,-i,}$ is defined as the prediction of the *i*th data point by the model that did not use the *i*th point in the calculation of the coefficient. The press residuals are calculated as

$$\delta_i = \left(\frac{e_i}{1 - h_{ii}}\right), \qquad i = 1,2,3,\ldots.n, \tag{15}$$

Moreover, the out of sample data is used to avoid over fitting or under fitting. The classification model calibrates its performance with this out of sample data with the least cost function. This sets the stage for training decision trees for classification with the highest accuracy for the training data, but generalizing the classification from the training data to an independent and similarly distributed data (out of sampling) that should not result in a substantial decrease in the accuracy.

The test data set is also created to achieve unbiased estimates of the model performance from a single selected model. The partition strategy here is to select more data devoted to training result, which leads to a stable classification model. The partition node maintains the proportion of zeros and ones in the training and validation partition.

For example, consider a data set shown in Figure 1 with two inputs and binary target. The input L and P locate the case value in the unit square. The target outcome is represented by color: Green is zero and blue is one.

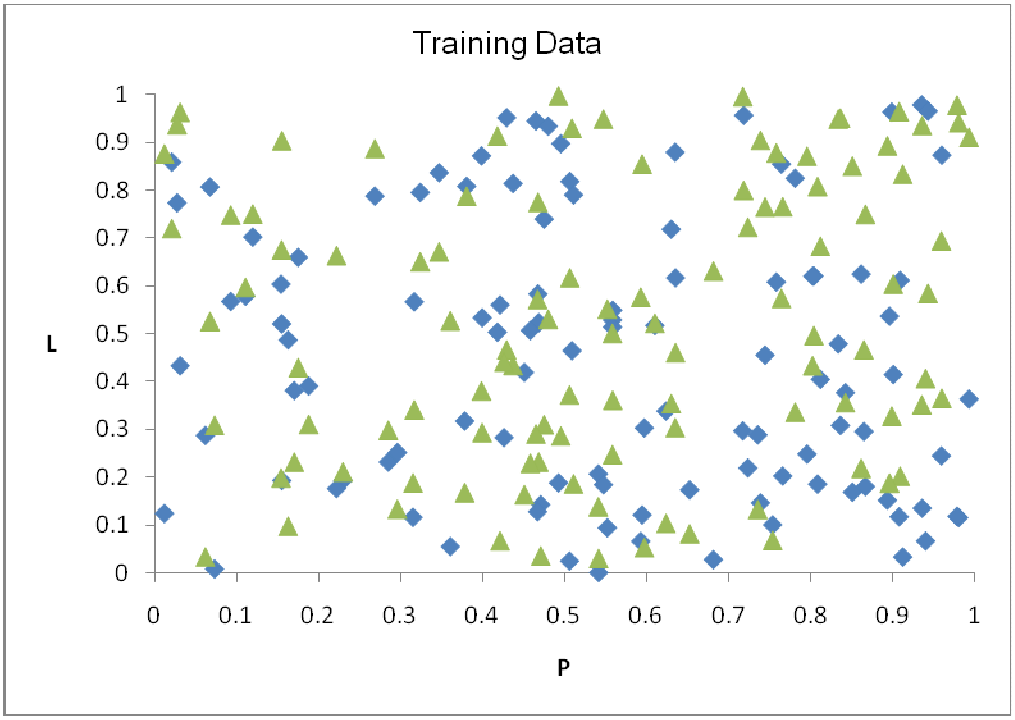

Figure 1: Simple Classification Demonstration

Decision tree uses regulations including values of the input to classify cases. These rules or regulations are assigned hierarchically with a tree like structure, hence the name decision tree. In Figure 2, the data is subdivided in two branches at the root node based on values < 0.4 or ≥0.42. The input values of the cases eventually terminate at the terminal node, T.

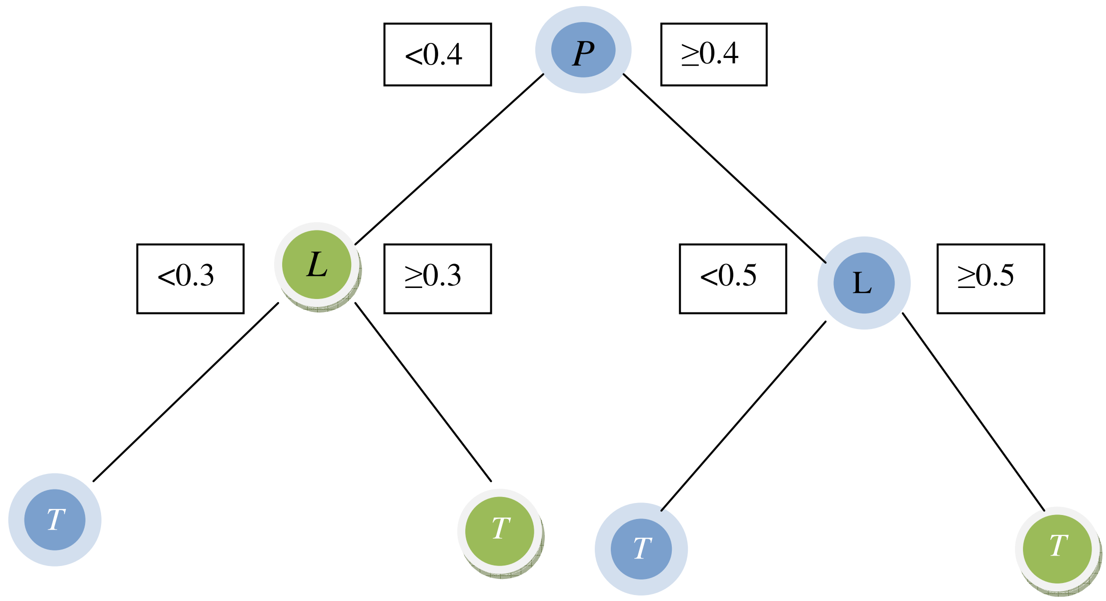

Figure 2: Decision Tree Example

The first part of the algorithm is the split search, which starts by choosing a variable for partitioning training data. Since the measurement scale for the voltage angle is interval, each unique input serves as an optional split point for the data. For a selected variable and fixed split point, two groups are generated. Cases with variables less than the split point are on the left branch. Cases with variable values greater or equal to the split point are on the right branch. The column in the data table specifies the branch direction, and rows determine target value as zero or one. The single split is selected. At each split the data will be more pure than the parent branch. The process will continue down the tree by making each child split more pure than the parent split till the splitting does not add to the accuracy of the result. At this point the tree will be terminated and final is defined as the terminal node. Since the output response is binary as zero for islanding cases and one for non-islanding cases, the chi-squared statistic is computed for testing of the assessment in the SAS Enterprise Minor program.

3.3.3.2. chi-squared Distribution

In order to quantify the independence on values in the data column, the Pearson chi-squared statistic is used. As described by [49] the chi-squared distribution for the continues random variable X, with v degree of freedom exist if its density function define as

$$f(x) = \frac{1}{2^{v/2}\tau(v/2)} x^{v/2-1} e^{-x/2}, x > 0 \tag{16}$$

And

$$f(x) = 0 \quad \text{anywhere else,} \tag{17}$$

Where v is a positive integer.

For the chi-squared distribution, the mean and variance are not difficult and complex to achieve.

The mean and variance of the chi-squared distribution are

$$\mu = v \quad \text{and} \quad \sigma^2 = 2v \tag{18}$$

The chi-squared statistic is computed for testing in the SAS Enterprise Minor program, since the output response is binary as zero for islanding cases, and one for non-islanding cases. It can be summarize that each test uses the squared, scaled residuals from the analysis with the particular source variable as the predictor variable. The scaling of the squared residuals consists of dividing each by the error sum of squares which is itself divided by the number of observations.

3.3.4. Decision optimization

Since the target value that defines islanding or non-islanding is a binary value, based on the outcome the author considered a primary and secondary classification type. Decision classification rate by their accuracy is based on the amount of agreement between classification and outcome. In order to test the accuracy of the classification for islanding and non-islanding cases, the misclassification rate is considered, which is defined as the amount of disagreement between classification and outcome. The tree model with the least branches and minimum misclassification rate is selected as final result. The SAS Enterprise Minor programs calculate the misclassification rate and plot the result for each tree model.

3.3.4.1. The factors affecting behavior of the algorithm

Many parameters in the SAS Enterprise Minor determine the behavior of the algorithm and eventually affect the optimization process which is categorized in four major groups: the metric use to evaluate unlike splits, the pruning method used to increase efficiency of grow tree, the number of split in a node, and the rules used to end the growth in each model. The selection of each of these parameters affects the classification results and optimization. The heuristic algorithms, which combine branches and convey a consolidated set of observations to different branches, are used for this work. The split with the best worth value is selected for each classification. For example, based on the application, more than one split may be selected for each node. The proposed procedure is based on the single split for each node. The rules defined for termination of the tree growth are based on accuracy. If splitting more data does not add to the accuracy of the result the splitting terminates and the node is define as terminal node. The other two factors are similar splits and pruning method. The similar splits are eliminated to increase the accuracy of results. Pruning methods begin from the terminal nodes by eliminating the nodes that have no effect on the cost function results.

3.4. Predictive Modeling: Neural Network

The neural network models are known as mysterious and powerful predictive tools. One of the positive features of neural network modeling that makes it a better candidate than regression prediction is that, for a properly trained model, it can make any association between input and target value. Since flexibility has always a price, the neural network does not address input selection as decision trees do. This is not a disadvantage for this work since input selection is performed based on the characteristic of the parameters in the power system and the work performed by decision tree. The neural networks predict cases using a mathematical equation, which includes values of the input variable. The neural network has biological roots, which are the reason for its components to have different names. The effort of this work is to use neural network training to compare accuracy and predict control islanding in the power system.

The neural network is an interconnection of nodes. As described by [50] the neural network is composed of three main parts; node, network topology, and learning rules. A simple neural network is shown in Figure 3.

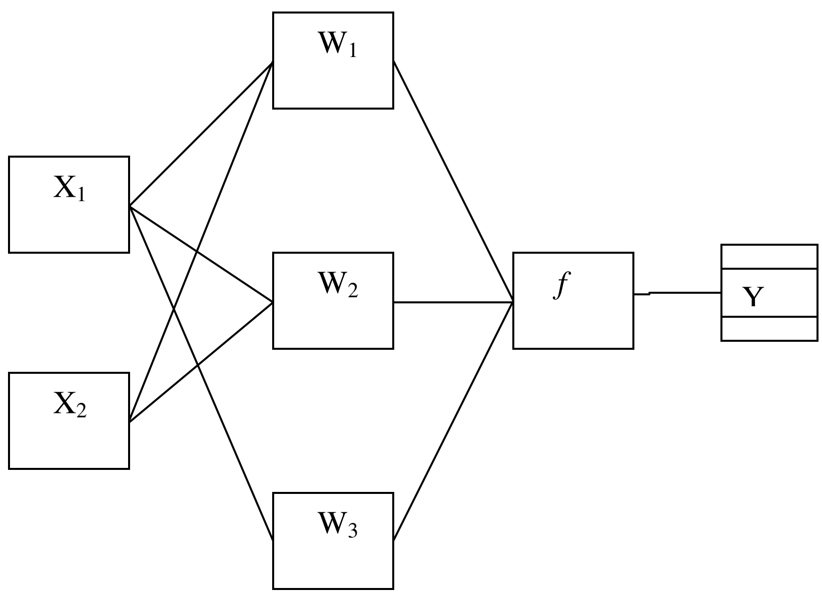

Figure 3: Simple Neural Network Diagram with Transfer Function and Weight

Each node receives inputs from other nodes that have their own weights. The relationship between output response and node weight is expressed as the following mathematical model

$$y = f\left(\sum_{i=0}^{n} w_i x_i - T\right) \tag{19}$$

Where

f : Transfer Function

y define as output of node

W_i The weight (20)

X_i The input

T define as threshold value

The simplest transfer function can be a step function, which is defined as

$$y = 0 \text{ if } \left(\sum_{i=0}^{n} w_i x_i > T\right) \text{and } y = 1 \text{ if } \left(\sum_{i=0}^{n} w_i x_i < T\right) \tag{21}$$

In a neural network model input layers, output layer, and hidden layers exist. It is the work and art of the designer to determine the best number of nodes in each layer, the number of layers, and the optimal connection between nodes and layers, which is determined by the model that gives the least misclassification rate. The input layers are voltage angles at the high voltage buses in the transmission system, and output layers are defined as zero for islanding and one for non-islanding case. Many models are generated with verity of combination for hidden layers and neurons in each of these hidden layers, based on calculated misclassification rate by the program the optimal result is selected. These combinations of hidden layers and their neurons are generated manually and results are inspected individually.

