Methodology for a Security-Dependability Adaptive Protection Scheme based on Data Mining

Emanuel E. Bernabeu

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

Committee Members

James S. Thorp (Co-Chair)

Virgilio A. Centeno (Co-Chair)

Jaime De la Ree Lopez

Yilu Liu

Werner E. Kholer

Luiz DaSilva

December 9th, 2009

Blacksburg, Virginia U.S.A

Keywords: critical locations, adaptive protection, wide area measurements, data mining, decision trees.

Methodology for a Security-Dependability Adaptive Protection Scheme based on Data Mining

Emanuel E. Bernabeu

Abstract

The power industry is currently in the process of re-inventing itself. The unbundling of the traditional monopolistic structure that gave birth to a deregulated electricity market, the mass tendency towards a greener use of energy, the new emphasis on distributed generation and alternative renewable resources, and new emerging technologies have revolutionized the century old industry.

Recent blackouts offer testimonies of the crucial role played by protection relays in a reliable power system. It is argued that embracing the paradigm shift of adaptive protection is a fundamental step towards a reliable power grid. The adaptive philosophy of protection systems acknowledges that relays may change their characteristics in order to tailor their operation to prevailing system conditions. The purpose of this dissertation is to present methodology to implement a security/dependability adaptive protection scheme. It is argued that the likelihood of hidden failures and potential cascading events can be significantly reduced by adjusting the security/dependability balance of protection systems to better suit prevailing system conditions.

The proposed methodology is based on Wide Area Measurements (WAMs) obtained with the aid of Phasor Measurement Units (PMUs). A Data Mining

algorithm known as Decision Trees is used to classify the power system state and to predict the optimal security/dependability bias of a critical protection scheme.

To My Parents: Hugo Bernabeu and Susana Chacon.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my academic advisors, Dr. James Thorp and Dr. Virgilio Centeno, for their patient guidance and support throughout my research. Dr. Thorp enthusiastically shares his knowledge. He has been a mentor and an invaluable source of inspiration for the past three years. Dr. Centeno took his role of advisor beyond what it is expected. I am proud to call him a mentor and a friend. He has advised me on a personal level, and for that, I will always be grateful.

My appreciation also goes to Dr. Jaime De La Ree, one of the best teachers I have had the pleasure to meet. With discussions and insightful comments he has greatly contributed to this research. My gratitude also goes to the remaining members of my advisory committee, Dr. Yilu, Dr. DaSilva, and Dr. Kholer, for their genuine interest and support. I would also like to acknowledge my friends and labmates at the Power Lab. Special thanks to Margarita for granting me the support to achieve my dreams.

Last but not least, I would like to thank my parents, Hugo and Susana, and my sisters, Julieta and Magdalena, for their love and encouragement. In spite of the distance, they are always with me in all my endeavors.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In the following sections the underlying philosophy of adaptive protection is explained. It will be argued that the proposed paradigm shift is of vital importance due to the manner in which power systems have evolved; stressed systems pushed to their limits with ever decreasing margins for errors. Recent blackouts [3, 6, 7] offer testimonies of the crucial role played by protection relays in a reliable power system.

One of the enabling forces for adaptive protection is the advancement of Phasor Measurement Units (PMUs). Throughout this dissertation an emphasis on a wide-area perspective of power systems will be made. Utilities are increasingly relying on their neighbors for the daily operation of the grid. This increased inter-dependency enhances the need for a broad view of the system. Wide-Area Measurements (WAMs) provide invaluable information of the system state and promise to revolutionize the operation, control, and protection of the power system.

In the chapter, key concepts like Security/Dependability and Hidden Failures are defined. Motivation for the adaptive protection paradigm shift is given. Finally, an overview of this dissertation main contribution, the methodology for an adaptive security/dependability voting scheme, is given. It is also argued that the peculiar characteristics of the California power grid make the application of the proposed adaptive scheme particularly attractive.

Chapter 2 presents a comprehensive review of adaptive schemes and Data Mining applications to power systems. In Chapter 3, an implementation oriented description of Decision Trees is presented. In Chapter 4, the proposed methodology is thoroughly described. Chapter 5 presents simulation results obtained from a highly accurate model of the California system. Finally, in Chapter 6, conclusions and final remarks are made.

1.1 Adaptive Philosophy of Protection Systems

The concept of adaptive relaying has been around for decades and yet very few adaptive schemes have been designed and implemented. As early as 1988, adaptive relaying was defined as the ability of relays to change their settings, operation, or logic to adapt to prevailing system conditions [8].

As defined by the Institute of Electrical and Electronic Engineers (IEEE) a protection relays is *"an electric device that is designed to respond to input conditions in a prescribed manner and, after specified conditions are met, to cause contact operation or similar abrupt change in associated electric control circuits"* [9]. Conventional relays react in a predetermined and fixed manner and are typically biased towards dependability. These dormant sentinels protect the grid with nothing more than local voltage and current measurements as their weapons to detect faults[1]. Experience shows that such rigid relay settings may become unreliable under abnormal stressed conditions. The implicit system assumptions hidden in the relay settings do not longer hold under extreme conditions, which lead to unforeseen or unwanted relay operations. In the 2003 USA/Canada blackout a total of 14 impedance relays miss-operations were reported [6]. Needless to say, this unnecessary line trips enhanced the speed of propagation of the blackout.

Protection engineers have been reluctant to accept the concept of adaptive relaying. However, it is argued in this dissertation that the rewards offered by adaptive protection easily outweigh the efforts required by the paradigm shift. Under the new protection philosophy, relays can tailor their operations to prevailing system conditions. The driving technologies that facilitate the new philosophy are digital relays and Wide Area Measurements (WAMs) obtained through Phasor Measurement Units (PMUs).

It is understood that not every protection system needs to be re-designed to be adaptive. There are, however, some *critical locations* in the power system where adaptive schemes would be extremely beneficial. A systematical analysis to discover and identify critical locations from the protection's point of view is presented in Chapter 4.

The scope of adaptive protection is intended for non-instantaneous protection. High-speed primary protection relays operate within 1 to 3 cycles (17 to 50 msec). Since the current

[1] Pilot-schemes also have information transmitted from an adjacent bus on their arsenal.

power system communication infrastructure[2] accommodates a wide range of protocols, with transfer rates that go from low 56kbits to fiber optics, primary protection based on WAMs is unfeasible. However, there is a niche for adaptive relaying in non-instantaneous protection systems. The most promising candidates are:

- **Backup protection**. Backup protection is defined as *"protection that operates independently of specified components in the primary protective system"* [9]. In general, the necessary coordination delay for backup protection is within the range of 0.3 to 1 second, which gives an adequate window for transmitting and processing WAMs information. A supervised impedance relay design based on WAMs is discussed in Chapter 2.
- **Stability related protection**. Angle, voltage and frequency instability evolve in a relatively long period of time; it may take a few seconds, minutes, or in the case of voltage instability, hours. Adaptive schemes are particularly suited for these extreme events and potentially huge rewards can be achieved by embracing the adaptive philosophy. An adaptive out-of-step and a WAMs based load shedding scheme is presented is Chapter 2.
- **Settings for group of relays**. The idea of relays autonomously changing their settings terrifies protection engineers and it is probably the main deterrent of adaptive protection. However, it is possible to alter the functionality of a group of relays without directly modifying relay settings. This is the main proposition and contribution of this dissertation. The methodology used to alter the security/dependability balance of protection systems is thoroughly discussed in Chapter 3 and Chapter 4.

[2] Significant updates need to be made to ensure data confidentiality, integrity, availability, and privacy. Contemplating such needs and its implications to the reliability of the power grid, NERC has developed and is currently updating a series of cyber-security standards 10. NERC, *Addressing the directives issued by FERC, in Order 706 relative to the approved Cyber Secuirty Standards CIP-002-1 through CIP-009-1*. 2008-06..

To conclude, the philosophy of adaptive protection can be summarized by three principles:

1. ***Adaptive Relaying:*** is defined as the ability of relays to change their settings, operation, or logic to adapt to prevailing system conditions [8].
2. ***Critical Locations:*** sites in power systems where embracing adaptive schemes would be most beneficial.
3. ***Adaptability Scope:*** WAMs should not directly intervene with high speed protection.

These three principles guided the author while developing the methodology for the proposed security/dependability adaptive voting scheme.

1.2 State of the Art Wide-Area-Measurements

The electric power system can be considered to be the largest machine ever invented by men. Generators deliver power through an interconnected grid that extends for hundreds or even thousands of miles. Being a synchronized system, the stability of the system and its ability to deliver power depends on, with more or less extend, each and every component of the system.

The power system has a dynamic nature: load randomly varies with time, the system topology changes, generators are constantly reacting to small perturbations, machines are brought in and out of service, auxiliary equipment is temporarily disconnected for maintenance purposes, transmission lines suffer short circuit faults, etc. Major disturbances can alter the energy balance in the system and jeopardize its operation. In order to analyze the security of the system and to ensure its proper operation it is imperative to track the system-operating condition.

The state of a power system is basically a snap shot of the system at a certain point in time. It is usually helpful to classify the system-operating condition into five states [11]: normal, alert, emergency, in-extremis and restorative. A basic schematic is shown in Figure 1-1. This characterization provides the framework in which control strategies and operator actions are determined to deal effectively with each state. A similar but simpler categorization of the system state will later on lay the foundation of the proposed adaptive protection system.

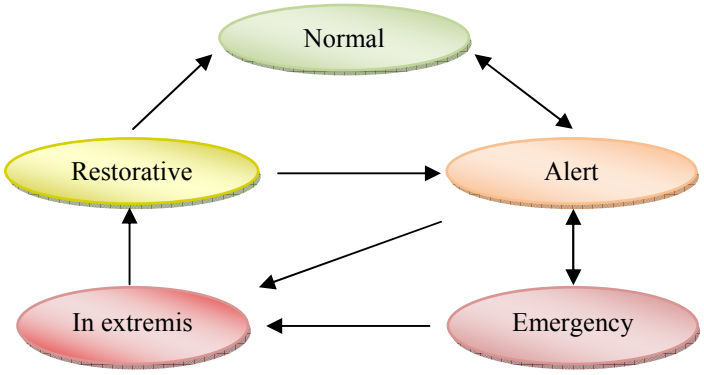

Figure 1-1. Power systems operating states.

In order to estimate the state, a set of measurements, analog or digital, are acquired from different areas in the system. Traditionally, these measurements usually consist of power injection, power flows, voltages and current injections. Remote Terminal Units (RTUs) collect measurements from substations. This information is then transmitted to a control center where the Supervisory Control and Data Acquisition (SCADA) module processes the data and estimates the state of the system.

With the advancement of Phasor Measurements Units (PMUs) great improvements appear on the horizon for state estimation. A set of complex phasor voltages at every bus fully specifies the system state (assuming that the topology of the system is perfectly known).

A phasor is a complex number that represents a sinusoidal wave. Consider a pure sinusoidal wave given by,

$$x(t) = X_m \cdot \cos(\omega \cdot t + \theta) \tag{1.2.1}$$

where X_m is the wave's peak amplitude, ω is the signal frequency in radians per second, and θ is the phase angle in radians. Equation (1.2.1) can be re-written using Euler's notation as,

$$x(t) = \mathrm{Re}\left\{X_m \cdot e^{j\omega t} \cdot e^{j\theta}\right\} \tag{1.2.2}$$

The complex number X is known as its phasor representation and it is given by,

$$X = \left(\frac{X_m}{\sqrt{2}}\right) \cdot e^{j\theta} \tag{1.2.3}$$

where the term in brackets is the root mean square (RMS) of the signal. Note that it is implicitly understood that ω is the frequency.

For the purpose of this dissertation, a PMU is a black-box that provides synchronized phasor measurements, i.e., time-tagged voltages and currents represented as RMS amplitudes and phase angles. A one pulse-per-second obtained from Global Positioning System (GPS) receivers provides a common synchronizing signal. The ability to keep devices that are miles apart synchronized constitutes the breakthrough that gave birth to PMUs. The functional principles and history of PMUs can be found in [12].

Currently, utilities are actively and aggressively deploying PMUs throughout the power grid. Figure 1-2 shows a map with networked PMU in the U.S. The source of this map is the North America Synchrophasor Initiative (NASPI) [2]; a collaborative effort between the Department of Energy (DOE), the North American Reliability Council (NERC) and North American electric utilities. Their objective is to delineate a clear path for PMU deployment, phasor data-sharing, and PMUs applications.

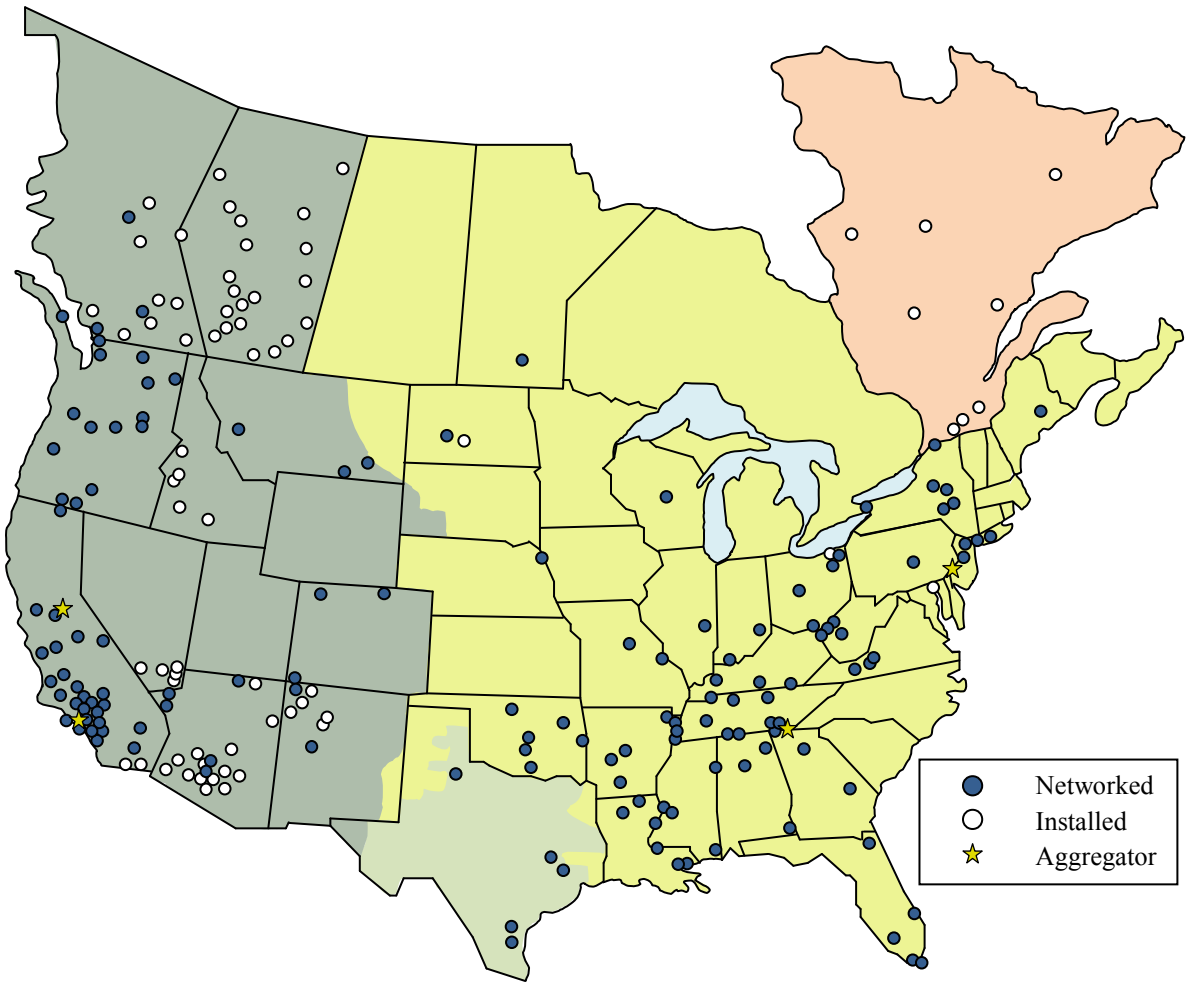

Figure 1-2. Phasor measurement units in North American power grid [2].

1.2.1 Need for WAMs

Recent blackouts have put in evidence the need for a wide-area perspective of the power grid [3, 6, 7]. Consider, for example, the Italian blackout of 2003 which was originated by a fault in Switzerland. Prior to the event, Italy was importing 6.6 GW of power, which represented approximately 28% of its total electricity consumption. After the first contingency, a line-to-ground fault in Switzerland, several tie-lines became overloaded. The communication of such information from ETRANS (Swiss TSO) to GRTN (Italian TSO) is still disputed. A call between ETRANS and GRTN did occur, in which ETRANS requested GRTN to reduce by 200 MW the imported power. The Italian TSO maneuvers successfully reduced the power transfer by the amount requested. However, the amount of relief needed was underestimated and after a second line-to-ground fault a cascade of line trips isolated Italy from its neighbors. Significant voltage

dips caused several impedance relays miss-operations which exacerbated the already critical scenario.

The lack of wide area measurements was found to be a major contributor to the blackout. Prior to the disturbance, Italy had generation reserves well above 6.4 GW and interruptible industrial load of approximately 1.2 GW. Therefore, had the Italian TSO been aware of the developing situation in Switzerland further preventive actions could have been implemented and the blackout could have potentially been prevented. The final reports on the U.S/Canada blackout of 2003 [6], and the UCTE split of 2006 [3], also recognize the lack of situational awareness as a major cause of the blackouts. Both reports strongly recommend deploying PMUs to improve the real-time system visualization.

WAMs will revolutionize the power industry. Most certainly, every aspect of the power system can be improved with the aid of PMUs. Promising candidates include:

- State Estimation.
- Situational Awareness and Security Assessment.
- System Modeling.
- Post-mortem Analysis.
- Protection Systems.

The author has devoted his efforts towards developing protection systems based on WAMs. Throughout the current dissertation an emphasis on the wide-area perspective of power systems will be made.

1.3 Hidden Failures and the Region of Vulnerability

A hidden failure is defined as a permanent defect on a relay system that will cause the incorrect removal of a circuit element as a direct consequence of another event [13]. As conveyed by the definition, hidden failures remain dormant until a particular event causes its manifestation and associated relay miss-operation.

The modes of hidden failures are a function of the relay type, i.e., different protection schemes are prone to different hidden failures. The analysis of the different modes is highly

correlated with the logic diagram of the protective scheme. A detailed description of the different modes of hidden failures for each relay type can be found in [14].

The region of vulnerability is defined as a physical region in the network such that any fault within that region will trigger the hidden failure and produce an unwanted operation [13]. The length of the region of vulnerability is a function of the relay type, the relay settings and the topology of the system. It can be expressed in kilometers by:

$$RV_{km} = \left(Z_{line} \cdot Kf\right) \cdot \left(Bus_{density} - 1\right) \cdot \left(Z_{Base}\right) \cdot \left(OKM_{factor}^{-1}\right) \tag{1.3.1}$$

where Z_{line} is the impedance of the line in per unit, Kf is a relay setting dependent variable, $Bus_{density}$ is the number of transmission lines connected to a bus, Z_{base} is the base impedance and OKM_{factor} is an ohms per kilometer factor.