The two popular learning process, error correction method and nearest neighbor method are used to train the networks. The error correction method due to its simplicity is used for this work. Training the network is an exhaustive process that is part of the author contribution for this work. Lau [51] describes the error correction method as following:

The error function is defined as the difference between the node output and target output

$$e_k = y_k, n - y_k^* \tag{22}$$

y_k, n output of the *Kth* output node at step n

and y_k^* the target output for the *Kth* node (23)

If λ is a positive constant that determines the rate of weight adjustment, one can calculate the new weight for input x_i as

$$w_{kj,n+1} = w_{kj,n} - \lambda e_k * X_j \tag{24}$$

This weight vector is adjusted in every single step until the model converges.

The nearest neighbor method [51], which is often used in classification modeling, is based on the spatial similarity. This mode is popular in the scientific and engineering field because of its simplicity.

In this method for any data x_t , its nearest neighbor should have a distance that meets the following

$$\min_{t=1}^{N}[D(X_i - X_t) \tag{25}$$

Where

$$\begin{aligned} &D \text{ distance measurement function} \\ &N \text{ training sample size} \end{aligned} \tag{26}$$

There are two classifications of neural network modeling; feedforward network and feedback ward network, which are described in the following sections.

3.4.1. Feedforward Neural Network Model

The feedforward is a type of connection between nodes that is in one direction, and it has no loop back. This network is static, which means one input is related to a particular output. The feedforward network that is used in this work is perceptron [51]. The single layer perceptron is used with continue and binary input. This network has capture attention of many researchers and scientists due to its simplicity for training it to recognize the pattern in the data, which is the main reason to use this method for this work. The function for this network is as follow

$$y = f_n\left(\sum_{i=0}^{N-1} w_i x_i - \theta\right) \tag{27}$$

Then the class is defined as follow

$$y = +1 \Rightarrow classA \tag{28}$$

$$y = -1 \Rightarrow classB \tag{29}$$

In this network, a single node computes a weighted sum of the input elements, subtracts threshold θ, and calculates the result through a nonlinear process whose output is +1 or –1 [51].

3.4.2. Feedback Neural Network Model

The other type of connection for a neural network is loop back connection. This means that the output of a node can be input to the same or to a previous level node. This feedback network is dynamic. For example, hopfield net is a feedback neural network method, which uses binary inputs. The algorithm for hopfield network is described by [51] as follow:

First: assign connection weights

$$t_{ij} = \left(\sum_{s=0}^{m-1} x^s{}_i x^s{}_{j,} \right) \quad i \neq j \tag{30}$$

$$t_{ij} = 0 \quad i = j\ , 0 \leq i, j \leq M - 1 \tag{31}$$

Where t_{ij} is the connection weight from node *i* to node j

Then initialize with an unknown input pattern

$$\mu_i(0) = x_i \tag{32}$$

$$0 \leq i \leq N - 1 \tag{33}$$

Consider $\mu_i(0)$ is the output of node *i at time t*

Then it iterates until convergence.

$$\mu(t+1) = f_n\left(\sum_{i=0}^{N-1} t_{ji}\mu_i(t)\right),\ 0 \le j \le M-1 \tag{34}$$

Until there is no change in the output, this process repeats from initialization with an unknown input pattern process step. There are other network training methods, but discussion of each is beyond the scope of this work. For a thorough review see [51]. The feedback method is not used for this work as the amount of data in power system at any time is large and it is critical to have a model that requires the least computational time and minimum amount of complexity if possible. The model obtained with feedforward has less number of neuron and layers, thus it is more feasible for the purpose of this research.

3.5. Neural Network Modeling

In a neural network model, input layers, output layers, and hidden layers exist. It is the work and art of the designer to determine the best number for nodes in each layer. Neural network has two classifications-feedforward network and feedback ward network. The feedforward network is the base for modeling in this work due to its simplicity for implementation. The amount of data in power system to analysis for each time is very large and it is critical to have model that requires the least computational time. In real time by knowing loss of synchronism fast enough, operators can detect islanding and carefully designed transmission interfaces and rapidly stabilize each island [21].

3.5.1. Neural Network Training

Training a neural network is a time consuming process. Typically, one hidden layer is chosen for the model. The number of output nodes depends on the number of classes. About fifty-one attributes are inputs to the model. These attributes in this research are voltage angles measured by PMUs at 500 KV buses in the California power system. Considerable amount of time is required to come up with the optimal number of the hidden nodes, which typically are somewhere between the number of input nodes and twice the number of input nodes.

First, the weights are initialized randomly. Then the inputs are propagated forward by the following steps:

- The input I to a node is a linear combination of its inputs
- The output O is a non-linear function of I (sigmoid): $1/(1+e^{-I})$

This process is repeated with number of iteration till the weight is small or the cost function is minimized for the optimal number of layers. The weight w_{ij} is weight from node i in one layer to j in the next layer.

This process is performed by applying the input pattern to the first layer. Then the values of each node are computed layer by layer. The threshold is often used at output nodes to make a binary response. For the purpose of this work the output answer is zero or one.

The Sigmoid function, which is a mathematical function having an "S" shape is used instead of a linear function so that the network can model classes with non-linear boundaries. The sigmoid function is defined as follow:

$$p(t) = \frac{1}{1+e^{-t}} \quad (35)$$

The value of sigmoid is in the range between zero and one (zero for large magnitude negative, one for large magnitude positive).

3.5.2. Number of hidden neurons

The selected number of neuron and hidden layers are based on misclassification rate, for each model that is selected as an optimal model. There are two sets of data; one is used as the training set and the other one is used as out of sampling data. First the number of neurons is changed from one to 15, one at a time. The network is trained for each set and the misclassification rate is calculated for both training data and out of sampling data for each case separately. Finally, the number of hidden neurons is chosen that gives optimal average validation performance. This performance is measured on the validation set as described in the following section.

3.5.3. Complexity Optimization

One of the essential parts of the neural network modeling is complexity optimization. In the final step the iteration with the smallest value of the selected fit statistic is selected as final model.

At the beginning of the model optimization, an initial value is assigned to each of the weights in the model. An initial fit statistic is calculated on training and validation data. Since in this work the target is binary, fit statistic is proportion to the log likelihood function, which is described by [52] as

$$\sum \log\left(\hat{p}_i\left(\hat{w}\right)\right) + \sum \log\left(1 - \hat{p}_i\left(\hat{w}\right)\right) \tag{36}$$

$\hat{p}_i$ is the predicted target value

$\hat{w}$ is the current estimate of model parameter

The parameter estimates update in a way that minimizes the value of the objective function in the neural network-training model.

The author selects the iteration with minimum misclassification rate and less possible complexity for the model. The next chapter presents the simulation results for both decision trees and neural network model.

4. SIMULATION RESULTS

Two seasonal models of the California system, heavy winter and heavy summer, are used to demonstrate the methodology proposed in this work. The author created an exhaustive list of study cases. A case is defined as “zero” or “island” if the combination of faults generates islands. However, the cases in which faults did not create islands are considered “one” or “non island” cases. Further details on these generated study cases are thoroughly discussed in this chapter. The summary of the simulation steps and process are described in Figure 4.

It is assumed that PMUs are placed at all 500 KV buses in the system. The author initially studied many attributes like current magnitude, current angle, and voltage magnitude. However, the optimal results were captured by the voltage angle attribute. The decision tree has the best splitting results with the voltage phase angles and is a good predictor parameter. It is observed that with the other parameters the decision tree model will have many branches with high misclassification rate. Moreover, the voltage phase angles can reveal the generator swing patterns at an early point of the swings, especially at the locations close to a cluster of generators.

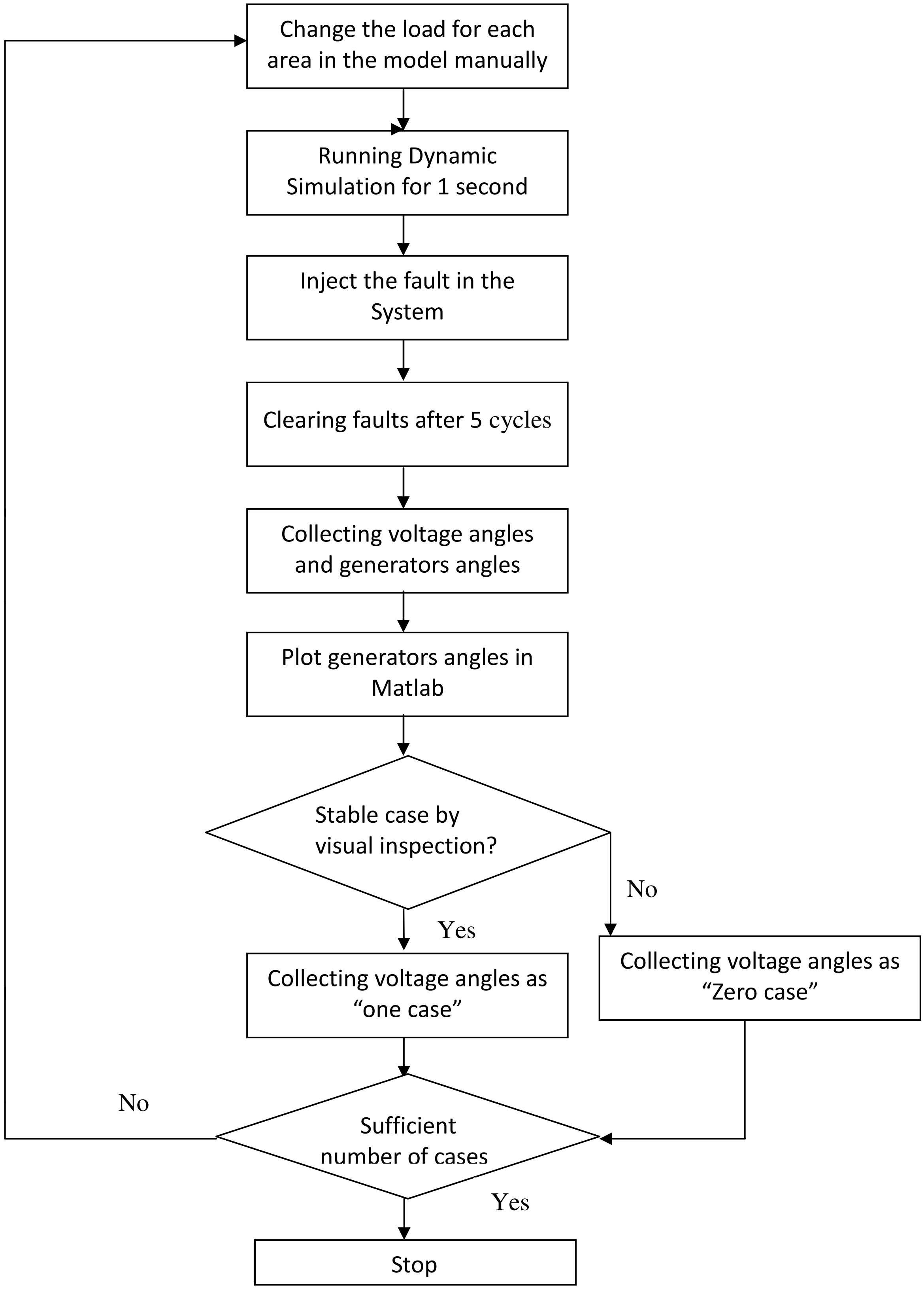

Figure 4. Summary of Simulation Process

In order to generate enough cases, the load in each area for California system is changed. This variation is performed manually for one area at the time or combination of more than one area at the time. Then the following steps are followed.

- Running dynamic simulation for 1 second
- Injecting faults in the system (line fault or bus fault)
- Clearing fault after 5 cycles
- Collecting voltage angles of all 500 KV buses and generator angles
- Transferring generator angles of all generators in the system to Matlab
- Plot the generator angles in Matlab
- If generator angles are stable, then collect all 500 KV bus voltage angles as "one case" or non-island (This is one row of entry to the SAS Enterprise Minor program)
- If generator angles are non-stable, then collect all 500 KV bus voltage angles as "zero case" or island (This is one row of entry to the SAS Enterprise Minor program)
- Continue from step one to generate all zero and one cases

There are two main programs that are used in this research work: Positive Sequence Load Flow program (PSLF), and SAS Enterprise Miner statistical analysis tool for data mining. PSLF is designed to provide comprehensive and accurate load flow, dynamic simulation and short circuit analysis [53]. This program is well known and it is used by researchers in the industry and academy for its capabilities to analyze power system. PSLF is ideal for simulating transfers of large blocks of power across a transmission grid, importing or exporting power to neighboring systems. changes the load for each areas of the system, and perform dynamic simulations.

The Enterprise Minor analysis tool of the SAS statistical program is used for data mining modeling as decision tree and neural network. The SAS Enterprise Miner program reorganizes the data mining process to create highly accurate predictive models based on analysis of large amounts of data [54]. This software package is used for research in academy and industry for its great set of incorporated abilities for generating and distribution intuitions that can be used to drive optimize decisions modeling. There is no preference to use this tool compare to other statistical analysis program available in the field such as CART.

Matlab is also used for converting the text data files obtained from dynamic simulations to plots for visual investigation of islanding and non-islanding cases. There is a need to plot all the generator angles at the same time for visual inspection with numerous amount of information to be analyzed and this is beyond the ability of the PSLF program plot.

4.1. Heavy Winter Model: Decision Tree

The various operating conditions are generated for the system. As described in chapter 3, the type of contingency is a 500 KV bus fault, followed by single line tripping. Many cases are created by N-2 contingency analysis on 500 KV and 230 KV lines. Loss of a 500 KV bus followed by a breaker fault generates islanding cases in some areas. It is important to mention that the faulted lines need to be close to each other; if they are far away islanding is not possible. The losses of two lines that are heavily loaded causes loss of synchronism. The assumed condition can easily results in a real system when a critical line is out of service, and nearby highly loaded line trips. Programs are developed with PSLF software for introducing faults in the system, dynamic simulation, and collecting angles data as shown in Appendix A. Figure 5 and 6 show results of a stable and a unstable case for the heavy winter model. Figure 5 represents a stable case where the y axis shows the generator angles and the x axis the time period of simulation. The fault is introduced to the system after one second and causes a minor fluctuation in the generator angles, but does not cause any of them to make a coherent group and separate from the rest of the system.

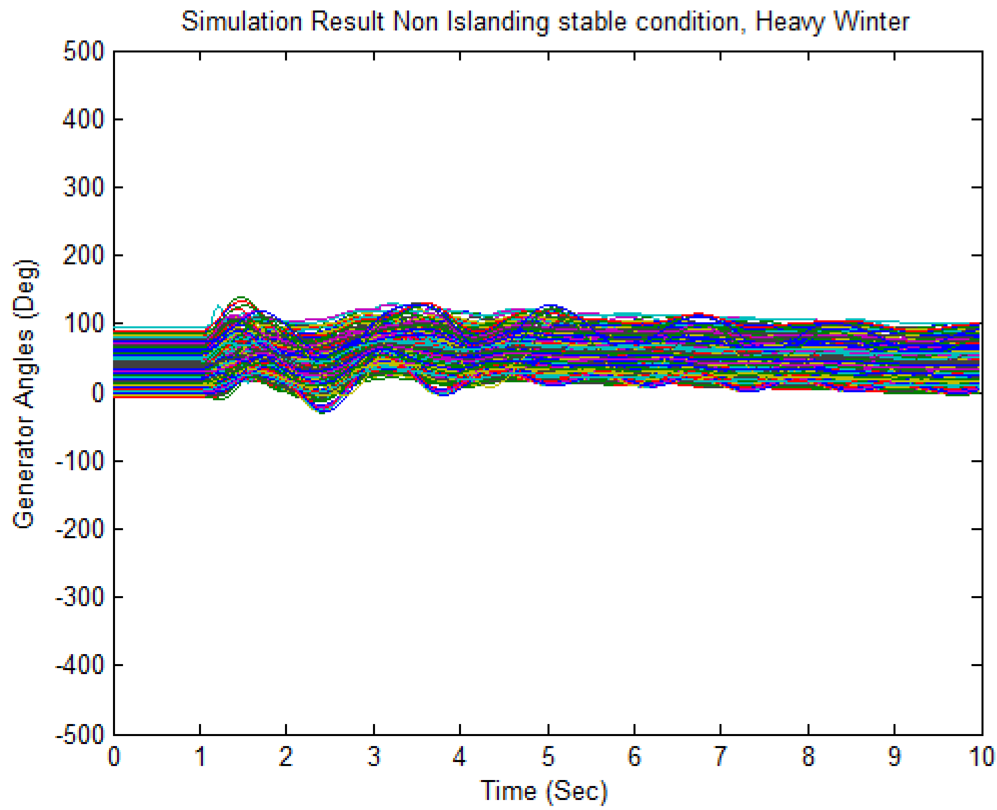

Figure 5: Generator Rotor Angles of case 1065, Stable Case

Figure 6 shows an unstable case where a group of generators separates as a coherent group. The decision trees are the simplest type of classification that usually relate to a categorical target variable, which can be binary, nominal or ordinal. For the purpose of this work the target variable is defined as binary. The learning sample contains more than 50 attributes for voltage angle, measured in degrees, at 500 KV buses in the system.

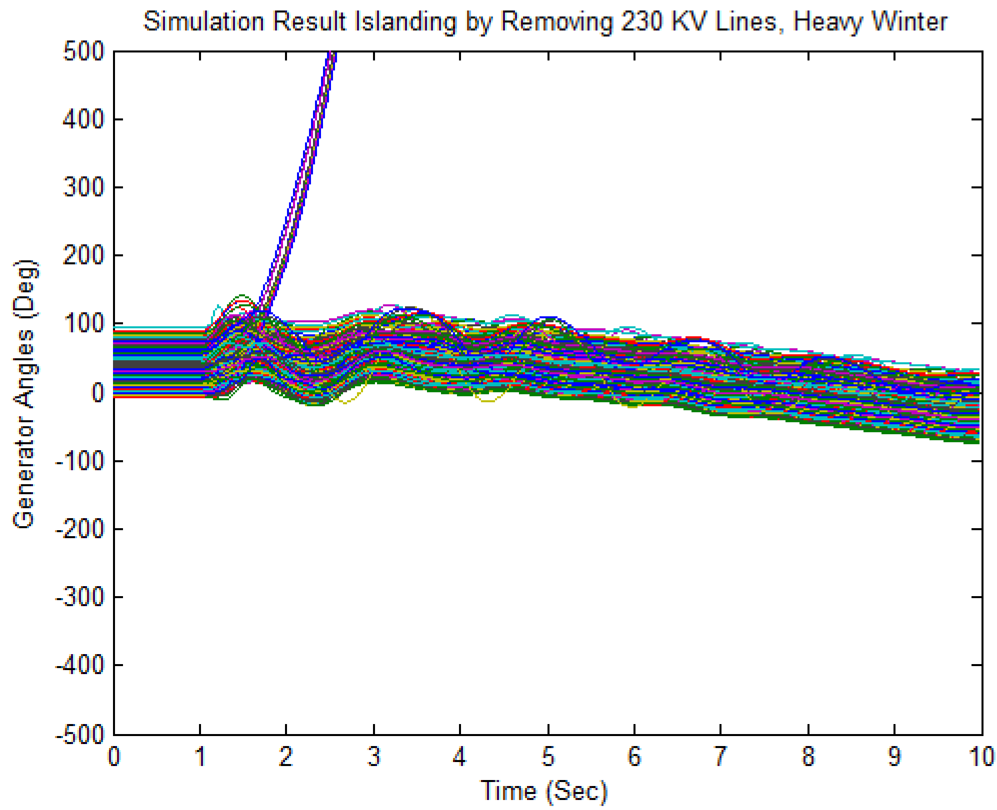

Figure 6: Generator Rotor Angles of case 151, Unstable Case

This arrangement forms a matrix of more than 1000 rows (cases) and more than 50 columns (attributes) for each simulation. Table 1 shows an example of part of the data table for the heavy summer model.

Cases	Response	Voltage angle Bus 1	Voltage angle Bus 2	…….	Voltage angle Bus 52
1	1	-2.71	-1.22	………	14.7
2	1	-2.4	-1.17	………	15.27
3	1	-2.44	-1.11	………	13.43
4	0	-2.38	-1.11	………	14.25
5	1	-2.39	-1.28	………	13.37
6	0	-2.85	-1.2	………	12.59
………	………	………	………	………	………
Case 1392	1	-2.83	-1.07	15.39	12.83

Table 1: Sample collected cases data for heavy summer model

The majority of the work is associated with generating these cases, and data collection based on observation method. The goal is to generate the decision tree with the highest accuracy and minimum cost function. Figure 7, shows the decision tree result for heavy winter model. The recommended PMU location based on the result for the heavy winter model is Tesla.

Figure 8 shows the decision tree for the heavy winter model of the California system. The decision tree contains the following items:

- Root node (First red block on the top), this node in the tree contains all observations. For this selected model there are a total of 1276 observations for training.

- Leaf nodes (blocks below the root node), for Figure 8 these nodes are also the terminal nodes that contain the final classification for the set of observations.

Figure 9 shows the result of a decision tree with more leafs or PMU locations.

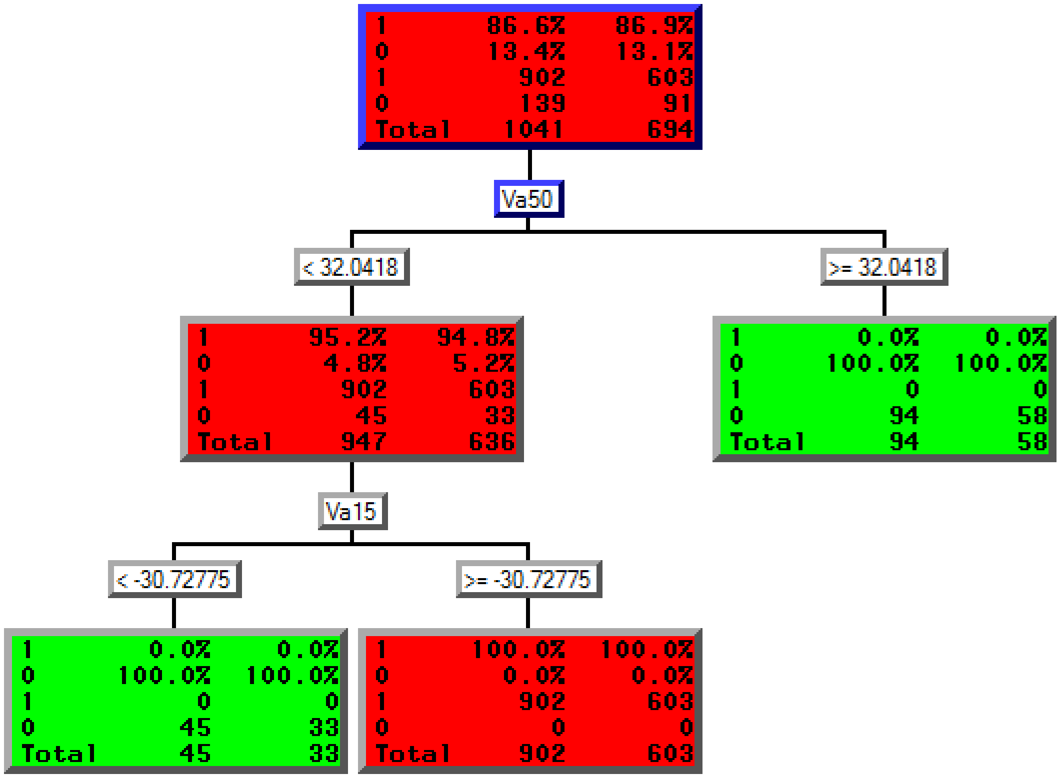

Figure 7. Decision Tree Result for HW with more Branches

In each leaf in figure 7, the second column displays results for training data and the third column for out of sampling data. The root node shows that 1041 cases are used for training and 694 cases are used for out of sample testing. The first and second rows in each node show the percentage of cases classified as ones and zeros respectively. The third and fourth rows show the actual number of one and zero cases. The final row shows the total number of cases for training and validation. For example in the root node, 86.6%

or 902 cases out 1041 cases for training are “one” cases, and 13.4% or 139 cases are zero cases. The third column shows that 86.9% or 603 cases out of 694 cases for out of sampling are classified as “one” and 13.1% or 91 cases are classified as “zero” cases. The location that tree splits is shown as Va 50, Va 15. The splitting value for the Va 50 is 32.04 degress, and for Va 15 is -30.72. At location Va 50 the data is split and classified as case “one” for values greater than 32.04 degree, and as “zero” cases for values less than or equal 32.04 degree. In the same manner at location Va 15 data is split and classified for values greater than -30.72 degree as “zero” cased and zero cases consider for values less than or equal to -30.72 degree.

In order to measure the accuracy of the modeling the misclassification rate is calculated by the program for both training data and out of sampling data. Misclassification rate can be defined as percentage of the data that is in the wrong category at the end of the tree growth. For example, if there are 100 data points that classify as zero cases the ideal model with zero percent of misclassification rate will have all of these data in the zero blocks.

Figure 8 represents the misclassification rate plotted by the program as assessment for the model accuracy. The misclassification rate is less than one present for tree model with the higher number of branches. This misclassification rate shows that the model has accuracy of more than 99%, but to optimize the model is desired to have minimum number of branches for the tree model. The blue line represents the training data and the red line the out of sampling data.