As an example, consider the one line diagram of two adjacent transmission lines shown in Figure 1-3. Assume that an impedance relay is located at bus A and that its Zone-2 timer has a defect. The aftermath is a lack of coordination between breakers CB_{AB} and CB_{BC}. Therefore, a fault F1 within the reach of Zone 2 of relay A will caused the trip of breaker CB_{AB} and the incorrect removal of the line between bus A and B.

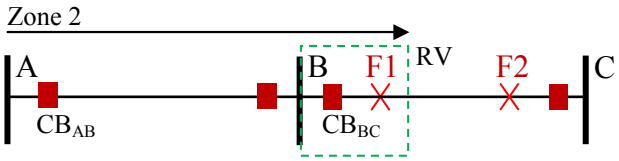

Figure 1-3. Example of a hidden failure in distance relay.

In Figure 1-3 the region of vulnerability is denoted by a dashed rectangle. It needs to be emphasized that any fault outside the reach of the region of vulnerability will not awake the hidden failure. Therefore, the fault labeled F2 would leave the defect on the timer hidden.

Overall, the probability of a protective relay having a hidden failure is relatively small. Manufacturers perform extensive quality control tests to insure a low rate of miss-operations. Then, why should we concern ourselves with hidden failures? The threat that hidden failures pose is due to the intrinsic high risk associated with them. Risk is defined as the product of the

probability of a hidden failure times its impact or consequence. Typically, hidden failures are prone to manifest themselves under stressed system conditions [14] and therefore their consequence tends to be rather noteworthy. In general, faults and other switching events tend to increase the likelihood of hidden failures. Prevailing systems conditions like overloaded lines, voltage dips, and overloaded generators also boost the probability of hidden failures.

An analysis of NERC outages reports indicates that hidden failures are involved in over 70% of cascading outages. The "Great Northeast Blackout" of 1965 [15] represents a quintessential example of the threat posed by hidden failures. The blackout was initiated by a hidden failure in a distance relay. The relay had been set for a typical load in 1963. However, line loading steadily increased in the next two years until it reached the outdated relay setting which tripped and initiated a cascading event that left 30 million people without power. Further examples of the interaction between major disturbances and hidden failures in protective relays are exposed in Chapter 2.

Significant research effort has been employed in developing technology to detect hidden failures and prevent them from causing unwanted operations. However, hidden failures in relays are low probability events so it is difficult to economically justify deploying systems to protect every relay in the system from hidden failures. Attention and resources must be concentrated on areas in which the severity of an unwanted disconnection due to a hidden failure is relatively high. These areas are defined as the critical locations of the power system.

1.3.1 Probability of Hidden Failures.

Quantifying the likelihood of manifestation of hidden failures is a non-trivial task. The probability of manifestation not only depends on the characteristics of the protective equipment but also on system topology and prevailing system conditions.

Cascading outages involving hidden failures were modeled in [16, 17] using a random search algorithm based on power system heuristics and stochastic models for hidden failures. Hidden failures were modeled by assuming that the probability of exposing hidden failures in

distance relays is a function of the apparent impedance seen by the relay. Hidden failures in generator's protections were model as a function of the reactive power margins of the machines.

Further research efforts have pursued the study of hidden failures in circuit breaker trip mechanisms [18]. In general, substation bus configurations, like breaker and half, double breaker, and ring-bus, are simplified in power system models and represented as a single bus; this yields the implicit assumption of perfect protection schemes. Classically, reliability assessments carried out by utilities, discard the possibility of protection system malfunction and assume a perfect operation of breakers. In order to evaluate the impact of hidden failures, a breaker-oriented substation model was built in [18]. The study shows that hidden failures in circuit breakers can significantly downgrade the power system reliability. A two-state Markov process was used to model the stochastic nature of hidden failures. In the study, the frequency of hidden failures was assumed to be independent of their own rate of occurrence and their repair rates.

To conclude, several heuristic stochastic models have been proposed to quantify the likelihood of manifestation of hidden failures. For the purpose of this dissertation, emphasis is made on the consequence of hidden failures, regardless of the probability of manifestation.

1.4 Security vs Dependability

Reliability in the context of power system protection comprehends two aspects, dependability and security. As defined by the IEEE [9] dependability is *"the degree of certainty that a relay or relay system will operate correctly"*, i.e., it is a measure of the certainty that the relays will operate correctly for all the faults for which they are designed to operate [19]. Security *"relates to the degree of certainty that a relay or relay system will not operate incorrectly"*. In general, enhancing security implies an intrinsic loss of dependability and vice versa. Protection engineers try to achieve an optimal balance between these two conflicting concepts; this is why power systems protection is often recognized as an art.

Traditionally, protection systems have been biased towards dependability. System topology and good stability margins justified such design. An adequate transmission line redundancy entails a variety of alternative paths for power to flow. Power systems that exhibit

sufficient transmission line redundancy can withstand losing a line due to lack of security without jeopardizing the systems operation; provided that lines have enough loading margins. Under this scenario, the consequence of not tripping when a fault occurs (lack of dependability) is far worse than tripping when it is not necessary (lack of security).

It is argued in this dissertation that due to the manner in which the system has evolved, this philosophy needs to be reviewed and that, under stressed system conditions, a favorable bias towards security can be beneficial. According to the Edison Electric Institute (EEI), a long downward trend can be observed in transmission investment [20]. Figure 1-4 shows a plot with historical investment in the transmission grid from 1975 to 2003; the dollar amounts are adjusted to 2003 dollars. The downward trend lasted for more than two decades and the latest survey indicates an increase in investment after 1999. However, it will take several years to upgrade the neglected grid; new high voltage transmission lines can easily take more than 5 years to build.

On the other hand, electricity consumption has steadily increased at an annual rate of approximately 2% [21]. As a result the system is operated under tighter and tighter conditions and it has become imperative, on some specific critical locations, to avoid tripping when it is not necessary, i.e. to be biased towards security.

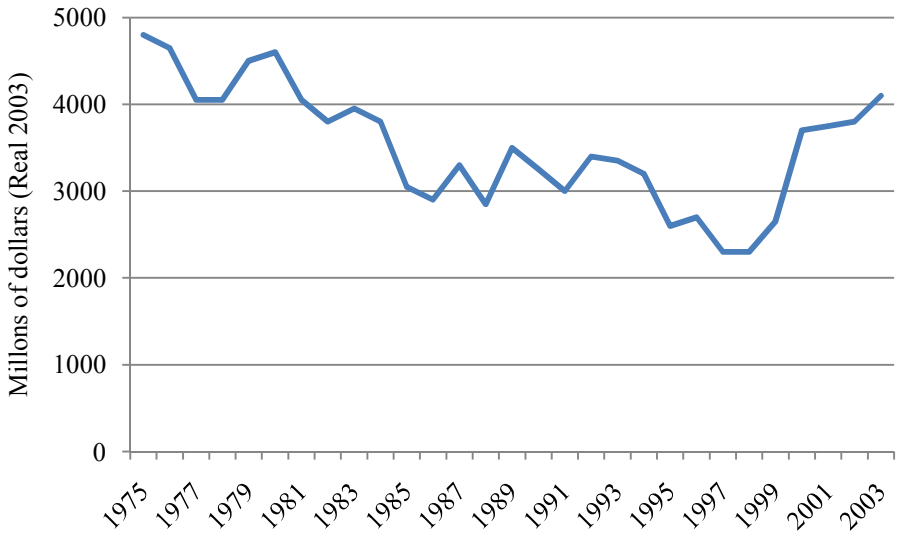

Figure 1-4. Historical investment in transmission infrastructure; source EEI [20].

The de-regulation of the power industry has also played an important role on the need for security biased protection. Inter-areas tie-lines where originally built with the purpose of sharing reserves and to increase reliability. Under the new electricity market, economics guide the operation of the system and inter-ties are now used to transfer bulk power. The large power flowing between areas reduce the system stability margin and presents new challenges to the system operator. During the Italian blackout of 2003 [7], auto-reclosers failed to restore key inter-tie lines due to the large angle across them; approximately 42 degrees. Typically, a maximum angle of 30 degrees is allowed in order to protect nearby generators from high transients stresses that occur during switching of network elements.

To conclude, when the power system is in a "safe" state, a bias towards dependability is desired. Under such conditions, not clearing a fault with primary protection has a greater impact on the system than a relay miss-operation due to lack of security. However, when the power system is in a "stressed" state, unnecessary line trips can greatly exacerbate the severity of the outage, contribute to the geographical propagation of the disturbance, and may even lead to cascading events and subsequent blackout. Under such states, it would be desirable to alter the reliability balance in favor of security.

The main contribution of this dissertation is the development of methodology to implement an adaptive protection scheme that can alter its security/dependability balance to suit prevailing system conditions.

1.5 Preliminary Overview of the California System

The topology of the California system, old inherited technology, and the manner in which the system is operated further motivates the need for the adaptive security/dependability protection scheme proposed in this dissertation. The advocated methodology was designed and tested using a highly detailed 4000 bus model of the California system. The backbone of the electric grid is shown in Figure 1-6. The figure depicts a representative one line diagram of 500 kV lines and DC lines. Figure 1-7 shows the different areas controlled by utilities in California. Further details on the model characteristics are given in Chapter 4.

At the *iPCGRID* conference in San Francisco CA 2009, it was reported by [22] that four generations of analog, solid state, and some early microprocessor based relays still coexist with modern protection systems. Improper relay operations due to hidden failures or lack of security are known in the jargon as "over-trips". The industry average "over-trip" rate, in California and neighbor utilities, is approximately 10% for Extremely High Voltage (EHV) relays systems [22]. In the course of 10 years, the Northern utility in California, Pacific Gas and Electric (PG&E) (see Figure 1-7), has experienced a rate *"well above the industry rate"* [22] in relay mis-operations; a particular concern is given to "over-trips" on 500 kV transmission lines.

California is a net importer of power. Figure 1-5 shows the ratio between the power produced in California and the imported power [23]. Instate generation accounts for approximately 70% of the electric energy consumed. The other 30% is imported from the Pacific Northwest, an approximate 8%, and from the Southwest, 22%.

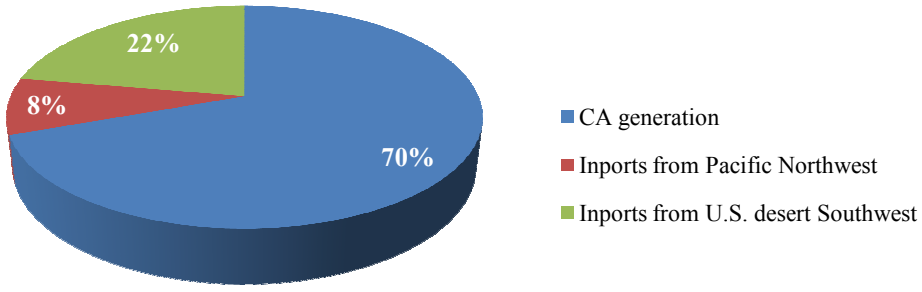

Figure 1-5. Percentage of power produced in California versus power imported from neighbor areas. Source [23].

The clear North-South topology of the system combined with the significantly large percentage of power imported by California turn relay miss-operations into an extremely costly affair which can potentially jeopardize the stability of the system. The unnecessary removal of one of the backbone 500 kV transmission lines has a profound impact in the system transmission capacity. Consequently, it may be necessary to re-dispatch generation which is, more often than not, done at a higher real-time market price. Some events may require load shedding and, during stressed conditions, hidden failures may cause a system collapse. Simulation results presented in Chapter 4 confirm the potentially catastrophic consequences of hidden failures.

This research effort is part of a project funded by the California Institute for Energy and the Environment (CIEE). The Technical Advisory Group (TAG) appointed to oversee the project

has welcomed the proposed adaptive methodology to enhance protection security. Recently, the U.S. Department of Energy (DOE) has awarded research funds to demonstrate the feasibility of the scheme by deploying equipment to test, in real-time, the proposed adaptive relay [24].

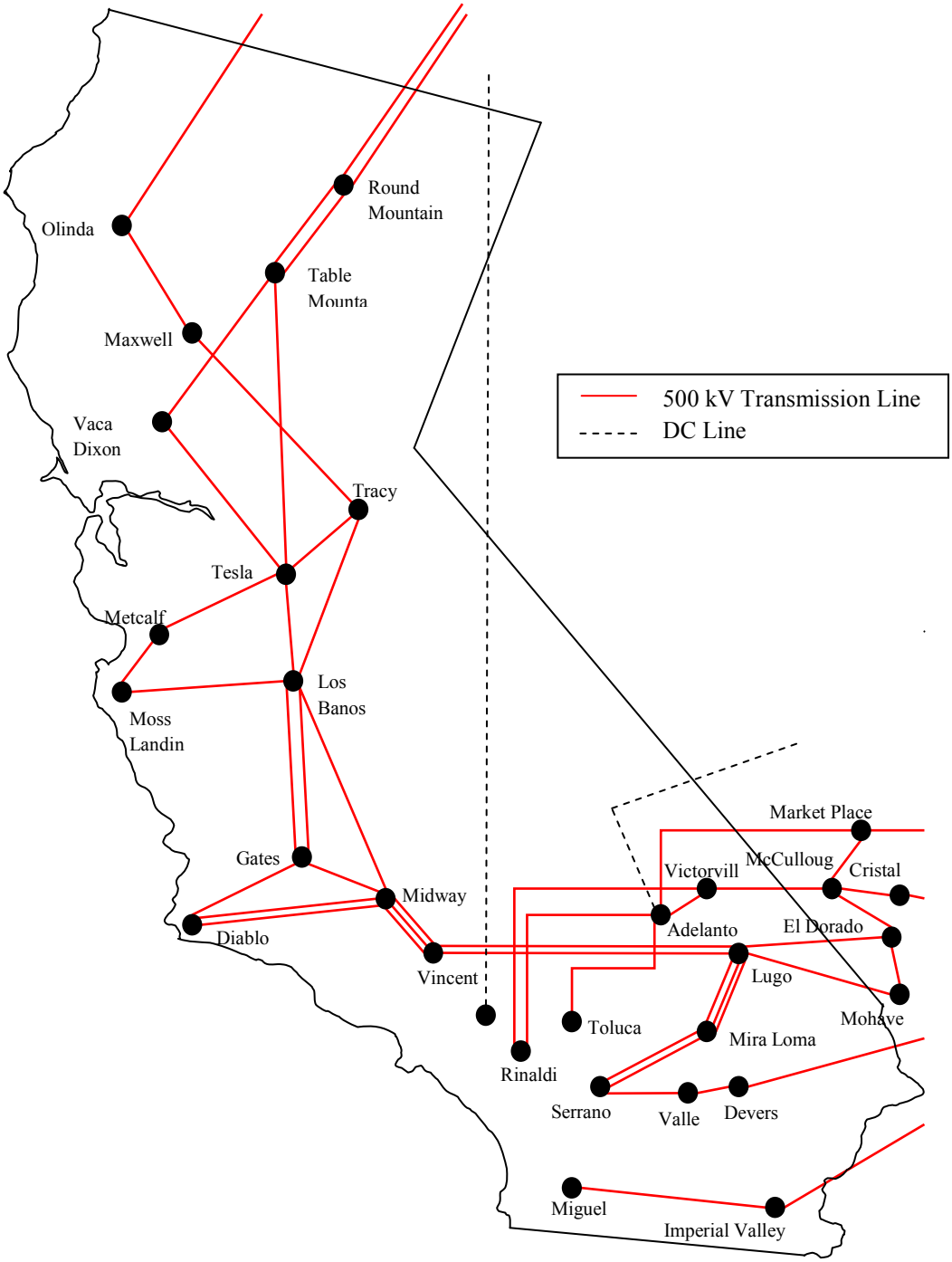

Figure 1-6. . One line diagram: High Voltage Transmission Grid of California.

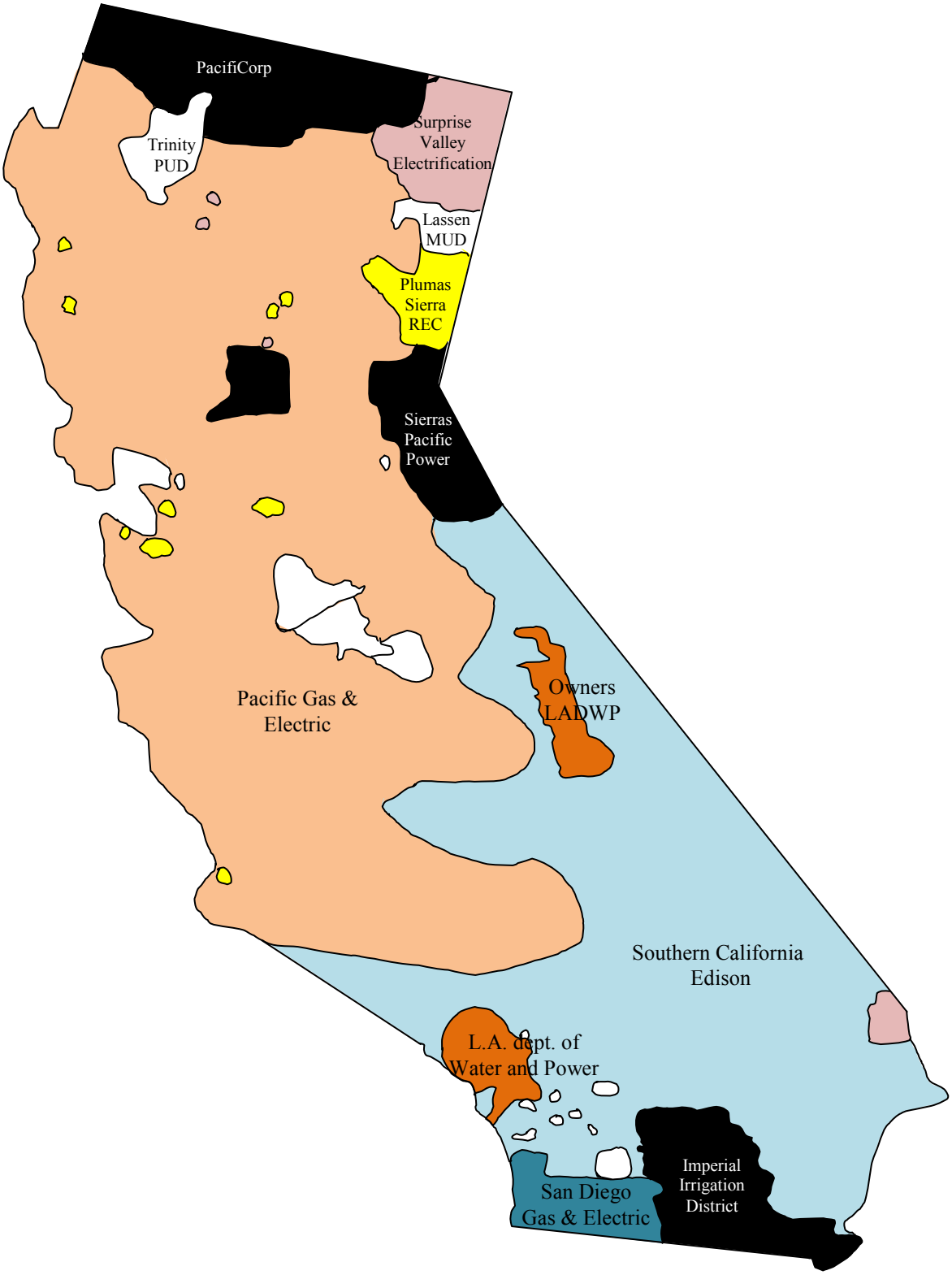

Figure 1-7. Utilities control areas in California; source [1].