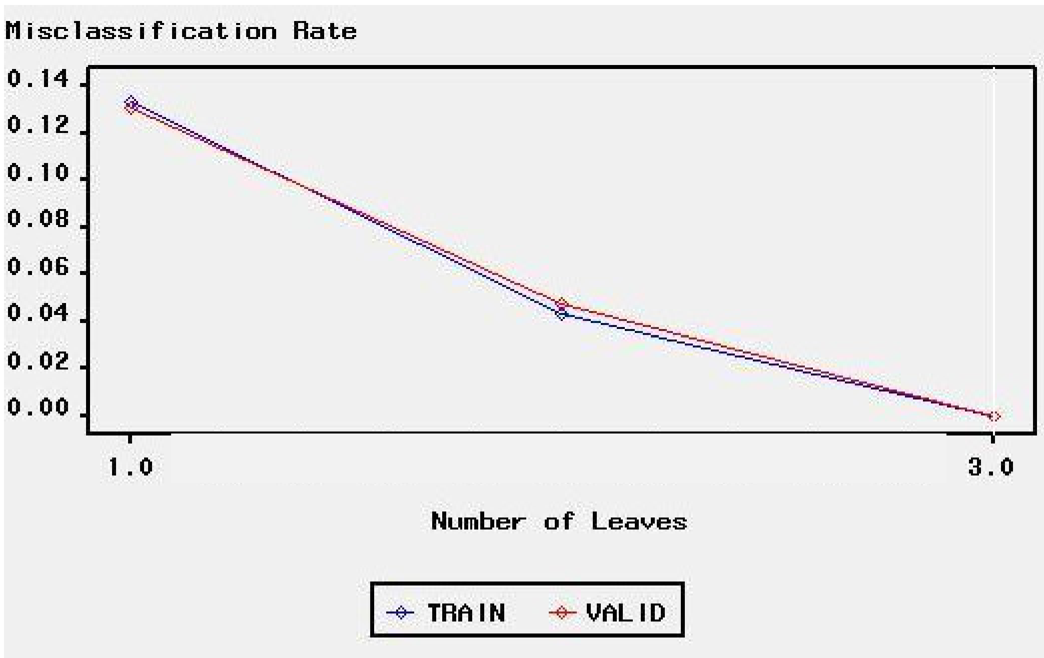

Figure 8. Misclassification Results for Higher Number of Branches Tree

Figure 9 and 10 show a sample of one of the non-optimal trained decision tree obtained in this research. One of the important goal for this decision tree modeling is to have a minimum number of branches, sine each branch location requires the installation of a PMU and required communication node. The goal is to achieve a decision tree model with high accuracy and minimum number of PMU installation. The decision tree model shown in Figure 9 is not the optimal one due to its number of branches. It is desired to have a decision tree within a small number of branches.

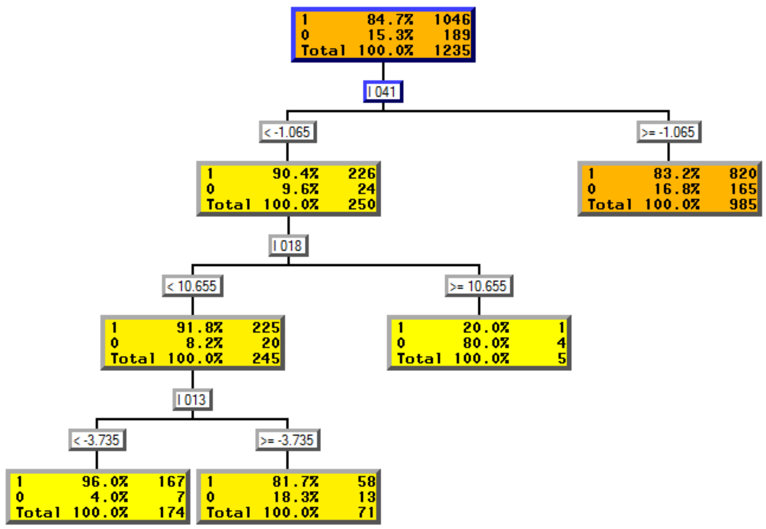

Figure 9. Heavy Winter Decision Tree Result with more Branches and Higher Misclassification Rate

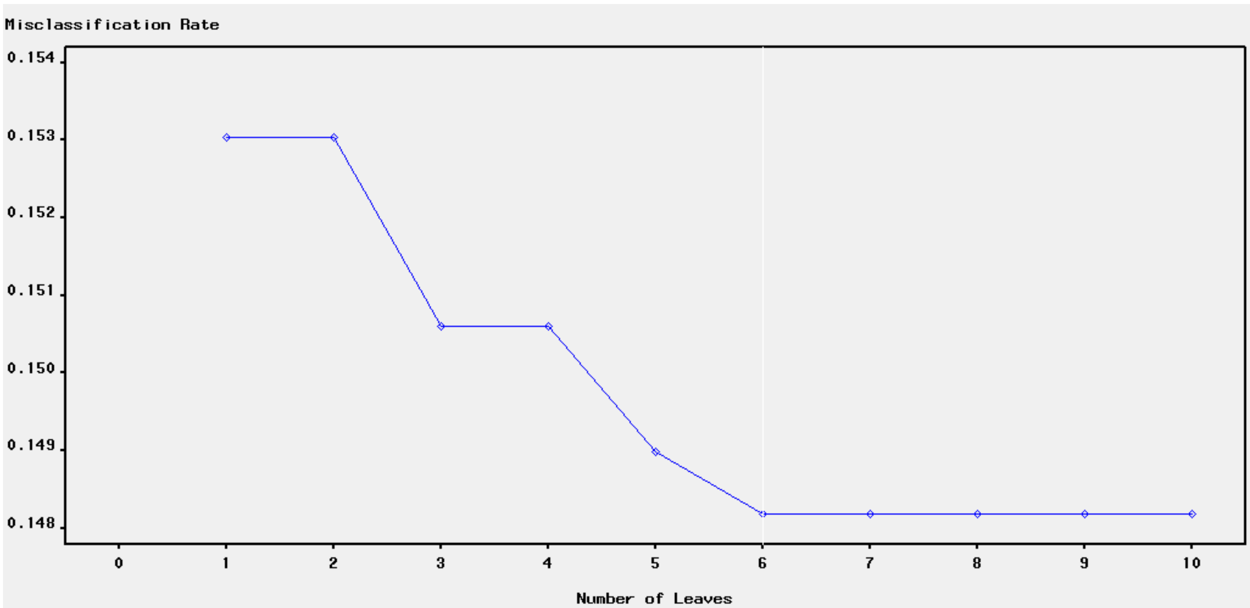

Figure 10. Misclassification Results for Higher Branches Tree

4.1.1. The Optimal Model for Heavy Winter

The following is the selected decision tree model as the optimal result for the heavy winter model, since it has minimum number of branches as two, minimum misclassification rate as less than 1%. The V_27 represents the voltage angle for Tesla. The voltage angles that are less than 20.5 degree for this location is considered as one case (non island), and voltage angles greater than or equal to 20.5 degree represents zero case (island). In each block the second column is for training data and the third column is for out of sampling data. 850 cases for out of sample data are used for this model. There are total of 1276 cases for training the decision tree. The first and second rows show the percentage of data for zero and one cases. The third and fourth rows show the numerical cases for zero and one cases, and final row show the total number of cases for training and validation.

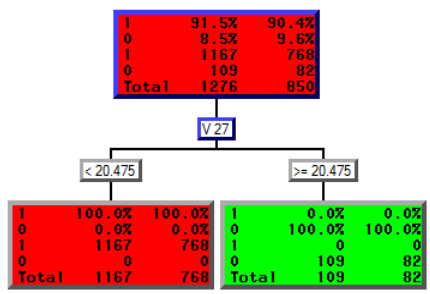

Figure 11: Decision Tree, Heavy Winter

The program generates the following plot based on the result obtained for decision tree model. The plot displays a plot of the misclassification rate on the vertical axis for the different subtrees. The tree node automatically chooses the subtree that optimizes the model assessment. The blue line represents the training data. It shows that the final numbers of the leaf are 2, with less than 1% misclassification rate. The misclassification rate for the heavy winter model is less than 1% as shown in Figure 12.

Figure 12: Misclassification Rate for Optimal Model for Heavy Winter

4.1.2. Performance Assessment: Heavy Winter Model

The out of sample data with variation in the operating points up to 6% for each area in the model is used for a final unbiased assessment of the classification. A reduced sample size can severely damage the decision tree classification. Computer-intensive methods, such as cross validation are used for both data fitting and assessment For this reason in each area the load is increased or decreased by 6%. Moreover, in order to generate more cases the load is increased or decreased for more than one area. Different combinations of changing load for one area or more than one area at the time are considered. This method generates some impractical cases, which account for unpredicted events in the system.

A plot of the estimated misclassification rate is shown in Figure 13. The Enterprise Miner program generates the plot in Figure 13 based on the result obtained for decision tree model. The plot displays a plot of the assessment values on the vertical axis for the different subtrees. The tree node automatically chooses the subtree that optimizes the model assessment. The blue line represents the training data. The red line represents the out of sample data. It shows that the final numbers of the leaf are 2, with less than %1 misclassification rate. It also shows that training data and out of sampling data have a close correlation.

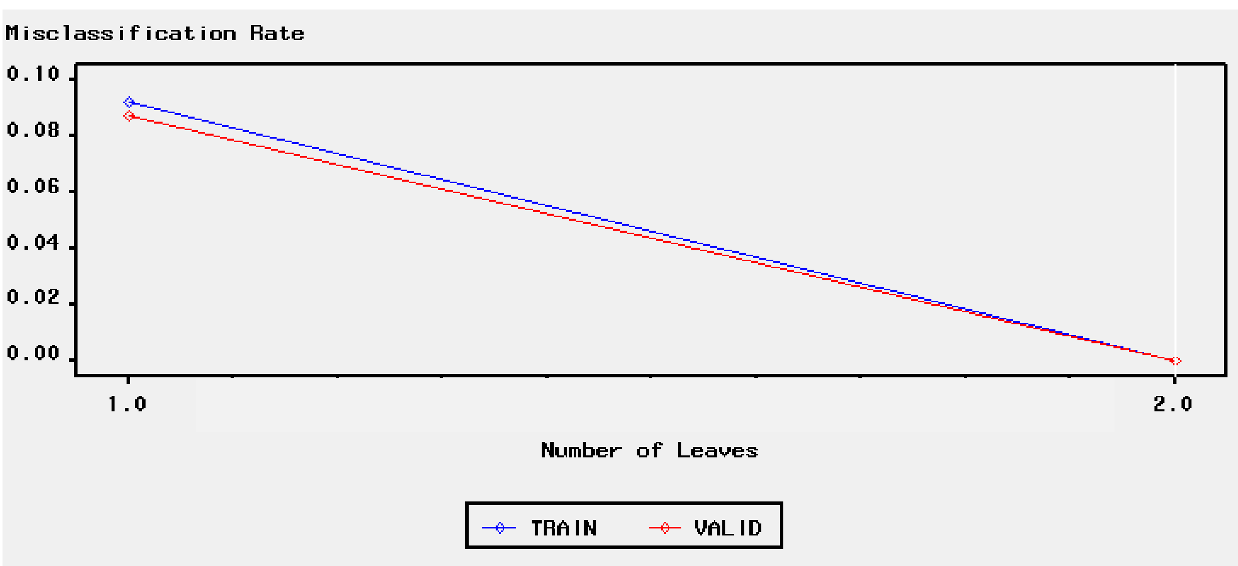

Figure 13: Data Validation for out of Sample

In addition, the analysis explores a secondary location of PMU in order to increase the robustness of the model. As shown in Table 2 the next bus to observe is Tracy. This figure shows the splitting rules and a measure of logworth, which is defined as how well a variable splits the initial data. The measure of logworth is calculated by the Enterprise Miner program and indicates how well a variable divides the data into each class. Good splitting variables have large values of logworth. The best location in the model is V_27 (Tesla) and second location is V_26 (Tracy).

Variable	Logworth	Groups	Label
V_27	107.374	2	V 27
V_26	107.355	2	V 26

Table 2: Secondary Split to Increase Reliability

The table 3 is extracted from the program and summarizes the splitting rules. The splitting rule lists the lower and upper splitting criteria. For V_26 (Tracy) the voltage angle that splits the data between zero and one is 20.45 degree.

V_26 Splitting Rule

Range	Lower	Upper
1.	...	20.45
2.	20.45	...

Table 3: Splitting Rules

Figure 14 shows the misclassification rate for secondary location in the heavy winter model which is selected as Tesla is less than 1%. There is close correlation between training and out of sampling data.