1.6 Conceptual Overview of an Adaptive Security/Dependability Voting Scheme

The adaptive philosophy of protection systems acknowledges that relays may change their characteristics in order to tailor their operation to the prevailing system conditions. The methodology proposed in this dissertation aims to reduce the likelihood of hidden failures and potential cascading events by adjusting the security/dependability balance of protection systems.

A conceptual overview of the security/dependability adaptive voting scheme is given by the schematic shown in Figure 1-8. First, it is recognized that there are some critical locations in the power grid where adaptive relaying is most beneficial. A systematic procedure to identify such critical locations of protective relays is discussed in Chapter 3. The analysis involves an exhaustive set of simulations that include faults and hidden failures in protection relays. A ranking is made by assessing the severity of different disturbances. Hidden failures at the top of the list are potential location candidates for placing the adaptive scheme.

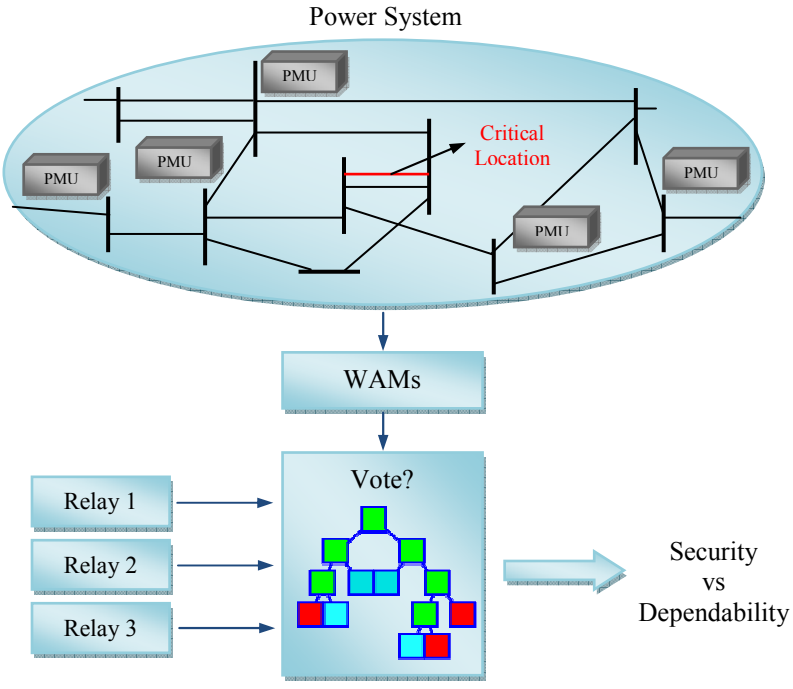

Figure 1-8. Conceptual schematic: adaptive security/dependability voting scheme.

The voting scheme consists of a set of three independent and redundant relays. The scheme can be categorized as an open-loop discrete-event control. Discrete event controls are characterized by a specific action taken when the state of the system exceeds a threshold value. Several stability controls and protection schemes operate in this manner: dynamic brakes, switched shunt capacitors, under-frequency load shedding, etc.

Wide area measurements are obtained with the aid of PMUs. The underlying hypothesis is that phasor measurements at specific buses provide enough information to discriminate the need for a bias towards security. These measurements are used to infer the state of the power system which is then classified as either "stressed" or "safe". If the system is found to be stressed, the proper course of action is to enable the voting scheme and therefore bias the protection system towards security. On the other hand, if the system is found to be safe, the voting scheme will be disabled and only one relay takes on the protective function, i.e., a favorable biased towards dependability.

The description above raises several questions. How to determine the optimal location, i.e. the most beneficial location, for the voting scheme? Where should PMUs be placed in order to infer the system state? How should a "stressed" or "safe" state be defined? In Chapter 3 the methodology to implement the proposed security/dependability voting scheme is developed. The classification of the system state into "stressed" or "safe" is accomplished using Data Mining; specifically, Decision Trees (DTs). Data mining is defined as the process of discovering patterns in data [25]. It is a non-parametric statistical analysis highly suited to power systems due to the complex non-linear behavior of the system. Decision Trees can extract information of large data sets and intuitively represent the gained knowledge through a series of if-else sentences. The main idea is to partition the state space in a clever way in order to develop decision rules to adjust the security/dependability balance of the protection scheme.

Chapter 2 Literature Review

As stated previously, the concept of adaptive relaying has been around for decades and yet very few adaptive schemes have been designed and implemented. In the following chapter an overview of further adaptive protection schemes is presented.

The methodology proposed in this dissertation is based on Data Mining. A wide range of data mining applications to power systems have been proposed in the literature [5, 25-34]. Encouraging results and the huge amount of information soon to be available through PMUs, promise to enhance the interaction between Data Mining methods and power systems. The following sections discuss some data mining applications to power systems.

2.1 Protection Systems Based On Wide Area Measurements

Recent major disturbances have made evident how critical protection schemes are for the reliable operation of the power grid [3, 6, 7]. The following examples portray smarter protection schemes that can be implemented using wide-area measurements. As a member of the Power System Research Lab at Virginia Tech, the author has contributed on the feasibility study and the design of these schemes. These PMU based protection schemes and other PMUs applications can also be found in [12].

2.1.1 Adaptive out of step:

The first PMU based protection scheme to be proposed and implemented was an adaptive out-of-step relay [35]. Power swings across the Florida-Georgia interties motivated the study of such scheme. The system exhibits a clear North-South topology and it behaves like a two machine system.

Consider the following two machine system shown in Figure 2-1.

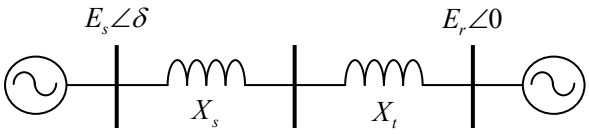

Figure 2-1. A two machine system

When the power system network is perturbed from its steady state the generators react following Newton's law of motion:

$$\frac{2H}{\omega_s}\frac{\partial^2\delta}{\partial t^2} = P_a = P_m - \frac{E_s E_r}{X}\sin(\delta) \tag{2.1.1}$$

where H is the inertia constant, ω_s is the machine synchronous speed, δ is difference between the machines rotor angle, P_a is defined as the accelerating power, E_s and E_r are the sending and receiving voltages respectively, and X is the equivalent reactance ($X = X_t + X_s$). In power systems equation (2.1.1) is known as the swing equation. When the angular difference between the

generator rotors move in a monotonic fashion the swing is said to be unstable and the generators need to be isolated from the system.

In general, an out-of-step relay scheme performs the following control actions:

1. For stable swings: it blocks line tripping.
2. For unstable swings: it initiates selective tripping to ensure an adequate generation/load balance after the system is islanded. It should also block line reclosing under unstable swings.

Conventional out-of-step relays are based on distance relays. The apparent impedance seen by a distance relay located at the sending bus can be computed by:

$$Z = \frac{E_S \angle \delta}{Y\left(E_S \angle \delta - E_r \angle 0\right)} \tag{2.1.2}$$

where Y is the admittance equivalent.

Equation (2.1.2) shows that the apparent impedance seen by the distance relay varies as the rotor angle difference between the generators changes; as the angle difference increases the apparent impedance decreases. Unlike line faults where the impedance changes almost instantaneously, the rate of change during swings is significantly slower. Therefore, the relay can distinguish between faults and swings by timing the rate of change of the impedance.

To conclude, traditional out-of-step relays use the apparent impedance as an indirect measurement of the angle difference between generators (or areas in a power system). The out of step scheme uses a combination of distance relays, timers and blinders to distinguish between faults, stable swings and unstable swings. Zones lengths and timer settings are derived using off-line simulations for credible contingencies.

With the advancement of PMUs it is now possible to simultaneously measure angles at any location in the system and then compute the difference between them. In the scheme proposed in [35] the network is modeled as a two machine system. Aided with direct angle measurements the equal area criterion can be used to determine the stability of angle swings. Therefore instead of solving the swing equation in (2.1.1) it is possible to determine graphically

the stability limit using the power-angle diagram described by equation (2.1.3) below. The details of the equal area criterion can be found on any power system textbook [12-18].

$$P_e = \frac{E_s \cdot E_r}{X} \sin(\delta) \tag{2.1.3}$$

The main drawback of this approach is that the equal area criterion is not applicable to a multi-machine system. As mentioned previously, the Florida-Georgia intertie behaves like a two machine system. However, other power systems in the U.S. do not have this particular characteristic. For example, studies at Virginia Tech show that the California system has several coherent groups of machines. Therefore, a new algorithm is needed to detect unstable swings involving groups of machines.

An ARIMA (autoregressive integrated moving average) model has been proposed by the power system research lab at Virginia Tech. Coherent groups of machines were identified through an extensive set of dynamic simulations. Voltage phase angles of buses that are electrically close to groups of machines accurately reflect the rotor angle excursions of the coherent group. For the statistical regression it is assumed that PMUs are placed at those locations. The time series algorithm then uses the time-tagged angles to predict out-of-step conditions.

2.1.2 Impedance Relays: Supervised Third Zone

Impedance protection relays are universally used to protect transmission lines. Their principle of operation is based on the apparent impedance seen by the relay. Since the impedance of the line is a known parameter, a measurement of the ratio between voltage and current can be used to detect faults.

Consider a distance relay located at bus *A* of the one line diagram shown in Figure 2-2. Zone 1 is an under-reaching zone that operates instantaneously. It is common practice to set this zone between 85% and 90% of the line AB length [19]. Zone 2 is an overreaching zone with a coordination delay in the neighborhood of 0.3 seconds. It is typically set to cover 120% to 150% of the line length AB. The third zone extends between 120% and 180% of the next line section

BC and has a delay of the order of 1 second. Further details on distance relays can be found on any power system protection book [17, 26-28].

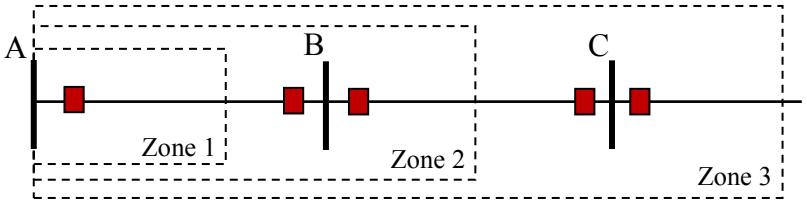

Figure 2-2. Three zone distance relay.

Load encroachment of overreaching zones has been a very well known dilemma for decades. However, after the Northeastern US/Canada blackout the debate over the benefits versus the disadvantages of a third zone gained significant emphasis.

The August 14, 2003 U.S./Canada blackout had its origins in the Cleveland-Akron area (see Figure 2-3). Depressed voltages at the Cleveland-Akron area, the lack of reactive support due to scheduled maintenance of capacitor banks and the outage of Eastlake unit 5, the lack of situational awareness from First Energy (FE) and from the Midwest Independent System Operator (MISO) as a consequence of a glitch in the computer that runs the state estimator, and the outage of three key 345 kV transmission lines caused by the lack of proper right-of-way maintenance, created a lethal cocktail that ended up with 50 million people without electricity.

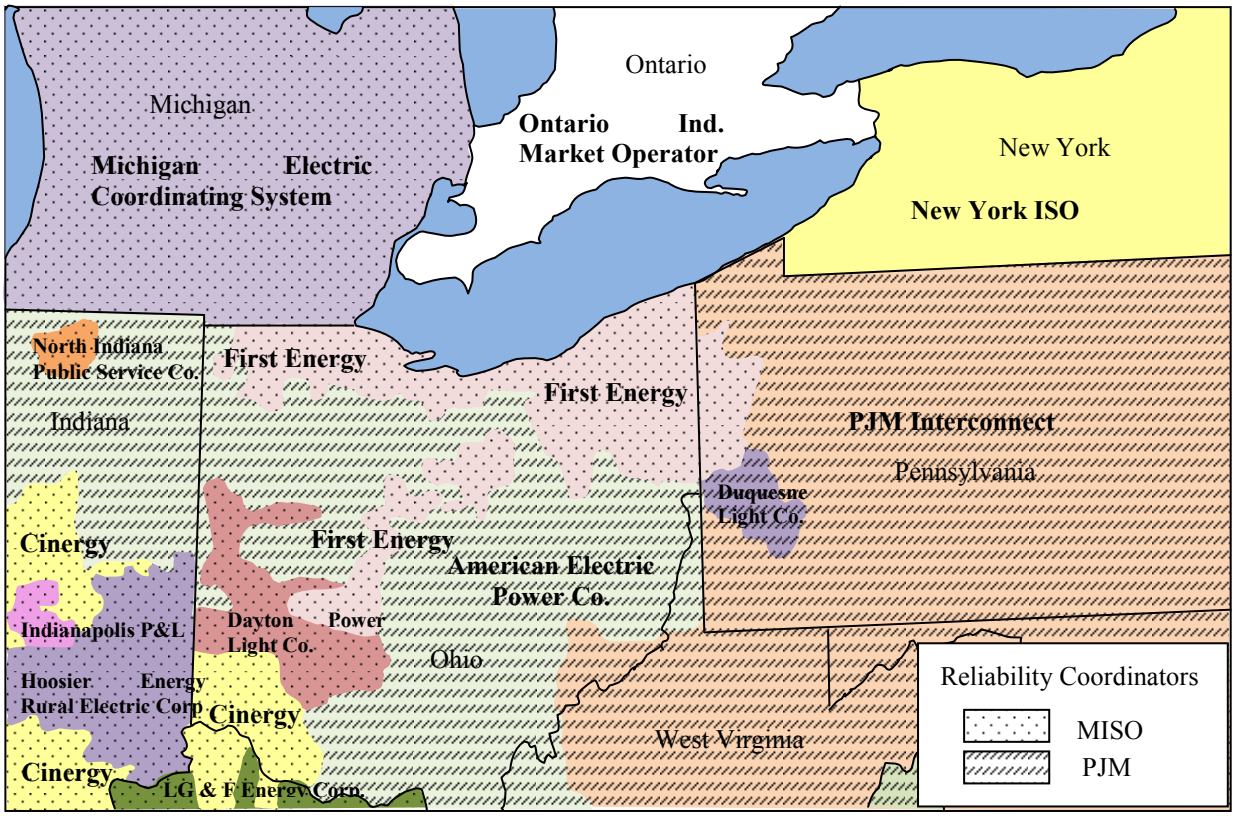

Figure 2-3. Reliability coordinators near Cleveland-Akron area [4].

The report developed by the U.S./Canada Power System Outage Task Force [4] showed a total of 14 impedance relay miss-operations due to third and second zone load encroachment. Figure 2-4 shows a schematic of the lines that were improperly tripped by impedance relays. In the report it is stated that: *"The investigation team concluded that because these zone 2 and 3 relays tripped after each line overloaded, these relays were the common mode of failure that accelerated the geographic spread of the cascade"*. The task team did not pursue any type of simulation analysis to determine whether the blackout could have been prevented, hadn't these lines tripped. However, since it may take several minutes for heavily loaded lines to sag enough for a ground fault to occur, it is likely that the cascading situation may have been recognized which could have led to further preventive and corrective actions. The window of time available under such overloaded condition is a function of wind speed, line loading, line tension and ground clearance.

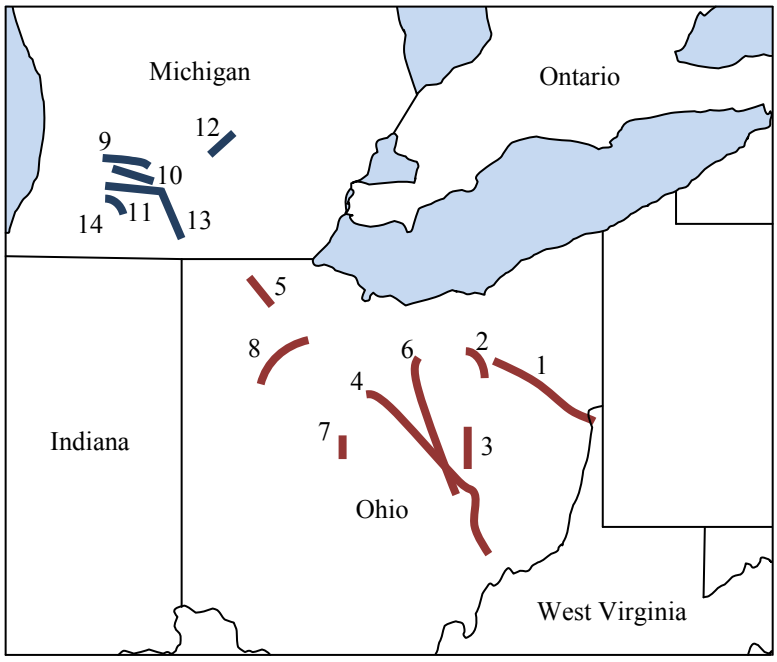

Figure 2-4. Third and second zone relay misoperations [4].

The following are some of the common practices used to increase loadability [36]:

- Changing the impedance characteristic from a circle to a lens.
- Adding blinders to limit the reach along the real axis.
- Using an impedance relay offset into the 1st quadrant.
- Enabling the load encroachment function of the relay.

Most digital impedance relays offer load encroachment functions to account for heavily loaded conditions. The North America Electric Reliability Council (NERC) has recommended using a 150% thermal rating with a 0.85 pu voltage at power factor angle of 30 degrees [36]. The magnitudes of these parameters were determined to be observed under extreme conditions but not in a cascading mode. Figure 2-5 shows the shape of a typical load encroachment function. Under normal operating conditions, load excursions are expected to follow a trajectory that lies within the shaded area of Figure 2-5. However, under extreme stressed conditions load excursions may not follow their customary path as indicated in Figure 2-5. Typically, such trajectories occur when large amounts of reactive power flow through transmission lines. As mentioned before, this was the prevailing scenario in the Cleveland-Akron area due to the lack of reactive power support.