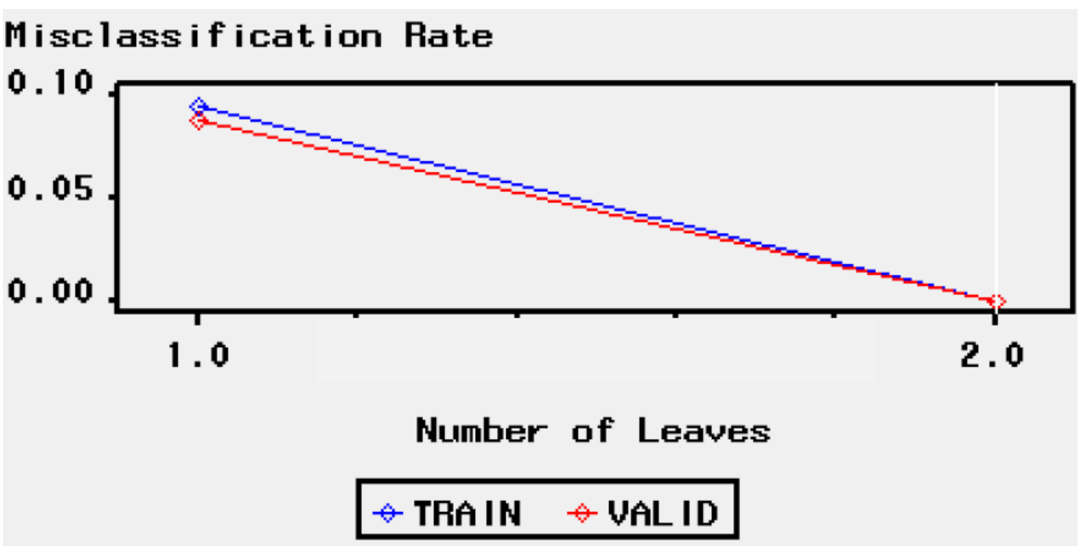

Figure 14: Misclassification Rate for HW Second Location

It is desired to test the performance of the decision tree by generating various out of sample data. Figure 15 shows the classification results for the cases generated by 6% variation in the operating condition of the heavy winter model. At total of 851 cases were generated. The error rate is less than 1%, which shows outstanding performance by the decision tree.

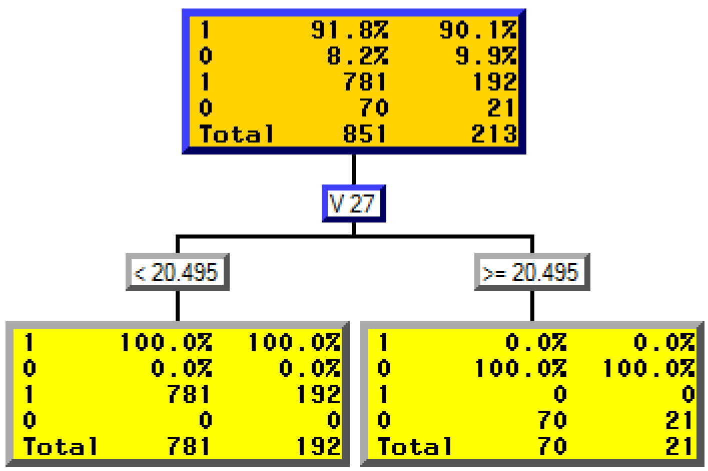

Figure 15: Decision Tree with 6% variation in operating condition

Out of sample data with variation of up to 6% in the operating points for each area in the model is used for a final unbiased assessment of the classification. The misclassification rate for both training data and out of sampling data is less than 1%. This is shown in Figure 16, where results for both training data and out of sampling data are aligned perfectly.

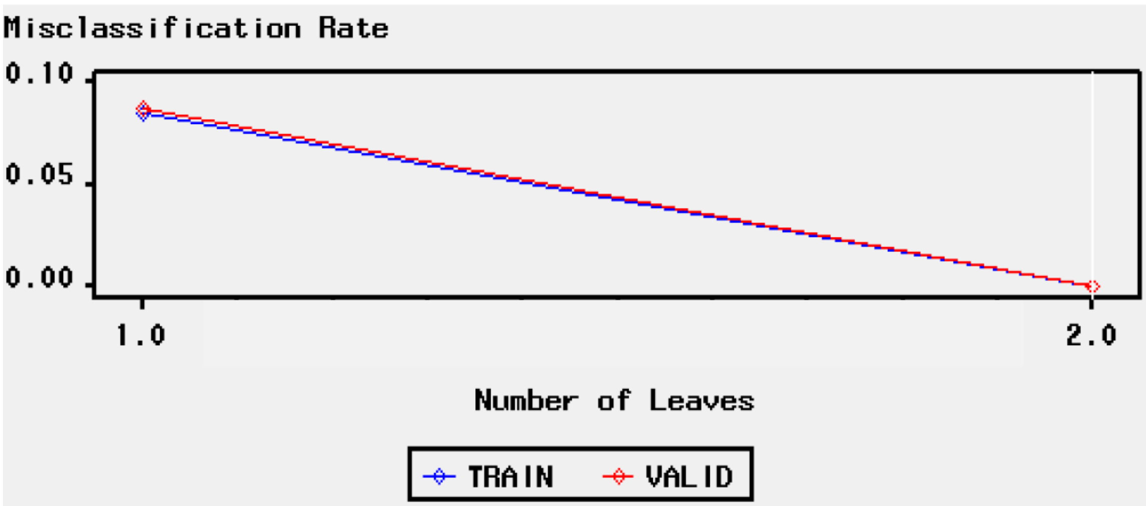

Figure 16: Misclassification Rate for 6% Variation in out of Sample Data

4.1.3. Heavy Summer Model: Decision Tree

The decision tree classification for the heavy summer model is created as well. As in the previous model, it is assumed that PMUs are located on all 500 Kv buses in the system. In order to have variation in the training cases, various operating conditions were generated in the same form as for the Heavy Winter model.

Figure 17 shows the simulation result for one of the islanding cases by having a bus fault on the Midway bus in the heavy summer model.

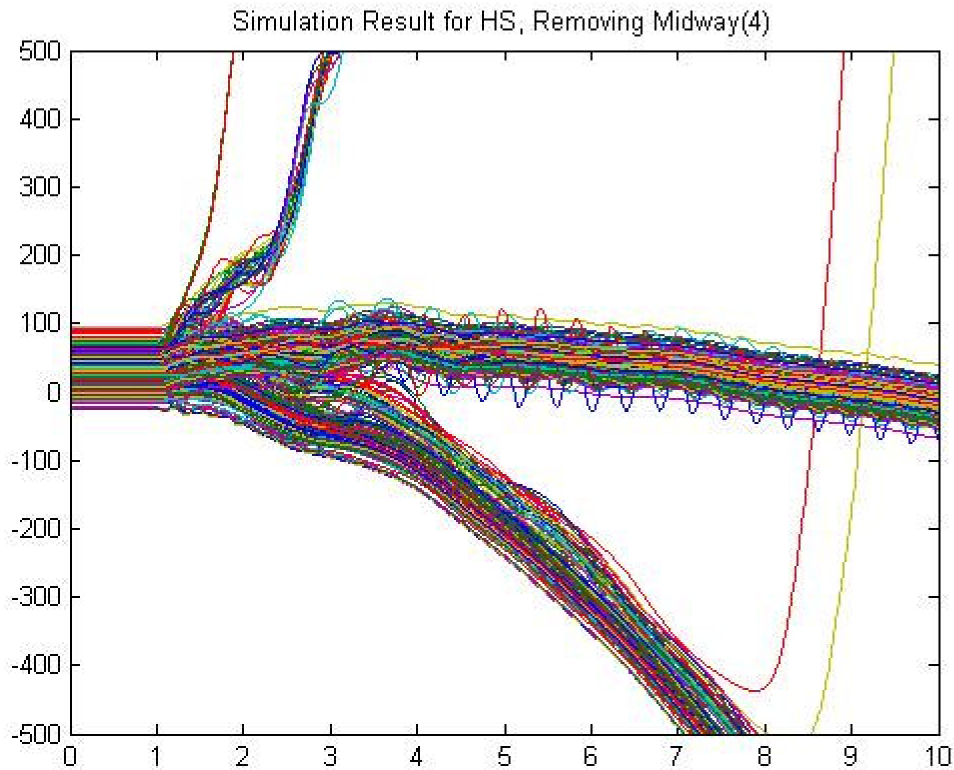

Figure 17: Islanding Case by Bus Fault for Heavy Summer Model

Figure 18 shows the decision tree derived for the heavy summer model. The recommended PMU location based on the result for the heavy summer model is Diablo. Figure 18 shows that data split for Va42 (Substation Diablo bus voltage). The voltage angles that are less than or equal -14.23 degree for this location are considered as one case (non island), and voltage angles greater than -14.23 degree represent zero cases (island). In each block the second column is for training data and the third column is for out of sampling data. 348 cases for out of sample data are used for this model. There are total of 1392 cases for training the decision tree. The first and second rows show the percentage of data for zero and one cases. The third and fourth rows show the numerical cases for zero and one cases, and final row show the total number of cases for training and validation.

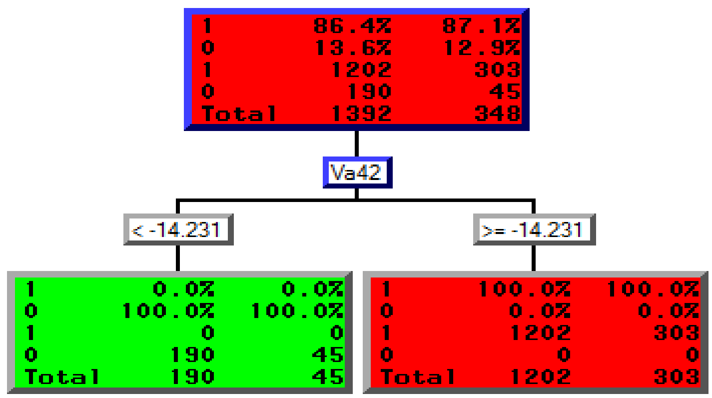

Figure 18: Decision Tree, Heavy Summer

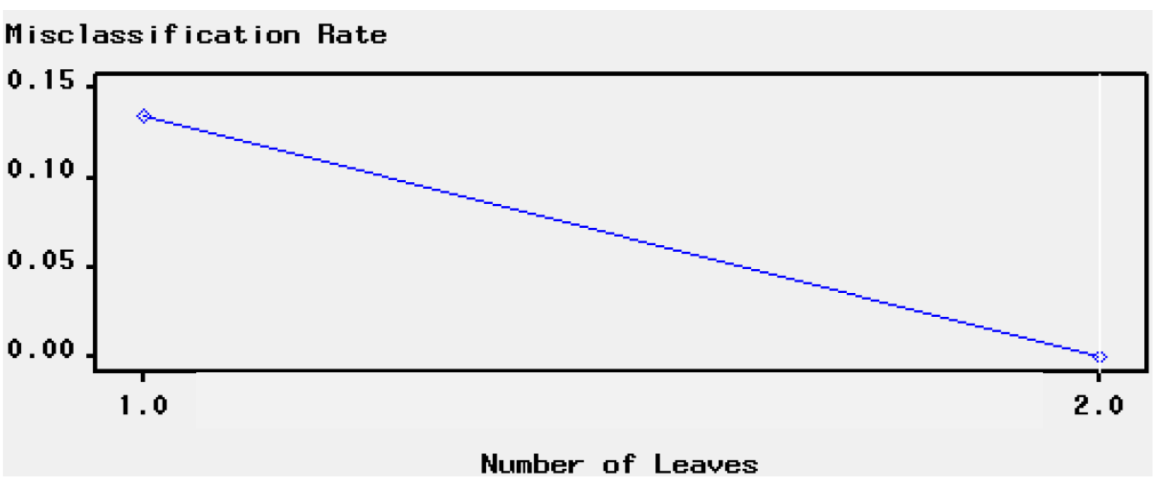

Figure 19: Data Validation for Heavy Summer Model

The best split of the tree is the one that reduces the misclassification rate the most with less number of branches possible. The above split was selected as final result since the tree has one node with misclassification rate of less than %1 for both training and validation data.

Figure 20 shows the model test with different number of training cases. As shown the model has high accuracy that changing the number of cases does not affect the splitting location or accuracy of the results. Figure 20 shows the simulation results for total of 1044 cases. It is observe that still has the same PMU location.

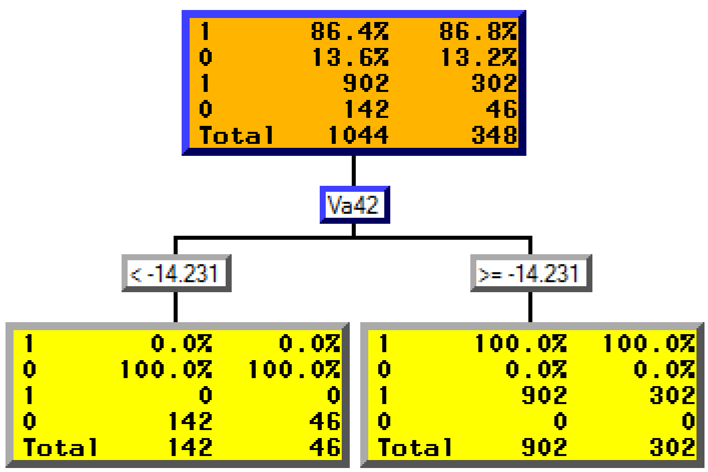

Figure 20: Simulation Result for Heavy Summer Model with 1044 Cases

Figure 21 shows the misclassification rate for total of 1044 cases with high accuracy of less than 1% for both training and out of sampling data.

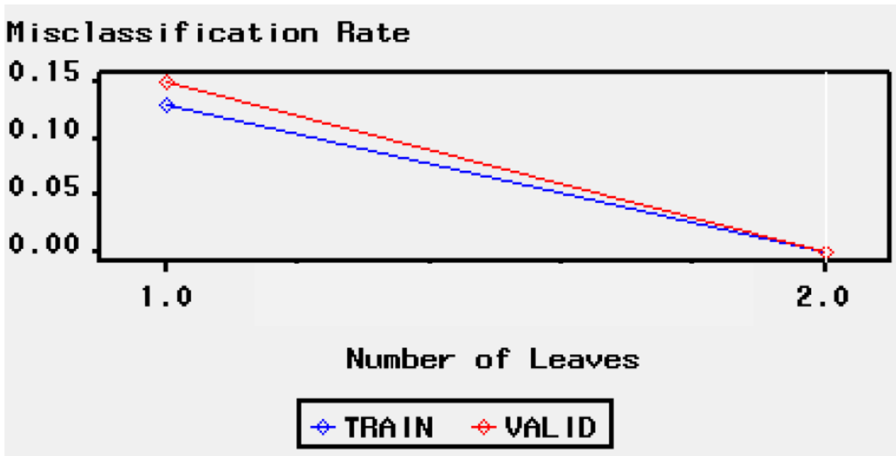

Figure 21: Misclassification Rate for the Heavy Summer Model with 1044 Cases

4.1.4. Performance Assessment: Heavy Summer Model

The 6% operating point variations performed for the heavy winter model were also performed in the Heavy summer model. These variations for each area are used for a final unbiased assessment of the classification for the heavy summer model.

A plot of the estimated misclassification rate is shown in Figure 22. The estimator used is known as cross-validation as discussed in the previous chapter.

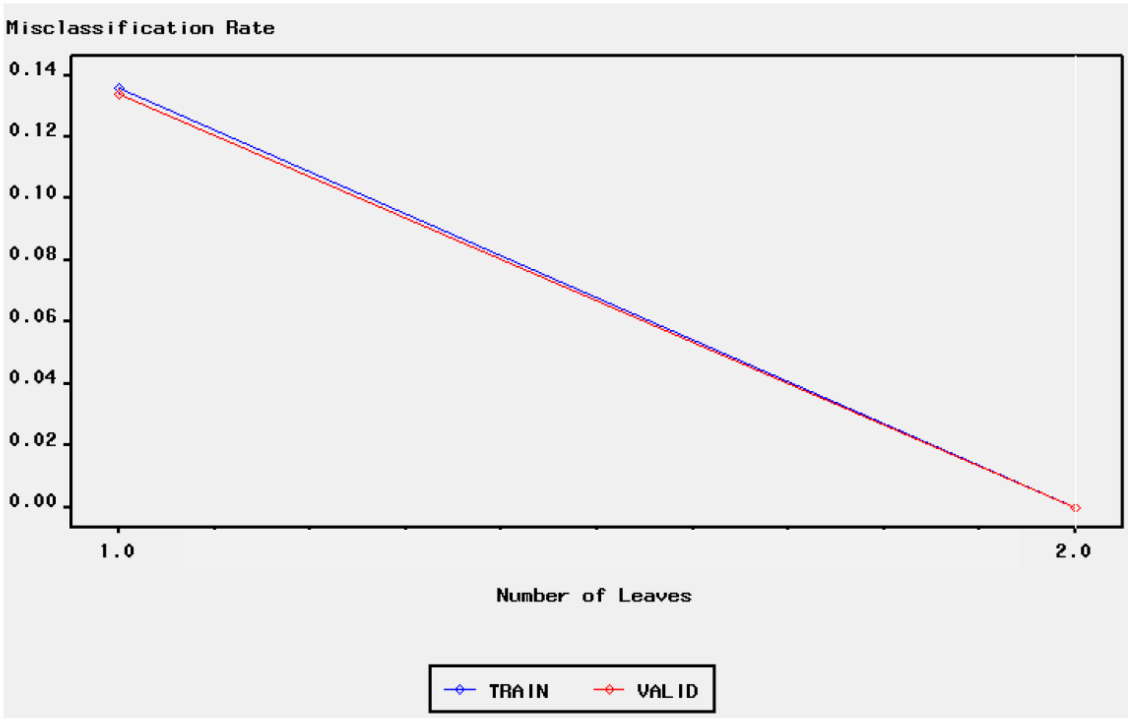

Figure 22: The Training Data and Validation Misclassification Rate

The error rate is less than 1%, which shows outstanding performance by the decision tree. It is desired to test the performance of the decision tree by generating various out of sample data.

Additional analysis was performed to investigate the secondary location of PMU in order to increase robustness of the model and reliability of the detection system. Table 4 shows the splitting rules and a measure of logworth, which is how well a variable splits in the initial data. The measure of logworth indicates how well a variable divides the data into each class. Good splitting variables have large values of logworth. The best location in the heavy summer model is VA_42 (Diablo) and second location is VA_43 (Midway) with highest logworth values.

Variable	Logworth	Groups	Label
VA42	151.541	2	Va42
VA43	151.541	2	Va43

Table 4: Logworth Values for Diablo and Midway

Table 5 summarizes the splitting rule for heavy summer decision tree classification. The figure below is extracted from the program and summarizes the splitting rules. The splitting rule lists the lower and upper splitting criteria. For VA_42 (Diablo) the voltage angle that splits the data between zero and one is -14.231degree.

VA42 Splitting Rule

Range	Lower	Upper
1.	...	-14.231
2.	-14.231	...

Table 5: Splitting Rule for Diablo

4.2. Neural Network Simulation

The SAS Enterprise Miner analysis tool is also used for data mining modeling as neural network. This software tool is available through Virginia Tech program for statistical analysis and research.

A properly trained neural network model can predict the association between input variables and target variables. The input variables are the same for both decision tree model and neural network model.

The basic neural network model is made of neurons in different layers. The first layer is the input layer, and the final layer is the target layer. All the layers between input and target are defined as hidden layer. This is the art of the designer to train a model with the appropriate layer for prediction with the highest accuracy and minimum cost.

The simple block diagram of the model is shown in the next figure, the lines between the block represent the formula that connect the input layer, hidden layer, and output layer.

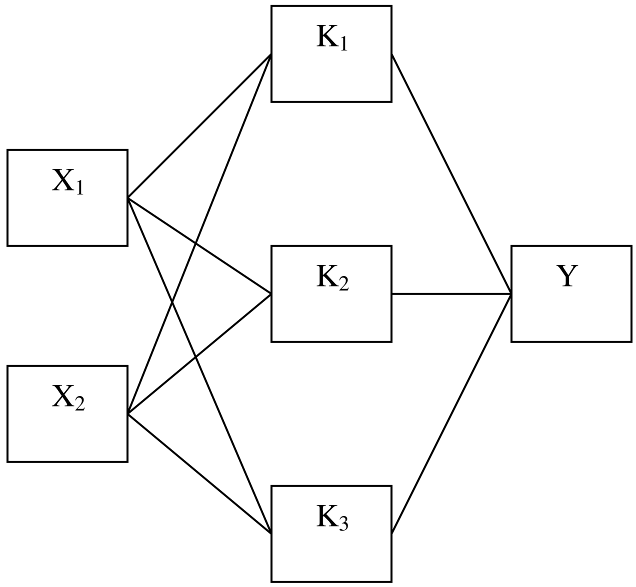

Figure 23: Simple Block Diagram for Neural Network Model

The weight estimate in the model is found by maximizing the log likelihood function, since the target for the purpose of this work is binary. After the prediction formula is established, one can perform the prediction, by entering the input measurement into the hidden unit expression.

After the prediction formula is established, one can perform the prediction, by entering the input measurement into the hidden unit expression.

Neural network needs a complete record for estimation. One of the advantages of neural network modeling compared to other modeling is the powerful linear and nonlinear association of the input and target values; at the same time the model creation is not a simple task, which has made it one of the challenging modeling tools.

4.2.1. Neural Network Training Results

The cases that were generated for decision tree are used in this modeling. One of the positive features of neural network, that makes it a better candidate than regression prediction, is that it is able to model any association between input and target value if the model is trained properly. This is the main advantage for control islanding application that eliminates the PMU relocation. Since flexibility always has its own price, the neural network does not address input selection, as other methods like decision trees do. This is not a disadvantage for this work since input selection is performed based on characteristics of the parameters in the power system, and the inputs are the same as for the decision tree model.

The voltage angles at 500 KV buses in the California heavy winter model are used as inputs to the network model, since the work in the previous sections shows that voltage angle is the best prediction parameter. Also it is desired to compare the performance for both models with similar parameters.

Two classes are defined here to the network as zero and one. The network predicts which class is most representative of the unknown inputs to the system. The multi-layer perceptron with 2 hidden layers of nodes between input and output is the best result of the model for this study. These additional layers compose of hidden units or nodes that are not directly connected to the input or output layer.

Time was dedicated through training of the network to come with the optimal number of neurons and layers for this study. The goal is to design the less complex model with the least possible misclassification rate. The different combination for layers and neurons are selected and the model that gives the less misclassification rate and less possible number of hidden layers and neuron is selected as the optimal result. Each time after choosing the number of neurons and hidden layers for each model, the training is automated through Enterprise Miner data mining program, but each model needs to be visually inspected by looking at misclassification rate for training data and out of sampling data to evaluate if that is a satisfactory rate. The following shows one of the design models with two hidden layers. The first column presents name for each 500 KV bus as Va1…..Va31. The weight column presents the weight assign to each location. The weights assign to all 500 KV buses are shown in Appendix B.

	From	To	Weight
1	Va1	H11	-0.005903848
2	Va10	H11	-0.01373444
3	Va11	H11	-0.03344635
4	Va12	H11	-0.03052996
5	Va13	H11	0.0101005637
6	Va14	H11	0.068593065
7	Va15	H11	-0.208951218
8	Va16	H11	0.1087623147
9	Va17	H11	0.0785984738
10	Va18	H11	0.0241356846
11	Va19	H11	-0.198294759
12	Va2	H11	0.0616575019
13	Va20	H11	-0.088486035
14	Va21	H11	0.2333950798
15	Va22	H11	0.0043153729
16	Va23	H11	0.1324124675
17	Va24	H11	0.0830817845
18	Va25	H11	0.0417588929
19	Va26	H11	0.1171083988
20	Va27	H11	0.1000123596
21	Va28	H11	0.0249131043
22	Va29	H11	0.0854203903
23	Va3	H11	-0.141589958
24	Va30	H11	0.0673830149
25	Va31	H11	0.2023163801

Table 6: Weight Model for 500 KV Bus 1 through 31

4.2.2. Neural Network Performance Improvement

The neural network performance improves by increasing the number of hidden layers. The statistic results show the model performance with both the average square error and misclassification scale. The value that is important for this modeling accuracy evaluation is the misclassification rate that shows a value of less than 4% for training data, and close to 7.3% for out of sampling data.

Table 8 shows the results for misclassification rates for training data and out of sampling data for different numbers of hidden layers. Different combinations of layers were tested to determine the optimal number of neurons and layers based on the misclassification rates obtained for training and validation data. Table 7 shows the number of neurons in each hidden layer for different models. Since adding more hidden layers has no effect in the misclassification rate after two hidden layers, the optimal model is selected as two hidden layers model with a misclassification rate of less than 5% for training data and 7.8% for out of sampling data. The misclassification rate for training data can be achieved with lower values but the little change in the misclassification rate of the validation data indicates an overtraining to the data for higher number of hidden layers.

Number of Hidden layers	Misclassification rate for Training Data	Misclassification rate for Validation Data
1	5.3%	8.1%
2	4.6%	7.8%
3	4.6 %	7.3%
4	2.5%	7.3%
5	1.2%	7.3%
6	Less than 1%	7.3%

Table 7: Misclassification Rate for Different Layers

The number of neurons in each layer for acceptable models is shown in Table 8. The optimal model is selected as two hidden layers model with a misclassification rate of less than 5% for training data and 7.8% for out of sampling data.