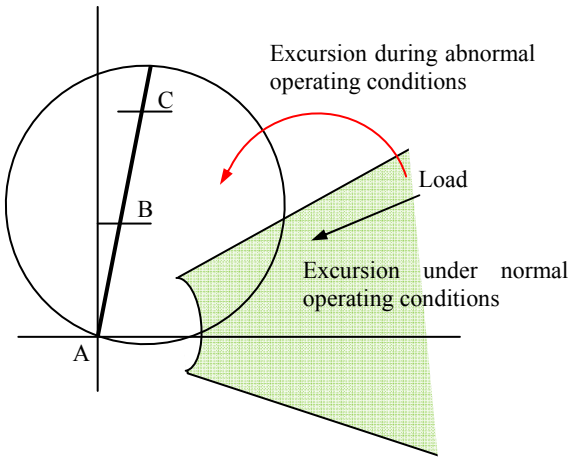

Figure 2-5. Encroachment settings on the R-X diagram.

The lack of a proper solution to load encroachment has led some regions in the US and Canada to completely eliminate the third zone on 230 kV lines and above [4]. However, as argued in [37], this policy cannot and should not be applied as a "one fits all" recipe. There are cases where removing zone 3 does not compromise the reliability of the system; there are also cases where zone 3 is necessary.

In essence, the problem arises when distance relays incorrectly interpret abnormal loads as faults. Aided with wide-area measurements it is possible to design a supervised third/second zone impedance relay based on the following characteristic [12]:

- Unbalanced faults are not associated with load encroachment. Therefore, by simply checking the negative sequence component one may distinguish between load excursions and unsymmetrical faults.
- Three phase faults are the only type of fault that resembles heavy loads. Wide area PMUs can verify the legitimacy of zone pick-up and transmit blocking signals.

Consider one more time the one line diagram shown in Figure 2-2. Assume that zone 3 has picked up at bus *A*. A significant presence of negative sequence would indicate an unsymmetrical fault and the third zone pick up would be appropriate. The lack of negative sequence would indicate that there is either a three phase fault or a loadability violation. A distinction between them can be made based on the information provided by PMUs stationed at

buses *B* and *C*. If any of the PMUs indicate a zone 1 pick up, then the third zone pick up is appropriate; otherwise, it can be inferred that we are under a load encroachment situation and a blocking signal should be sent to the distance relay located at bus *A*. Since the delay of zone 3 is in the order of a second, the communication this information can be done in timely fashion. The same methodology can be applied for zone 2.

As mentioned before, the unwanted operation of distance relays was not the triggering mechanism for the 2003 blackout. However, they significantly contributed to the geographical propagation of the disturbance. The supervised impedance relay scheme described can greatly reduce the likelihood of miss-operation due to load encroachment.

Under the same line of thinking, the power research group at Virginia Tech has also proposed implementing a Supervisory Boundary for zone 3 [38]. As shown in Figure 2-6, such boundary is designed as a concentric circle surrounding zone 3. An alarm is raised if the impedance seen by the distance relay reaches the supervisory boundary. Such encroachment may be due to a steady state increase in load or due to power swings in the system. The main purpose is to alert system operators about potentially dangerous conditions and to highlight the possible need for settings review to protection engineers. At the moment no control actions have been implemented.

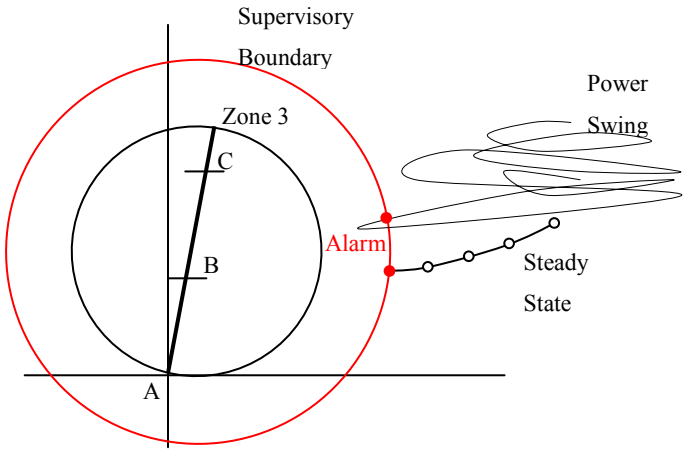

Figure 2-6. Supervisory boundary for zone 3.

2.1.3 Load Shedding Scheme based on WAMs

Any imbalance between load and generation is reflected in the system frequency. An excess of generation leads to an increase in the machines' rotational velocity and a corresponding increase in frequency. A shortage of generation leads to a decrease of the machines' rotational velocity and a corresponding decrease in frequency. The machines' rotational velocity and the system frequency are one and the same.

Under-Frequency Load Shedding (UFLS) schemes detect the onset of the decay in system frequency and shed appropriate amounts of load [19]. Ones the frequency reaches a threshold value predetermined quantities of load are shed. A rate of decay R with units of Hz/sec is used to set traditional UFLS schemes. The rate R can be derived from the swing equation and is defined as,

$$R = \frac{p \cdot L}{H} \cdot \frac{f_2 - f_1}{1 - \left(\frac{f_2}{f_1} \right)^2} \tag{2.1.4}$$

where p is an average power factor, L is a relative load excess factor defined in (2.1.7), H is the system equivalent inertia constant defined in (2.1.5), and $[f_1,f_2]$ is a frequency interval. The details of the derivation can be found in [19].

$$H = \frac{1}{2} \cdot \frac{J_{eq} \cdot \omega_s}{\sum S_i} \tag{2.1.5}$$

$$J_{eq} = \frac{\sum J_i \cdot S_i}{\sum S_i} \tag{2.1.6}$$

$$L = \frac{\sum L_i - \sum G_i}{\sum G_i} \tag{2.1.7}$$

Apprehensive for the dangers of voltage collapse, Under-Voltage Load Shedding (UVLS) schemes have recently become more popular. UVLS react when bus voltages at some particular locations reach a preset threshold value. Its main purpose is to prevent a voltage collapse. NERC has issued a series of standards and recommendations regarding UFLS and UVLS [39-46].

On 4 November, 2006 a major disturbance in the area regulated by the Union for the Coordination of Transmission of Electricity (UCTE, now called European Network of Transmission System Operators for Electricity – ENSO-E) led to a system split into three islands. A schematic with the different islands is shown in Figure 2-7.

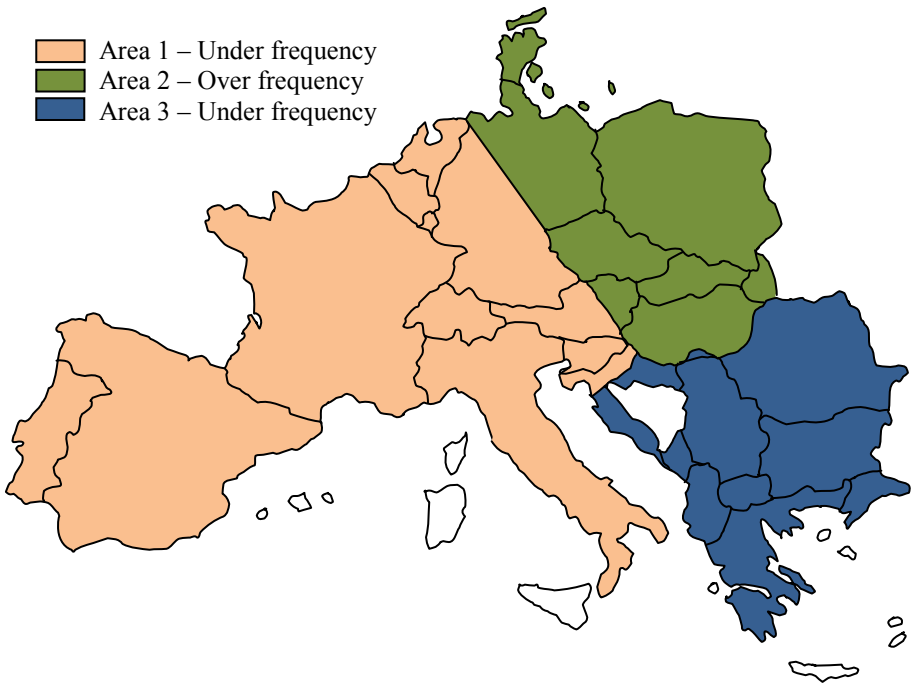

Figure 2-7. UCTE splits into three areas [3]

In the UCTE final report [3] it is stated that area 1 was importing roughly 9 GW of power from its neighbors prior to the system split. As expected, the huge imbalance between load and generation caused the frequency to drop to 49 Hz in approximately 8 seconds. A plot of the change in frequency in area 1 is shown in Figure 2-8. UFLS schemes shed approximately 16 GW of consumption load and 1.6 GW of pump load. An extra 663 MW were manually shed. The reader may notice the discrepancy between the original imbalance of 9 GW and the actual load shed required of approximately 18.2 GW.