Number of Hidden layers	Number of Neurons
1	52-6-1
2	52-4-4-1

Table 8: Number of Neurons for each Hidden Layer

4.2.3. Neural Network Optimal Model

The optimal model has a misclassification rate of less than 3% for the training data. The out of sampling data has misclassification rate of about 7.3%. The locations for both heavy winter model and heavy summer model are same as shown in Table 9:

PMU Locations	Assign weight in the Model
Diablo	**0.23**
Midway	0.20
Devers	0.13
Tesla	0.10
Tracy	0.11

Table 9: PMU locations based on the Neural Network Results

5. CONCLUSIONS

This work presents an approach for the use of the decision trees and neural networks for online detection of system islanding. The simulation results show that the voltage phase angles are the best indicators for the decision tree modeling.

5.1. Contributions

The proposed methodology is that PMUs at a few locations provide enough information for islanding detection. The underlying hypothesis is that phasor measurements at strategic buses provide enough information to predict system islanding.

The simulation results show that the decision tree modeling is highly dependent on the data provided for the model. If the model changes new locations for PMUs will be required. One of the advantages of neural network modeling over decision tree modeling is the powerful linear and nonlinear association of the input and target values; which it is made the model more flexible to adapt itself to changes in data. This model will have higher tolerance to changes in the provided data for the training. On the other hand, one needs to consider that the modeling creation is not a simple task, and does not have a graphical representation as decision tree model. The results show that the neural network model has a misclassification rate less than 5% for training data, and 7.8% for out of sampling data, compared to decision trees where the misclassification rate is close to zero for both training data and out of sampling data.

The goal is to design a classifier with low complexity and with a low misclassification rate. The comparison between both schemes shows that the optimized scheme for training data performs for decision trees and for neural network. The main advantage for neural network is the model flexibility, which is especially useful for real time modeling. Moreover, changes in the model that occur through time will not require new installations of PMUs. The neural networks allow the use of fixed location for PMUs but at the expense of a lower performance.

The proposed correctly detects instability for islanding prediction from the simulations for approximately 99% of the times for decision tree and about 96% for neural network model.. The neural network method offers an improvement over current practices since it has the ability to adapt itself to new data on line and eliminates the need for new installation of PMUs after changes in the power system network,

The main contributions of this work are:

- A systematic methodology developed to detect islanding on-line with high accuracy based on decision trees classification
    - The best parameter concluded as voltage angles for this approach
- A systematic methodology was developed to detect islanding based on neural network classification using voltage angles as parameters

- The final recommendation is to use the decision trees modeling in case of enough phasor measurement, and neural network modeling with limited phasor measurement

5.2. Future Work

The continuation of this work may be to optimize techniques to obtain optimal formation of islands by considering the dynamic performance of every island in the system. The basis for islanding is highly reliant on the nature of the utility. Usually, the creation of islands is based on the physical location of the generators, but there are other factors that may be considered in the island creation, which can improve the design. The neural network and decision trees modeling can be used to make the optimal decision for the system islanding, by considering other factors such as the dynamic performance of each island in opposition to different faults in the power system.

Another extension of the work presented in this dissertation would be to investigate if other power system attributes such voltage magnitudes, current magnitude, and current angle have better potential as predictors for neural network modeling. This may result in better misclassification rate for neural network model.

Finally, a simple extension of the work presented in this dissertation would be to explore the threshold for neural network results consistency within the power system model. This may be achieved by changing different parameters as input to the neural network model or varying the power system model parameters.

APPENDIX A

A.1 PSLF Program

The algorithms for PSLF Program to generate generator angles of all generators in the model that are active after injecting fault and clearing faults are presented as follow. The code is open source for educational purposes.

```
/* Removing Vincent Bus with external # 24156 (Case 968) and record angle in rotor txt
file for
/* all generators in the California Heavy Winter model, and clearing fault at 5 sec*/
/* Use an inurn EPCL program (info1)


/*---------------------------------------------*/
/* Initializations and load base case            */
/*---------------------------------------------*/
/* dim *path[1][100]
*path[0] = "C:\Documents and Settings\Andrew\Desktop\out of step"
@ret = change_dir(*path[0])                              /* Change Path */


$base = "ca11.sav"                                       /* Load-Flow Data under
different seasonal conditions */
$basedy = "ca.dyd"                                       /* Dynamic Data
         */
$outchan = "pslf.chf"                                    /* Output Channel
         */
$initrep = "pslf.rep"                                    /* Init Report
         */
$dynrep = "pslf2.rep"                                    /* Dynamic Report
         */
$inrun = "info1.p"                                       /* Inrun EPCL to output data
*/
$output = "output.txt"
$angle = "angle.txt"
$filechan1="channels1.txt"
$filerot="rot.txt"

mailbox[0].string[1]=$filerot
```

```
mailbox[0].string[5]=$filechan1
mailbox[0].number[2]=1/*Used for info loop (print every n time steps)*/
/*---------------------------------------------*/
/* Dynamic Simulation Variables               */
/*---------------------------------------------*/

        @first=1.0                                    /* fault on generator bus */
        @trip=5.0                                    /* tripping generator     */
        @final=10                                    /* Final Tie Line flow    */

        @ret = getf($base)



        @ret = buildeff(0)
        @ret = flowcalc(1)
        dispar[0].noprint = 0


      @ret = psds()                                   /* MUST be included to setup
modlib   */
      @ret = rdyd($basedy,$dynrep,1,1,1)              /* All flags 1 */

/**********************************************************************
*********/
  /*    Change Record Levels & Print Name of Channels I Need   [GENEROR
ANGLES]     */

/**********************************************************************
*********/
  for @mod=0 to dypar[0].nmodels-1     /*Scanning all models*/
  @ind_mlib = model[@mod].mod_lib_no
  if (modlib[@ind_mlib].name!="monit")
  model[@mod].rec_level=0             /*Set record level to 0 for all models*/
  endif
  next

  @chanrot=0
  @plimit=5
  @return = openlog($filerot)
  @return = openlog($filechan1)
```

```
logprint($filerot,"channels","<")

for @mod=0 to dypar[0].nmodels-1     /*scanning all models*/
@ind_mlib = model[@mod].mod_lib_no   /*Index to modlib library*/
@ind_gnbc=model[@mod].k              /*Index number for GENBC*/
@ind_gns=genbc[@ind_gnbc].kgen       /*Index number for GENS*/
if (model[@mod].st=1)
if (modlib[@ind_mlib].type="g")
if (modlib[@ind_mlib].name!="motor1")
if (modlib[@ind_mlib].name!="genwri")
if (modlib[@ind_mlib].name!="gewtg")
if (gens[@ind_gns].st=1)
if (gens[@ind_gns].pgen >= @plimit)
@bindex = gens[@ind_gns].ibgen


logprint($filerot,busd[@bindex].extnum:5:0,",",model[@mod].id:2,",",busd[@bindex].bu
snam:8,",",gens[@ind_gns].pgen:6:1,",",busd[@bindex].area,"<")
logprint($filechan1,@ind_gnbc,"<")

@chanrot=@chanrot+1

endif
endif
endif
endif
endif
endif
endif
next

logprint($filerot,"info1")

mailbox[0].number[1]=@chanrot /*Number Of Channels for [ROTOR ANGLES] */

@ret=close($filerot)
@ret=close($filechan1)
```

```
/***********************************************************************
**********/
  /*                    Initialize Dynamic Simulation                        */

/***********************************************************************
**********/

  dypar[0].run_epcl="info1.p"
  @ret = init($outchan,$initrep,"1","0")

     /*dypar[0].run_epcl = $inrun                          /* Name of the EPCL to In-Run */
                                                  /* Initialize Dynamic Simulation */

/***********************************************************************
**********/
   /*                   Set Parameters for Run                        */

/***********************************************************************
**********/

     dypar[0].tpause = @first                    /* Stop right before applying fault*/

     dypar[0].nscreen = 999999                           /* Dont Print to Screen (makes
simulation run faster */
     dypar[0].nplot = 999999
     @ret = run()

/***********************************************************************
**********/
   /*                  line fault                    */

/***********************************************************************
**********/
```

```
/*Apply a fault for X cycles*/
/*dypar[0].itfymx=1000*/
/*dypar[0].tolfy=0.001*/
dypar[0].delt=.00208330         /*Time Step*/
dypar[0].nplot=1                /*Record data every n time steps*/
dypar[0].nscreen=500            /*Show data on screen every n time steps*/
dypar[0].tpause=1+0.016666667*10  /*Fault span  (tpause-1sec) */
dypar[0].faultloc="24156"

/*Faulted Bus*/
dypar[0].faulton=1              /*Turn on fault*/

@ret=run()                      /*Run simulation*/


dypar[0].faulton=0              /*Turn off fault*/
```

```
/**********************************************************************
**********/
        /*      Removing Vincent Bus, and Transformer connected to it     (Case 968)
*/

/**********************************************************************
**********/


secdd[3442].st=0                /*Remove line connect to Vincent Bus*/
secdd[3443].st=0
secdd[3857].st=0
secdd[3858].st=0
secdd[3859].st=0
secdd[3860].st=0
secdd[3861].st=0
secdd[3862].st=0
```

```
        secdd[3863].st=0
        secdd[3864].st=0
        secdd[3865].st=0
        secdd[3866].st=0

        tran[1256].st=0
        tran[1257].st=0
        tran[1258].st=0
        tran[1259].st=0

/**********************************************************************
**********/
   /*                   Run the program                    */

/**********************************************************************
**********/

       dypar[0].tpause = @final
       dypar[0].nscreen = 999999                    /* Dont Print to Screen (makes
simulation run faster) */
       dypar[0].nplot = 999999
       @ret = run()
       @ret = dsst()


/*@ret = close ( $output )*/

End


/**********************************************************************
*********/ The inrun Program


/*                          Print Info into the file                         */
/**********************************************************************
*********/
if ((dypar[0].time>=0) and (mailbox[0].number[2]=1))

$rotor="rotor.txt"
$fcrot=mailbox[0].string[5]
```

```
@tot1=mailbox[0].number[1]


@ret=setinput($fcrot)

@ret=setlog($rotor)

logprint($rotor,"<")

logprint($rotor,dypar[0].time,"<")



@counter=0

for @lup=1 to @tot1                    /*<------Begining of for loop [ROTOR]------> */

@ret=input($fcrot,@ind_gnbc)

logprint($rotor,genbc[@ind_gnbc].angle:12:2," ")
@counter=@counter+1

if(@counter=10)
logprint($rotor,"<")
@counter=0
endif

next                                   /*<------End of for loop  [ROTOR]------------> */

@ret=close($rotor)
@ret=close($fcrot)



mailbox[0].number[2]=2                 /*<------print 4 times per second------------> */


elseif(mailbox[0].number[2]=2)
```

```
mailbox[0].number[2]=3

elseif(mailbox[0].number[2]=3)

mailbox[0].number[2]=4

elseif(mailbox[0].number[2]=4)

mailbox[0].number[2]=1



endif
end


*
/* Trip one generator and read angle for
/* other generators in the California Heavy Summer case */
/* An inrun EPCL program, output.p, is used to output results to a text file */


/*----------------------------------------------*/
/* Initializations and load base case              */
/*----------------------------------------------*/
/* dim *path[1][100]
*path[0] = "C:\Documents and Settings\Andrew\Desktop\out of step"
@ret = change_dir(*path[0])                                   /* Change Path */


$base = "cahse(6).sav"                                                 /* Load-Flow Data
under different seasonal conditions */
$basedy = "cahs_1.dyd"                                                 /* Dynamic Data
                    */
$outchan = "pslf.chf"                                         /* Output Channel
          */
$initrep = "pslf.rep"                                         /* Init Report
          */
$dynrep = "pslf2.rep"                                         /* Dynamic Report
          */
```

```
$inrun = "info.p"                                    /* Inrun EPCL to output data
*/
$output = "output.txt"
$angle = "angle.txt"
/*--------------------------------------------*/
/* Dynamic Simulation Variables              */
/*--------------------------------------------*/

@first=1.0                                          /* fault on generator bus */
@trip=5.0                                           /* tripping generator     */
@final=20                                           /* Final Tie Line flow    */

@ret = getf($base)



   @ret = buildeff(0)
   @ret = flowcalc(1)
   dispar[0].noprint = 0
   /*@ret[0].number[499]= @i                  /*Generator Index passed as a global
variable*/



     @ret = psds()                              /* MUST be included to setup
modlib   */
     @ret = rdyd($basedy,$dynrep,1,1,1)         /* All flags 1 */

     /*@gbus = gens[@i].ibgen
        /*logprint($output, busd[@gbus].extnum, ">", busd[@gbus].busname, ">",
busd[@gbus].basekv, ">", gens[@i].pgen, "<")*/

     for @mod=0 to dypar[0].nmodels-1          /*Scanning all models*/
       @ind_mlib = model[@mod].mod_lib_no
       if (modlib[@ind_mlib].name!="monit")
         model[@mod].rec_level=0               /*Set record level to 0 for all models*/
       endif
     next

     dypar[0].run_epcl = $inrun                 /* Name of the EPCL to In-Run */
                                                /* Initialize Dynamic Simulation */
```

```
    @ret = init($outchan,$initrep,"1","0")          /* Output Channel: firstdyn.chf */
                                                   /* "1" Fix Bad Data, "0" dont turn off
unused models */
    /* Set Parameters for Run */                    /*Run to @tfault, remove caps,
apply fault, run to @clrfault, remove F and HF lines, run to tend*/
    dypar[0].tpause = @first                       /* Stop right before applying fault*/

    dypar[0].nscreen = 999999                       /* Dont Print to Screen (makes
simulation run faster */
    dypar[0].nplot = 999999
    /*dypar[0].nchan = 20*/
    @ret = run()

    /* Trip Generator */

     gens[425].st=0

    /*After Generator Tripping*/
    dypar[0].tpause = @final
    dypar[0].nscreen = 999999                       /* Dont Print to Screen (makes
simulation run faster */
    dypar[0].nplot = 999999
    @ret = run()
    @ret = dsst()



/*@ret = close ( $output )*/
End
```

APPENDIX B

B.1 Neural network simulation results for 500 KV buses weight

Tables 10, and 11 are showing the weights assigned to each of 500 KV bus locations. This model is not selected since it has higher misclassification rate than selected model as final result.

	From	To	Weight
1	Va1	H11	-0.005903848
2	Va10	H11	-0.01373444
3	Va11	H11	-0.03344635
4	Va12	H11	-0.03052996
5	Va13	H11	0.0101005637
6	Va14	H11	0.068593065
7	Va15	H11	-0.208951218
8	Va16	H11	0.1087623147
9	Va17	H11	0.0785984738
10	Va18	H11	0.0241356846
11	Va19	H11	-0.198294759
12	Va2	H11	0.0616575019
13	Va20	H11	-0.088486035
14	Va21	H11	0.2333950798
15	Va22	H11	0.0043153729
16	Va23	H11	0.1324124675
17	Va24	H11	0.0830817845
18	Va25	H11	0.0417588929
19	Va26	H11	0.1171083988
20	Va27	H11	0.1000123596
21	Va28	H11	0.0249131043
22	Va29	H11	0.0854203903
23	Va3	H11	-0.141589958
24	Va30	H11	0.0673830149
25	Va31	H11	0.2023163801

Table 10: Weight Model for 500 KV Buses 1 through 31

	From	To	Weight
26	Va32	H11	0.2530721616
27	Va33	H11	-0.155329567
28	Va34	H11	-0.213787584
29	Va35	H11	-0.189229402
30	Va36	H11	-0.152872175
31	Va37	H11	0.0503238497
32	Va38	H11	-0.153249634
33	Va39	H11	0.0695039121
34	Va4	H11	-0.004770164
35	Va40	H11	-0.062382734
36	Va41	H11	-0.106206049
37	Va42	H11	-0.108219876
38	Va43	H11	-0.070975662
39	Va44	H11	-0.071754844
40	Va45	H11	-0.07547564
41	Va46	H11	-0.142543405
42	Va47	H11	0.0793541006
43	Va48	H11	-0.106702275
44	Va49	H11	0.221929627
45	Va5	H11	0.0552508853
46	Va50	H11	-0.196675182
47	Va51	H11	-0.049420858
48	Va52	H11	0.2809281365
49	Va53	H11	0.0406268744
50	Va6	H11	-0.207182947

Table 11: Weight Model for 500 KV Buses

LITERATURE CITED

1. NYISO, *Blackout August 14, 2003 Final Report.* 2005.
2. Chapman, S.J., *Electric Machinery and Power System Fundamentals*. 2002: McGraw Hill Publishers.
3. Sun, K., D.Z. Zheng, and Q.A. Lu, *Splitting strategies for islanding operation of large-scale power systems using OBDD-based methods.* Ieee Transactions on Power Systems, 2003. **18**(2): p. 912-923.
4. El-werfelli, M., J. Brooks, and R. Dunn. *Controlled island ng scheme for power systems*. in *Universities Power Engineering Conference, 2008. UPEC 2008. 43rd International*. 2008.
5. Senroy, N. and G.T. Heydt, *A conceptual framework for the controlled islanding of interconnected power systems.* Power Systems, IEEE Transactions on, 2006. **21**(2): p. 1005-1006.
6. Kundur, P., *Power system stability and control*. 1994, New York: McGraw Hill. 1176.
7. Warin, J.W., *Loss of Mains Protection*, in *ERA Conference on Circuit Protection for Industrial and Commercial Installations*
   1990: London.
8. Funabashi, T., K. Koyanagi, and R. Yokoyama. *A review of islanding detection methods for distributed resources*. in *Power Tech Conference Proceedings, 2003 IEEE Bologna*. 2003.
9. UCTE (2006) *Final Report: System Disturbance on 4 November 2006*.
10. Phadke, A.G. and J.S. Thorp, *Synchronized phasor measurements and their applications*. 2008, Springer: New York.
11. Centeno, V., A.G. Phadke, and A. Edris. *Adaptive out-of-step relay with phasor measurement*. in *Developments in Power System Protection, Sixth International Conference on (Conf. Publ. No. 434)*. 1997.
12. Thorp, J.S., A.G. Phadke, S.H. Horowitz, and M.M. Begovic. *Some applications of phasor measurements to adaptive protection.* Power Systems, IEEE Transactions on, 1988. **3**(2): p. 791-798.
13. Phadke, A.G. and J.S. Thorp. *Communication needs for Wide Area Measurement applications*. in *Critical Infrastructure (CRIS), 2010 5th International Conference on*. 2010.
14. Kang, H., B. Cvorovic, C. Mycock, D. Tholomier, and R. Mai. *PMU simulation and application for power system stability monitoring*. in *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*. 2009.
15. De La Ree, J., V. Centeno, J.S. Thorp, A.G. Phadke. *Synchronized Phasor Measurement Applications in Power Systems.* Smart Grid, IEEE Transactions on, 2010. **1**(1): p. 20-27.

16. Bhargava, B. and A. Salazar. *Use of Synchronized Phasor Measurement system for monitoring power system stability and system dynamics in real-time*. in *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*. 2008.
17. Vanfretti, L., J.H. Chow, S. Sarawgi, D.Ellis, B. Fardanesh. *A framework for estimation of power systems based on synchronized phasor measurement data*. in *Power & Energy Society General Meeting, 2009. PES '09. IEEE*. 2009.
18. Zaborszky, J., K. Whang, G. Huang, L. Chiang, S. Lin. *A Clustered Dynamic Model for a class of LInear Autonomous Systems Unsing Simple Enumerative Sorting.* Circuits and Systems, IEEE Transactions on, 1982. **29**(11): p. 747-758.
19. Phadke, A.G., *Voltage Stability Assesment Through Measurement of a Reduced State Sector.* IEEE Trans on Power System, 1990. **5**.
20. Galvan, F., S. Mandal, and M. Thomas. *Phasor Measurement Units (PMU) instrumental in detecting and managing the electrical island created in the aftermath of Hurricane Gustav*. in *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*. 2009.
21. Ruisheng, D., V. Vittal, K. Sun, S. Kolluri, S. Mandal, F. Galvan. *Decision tree assisted controlled islanding for preventing cascading events*. in *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*. 2009.
22. Archer, B.A. and J.B. Davies. *System islanding considerations for improving power system restoration at Manitoba Hydro*. in *Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on*. 2002.
23. Mori, H. *State-of-the-Art Overview on Data Mining in Power Systems*. in *Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES*. 2006.
24. Bernabeu, E.E., *Methodology for a Security-Dependability Adaptive Protection Scheme based on Data Mining*, in *Electrical Engineering*. 2009, Virginia Tech: Blacksburg.
25. Genc, I., R. Diao, V. Vittal, S. Kolluri, S. Mandal. *Decision Tree-Based Preventive and Corrective Control Applications for Dynamic Security Enhancement in Power Systems.* Power Systems, IEEE Transactions on, 2010. **25**(3): p. 1611-1619.
26. Galushkin, A.I., *Neural Network Theory*. 2007, New York: Springer. 389.
27. Damborg, M.J., M.A. El-Sharkawi, M.E. Aggoune, R.J. Marks. *Potential of artificial neural networks in power system operation*. in *Circuits and Systems, 1990., IEEE International Symposium on*. 1990.
28. Oleskovicz, M., D.V. Coury, and A.C.P.L.F. de Carvalho. *Artificial neural network applied to power system protection*. in *Neural Networks, 1998. Proceedings. Vth Brazilian Symposium on*. 1998.
29. Amjady, N. and M. Ehsan, *Evaluation of power systems reliability by an artificial neural network.* Power Systems, IEEE Transactions on, 1999. **14**(1): p. 287-292.

30. Delimar, M., I. Pavic, and Z. Hebel. *Artificial neural networks in power system topology recognition*. in *EUROCON 2003. Computer as a Tool. The IEEE Region 8*. 2003.
31. Bretas, A.S. and A.G. Phadke, *Artificial neural networks in power system restoration.* Power Delivery, IEEE Transactions on, 2003. **18**(4): p. 1181-1186.
32. Stankovic, A.M., A.T. Saric, and M. Milosevic, *Identification of nonparametric dynamic power system equivalents with artificial neural networks.* Power Systems, IEEE Transactions on, 2003. **18**(4): p. 1478-1486.
33. Wu, M. and P. Rastgoufard. *Optimum decision by artificial neural networks for reactive power control equipment to enhance power system stability and security performance*. in *Power Engineering Society General Meeting, 2004. IEEE*. 2004.
34. Wenjin, D. and W. Ping. *Application of Pattern Recognition and Artificial Neural Network to Load Forecasting in Electric Power System*. in *Natural Computation, 2007. ICNC 2007. Third International Conference on*. 2007.
35. Sobajic, D.J. and Y.-H. Pao, *Artificial Neural-Net Based Dynamic Security Assessment for Electric Power Systems.* Power Engineering Review, IEEE, 1989. **9**(2): p. 55-55.
36. Qin, Z., J. Davidson, and A.A. Fouad, *Application of artificial neural networks in power system security and vulnerability assessment.* Power Systems, IEEE Transactions on, 1994. **9**(1): p. 525-532.
37. Jensen, C., M. El-Sharkawi, and R. Marks, *Power System Security Assessment Using Neural Networks: Feature Selection Using Fisher Discrimination.* Power Engineering Review, IEEE, 2001. **21**(10): p. 62-62.
38. Rovnyak, S., S. Kretsinger, J. Thorp, D. Brown. *Decision trees for real-time transient stability prediction.* Power Systems, IEEE Transactions on power system, 1994. **9**(3): p. 1417-1426.
39. Centeno, V., J. De La Ree, A.G. Phadke, G. Michel, R.J. Murphy, R.O. Burnett. *Adaptive out-of-step relaying using phasor measurement techniques.* Computer Applications in Power, IEEE, 1993. **6**(4): p. 12-17.
40. Elizondo, D.C., J. De La Ree, A.G. Phadke, S. Horowitz. *Hidden failures in protection systems and their impact on wide-area disturbances*. in *Power Engineering Society Winter Meeting, 2001. IEEE*. 2001.
41. Helberg, C., *Data mining with confidence*. 2 ed. 2002, Chicago, IL: SPSS, Inc.
42. Fayyad, U.M., G. Piatetsky-Shapiro, S. Padhraic, R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. 1996.
43. Breiman, L., J. Friedman, C.J. Stone, R.A. Olshen. *Classification and regression trees,*1994, California: Wadsworth and Brooks, Monterey.
44. Mitchell, T.M., *Machine Learning*. 1997: McGraw Hill.
45. Quinlan, J.R., *C4.5: programs for machine learning.* 1993: Morgan Kaufmann Publishers Inc.

46. Bishop, C.M., *Neural networks for pattern recognition*. 1995: Oxford University Press.
47. Kohavi, R. *Irrelevent Features and the Subset Selection* in *Machine Learning*. 1994. San Francisco: Morgan Kaufmann.
48. Raol, J.R., G. Girija, J. Singh,, *Modeling and Parameter Estimation of Dynamic System.* 2004, London: The Institution of Electrical Engineers.
49. Walpole, R.E., R.H. Myers, *Power System Stability and Control*. 1989, New York: Macmillan Publishing Company.
50. Livingstone, D.J., *Artificial Neural Network: Methods and Application.* 2008: Springer.
51. Lau, C., *Neural Network: Theoretical Foundations and Analysis.* 1992, New York: IEEE, Inc.
52. Kim, S.H., *Statistics and Decisions: An Introduction to Foundations*. 1992, New York: Van Nostrand Reinhold. 336.
53. *PSLF Software*. 2011; PSLF.
54. WWW.SAS.COM, *SAS® Enterprise Miner.*