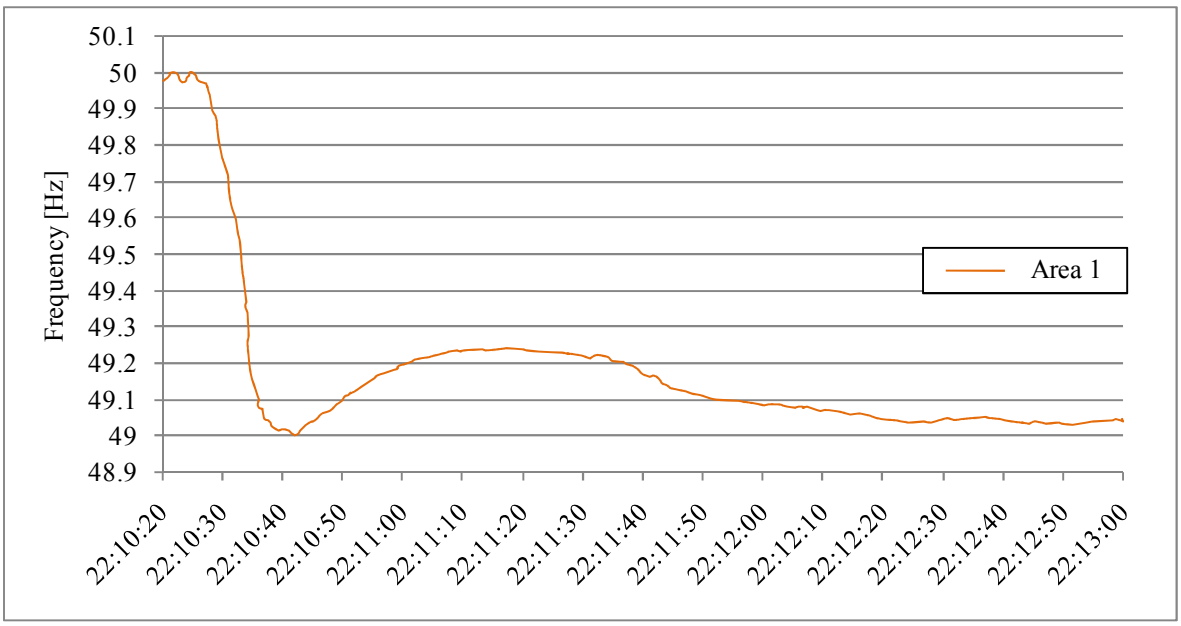

Figure 2-8. Frequency drop in Area 1 [3].

As briefly summarized previously, UFLS schemes react after the frequency has reached a predefined threshold. In general, Europe has a large penetration of wind and combined-heat-and-power generators. Typically, these are connected to the distribution grid and are therefore subjected to less constraining standards regarding frequency performance. Approximately 10.9 GW of generation were tripped as a consequence of the frequency drop which explains why the amount of load shed more than double the initial expected deficit of 9 GW.

A significant improvement can be achieved if the load shedding scheme reacts before having a significant drop in frequency. The load shedding schemes performed as designed but their effectiveness was undermined by the lost of generation. If corrective actions had been taken during the eight seconds that took the frequency to drop 1 Hz the severity of the disturbance could have been greatly reduced.

Another good example of the high cost of waiting for a frequency drop can be found in the August 1996 WSCC Blackout [47] where generation was also tripped due to the frequency dip. It is also important to emphasize that deciding the amount and location of load to be shed can be as critical as a fast operation. In the July 1996 WSCC blackout [47] load was shed at the power sending side which caused several tie-lines to be overloaded which in turn led to a loss of synchronism. In the 1977 New York blackout [48] excitation protection tripped several machines after a voltage rise caused by load shedding.

A load shedding scheme based on wide-area measurements (WAMs) was proposed in [12]. The authors suggest computing a real-time area control error (ACE) to estimate deviations on tie-line power flows. In general, the purpose of the ACE is to maintain a scheduled flow through tie-lines, i.e., a change in load in an area should be compensated by an appropriate change in generation in the same area; therefore maintaining the committed power exchange. The control signal is made up of tie-line flows and a frequency deviation measurement. Typically, the ACE is defined as,

$$ACE = \Delta P_{ij} + \beta \cdot \Delta f \tag{2.1.8}$$

where ΔP_{ij} is the power flow deviation, Δf is the frequency deviation, and β is an area frequency-response characteristic [11] and is a function of the steady sate speed versus load characteristic of the generators and of the effects of frequency variation on the system loads. Under extreme conditions, the protocol is to change the ACE control output to maintain frequency rather than the power exchange. Existing protocols were correctly applied during the 2006 UCTE disturbance.

A schematic of the WAMs based load shedding scheme is shown in Figure 2-9. Instead of simply making decisions based on frequency a smarter scheme can be design using relevant information provided by WAMs. A control signal can be designed as a function of:

- The real-time ACE.
- Power flow measurements at vital inter-ties.
- Critical machines' output power as well as real and reactive power margins.
- PMUs voltage and angle measurements of key buses in the system.
- Real-time load characteristics.

The main advantage of a WAMs based load shedding scheme is that it can adapt its response according to the state of the system. Extreme events, such as the 2006 UCTE disturbance, require an aggressive load shedding plan to prevent generation from tripping. Waiting for a drop in frequency is not an option under this scenario since the cost of waiting is too big.

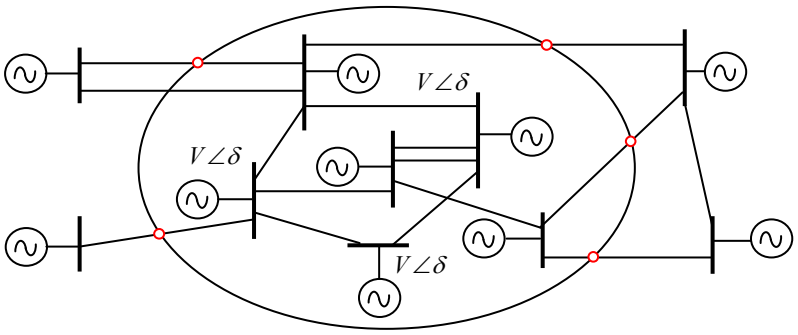

Figure 2-9. Schematic of a load shedding scheme based on wide-area measurements.

The underlining hypothesis is that it is possible to predict frequency drops before they happen. Such scheme is currently being developed by the power research group at Virginia Tech. A data mining approach, similar to the one used in this dissertation, could potentially be applied to build a decision tree 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 decision tree. For our purposes, optimality can be understood to be a function of: the amount of load to be shed, the location where load relief is needed, and the market cost of load interruption.

Other ideas for load shedding relays have been proposed in [49-51]. In [50] the problem is stated as a minimization problem where the protection system determines the minimum amount of load needed to maintain frequency and bus voltages within predefined margins. Margins for angle differences across critical inter-ties are also contemplated in the minimization problem. In [51] a polynomial function in conjunction with phasor based indexes is used to determine the amount of load to be shed. In [49] intelligent schemes are discussed in general terms and field test measurements are used to highlight the importance of load characteristics.

2.2 Data Mining Applications in Power Systems

Data mining is defined as the process of discovering patterns in data [25]. The goal is to extract rules or knowledge from regularity patterns exhibited by the data. Lately, there has been an exponential growth on the number of papers that exploit the benefits of data mining. The large scale of power systems and the complex nonlinearities that govern its behavior make data mining an attractive and simple tool to analyze the huge amounts of information available.

In [52] an overview on applications of data mining to power systems is given. Figure 2-10 shows a comparison between the different data mining methods used in power systems. It can be seen that Decision Trees (DTs) are on top of the list with 86.6% of the papers using it as the preferred method. The foundational theory behind DTs will be presented in Chapter 4 but at this moment it is important to emphasize that one of the attractive features of DTs is their simplicity. Complex knowledge hidden in data patterns is extracted and represented through series of questions with Yes/No answers; this is known as recursive binary partitioning.

Data mining has found a niche in a wide range of areas in power systems. Figure 2-11 shows the different applications of data mining in power systems. Security Assessment, fault detection, power systems control and load forecasting have been the major areas of focus.

In the following subsections some applications of Data Mining to power systems are briefly described. Special attention is given to Decision Trees. In general, studies using DTs have shown excellent predictive accuracies, which in part explain why it is the most popular data Mining method used in power systems.

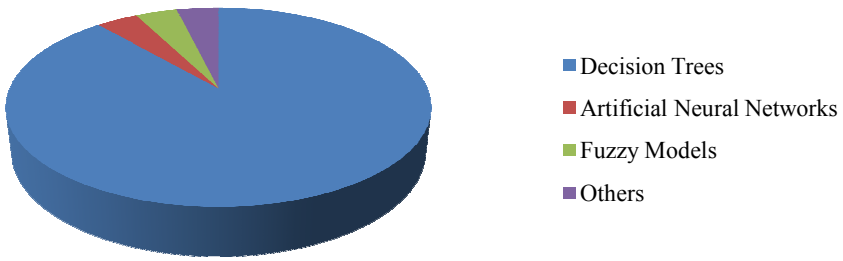

Figure 2-10. Data mining methods used in power systems [52].

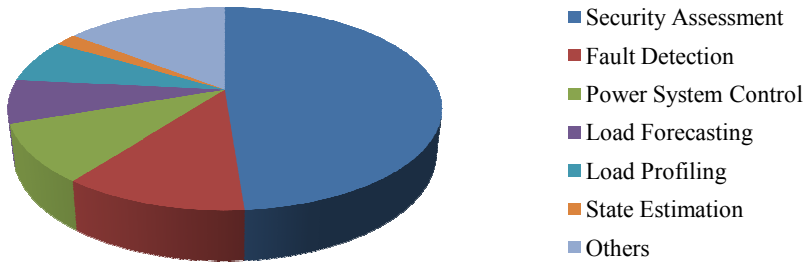

Figure 2-11. Data mining applications in power systems [52].

2.2.1 Data Mining and Security Assessment

As defined by the North American Reliability Council (NERC) security *"is the ability of the electric system to withstand sudden disturbances such as electric short circuits or unanticipated loss of system elements"*. Decision Trees are particularly suited to real-time security assessment due to their fast execution. A vector of attributes measured in real-time is dropped down the tree and settle into a terminal node almost instantaneously. The heavy work is done during the off-line training of the tree; the real-time execution is virtually instantaneous.

One of the first data mining application to power systems was proposed by [26]. Decision trees were used for transient stability prediction. The foundational hypothesis was that machine rotor angles contain enough information to predict stability. Specifically, three samples of rotor angles measured four cycles apart were used to compute two angle velocities and one angle acceleration per generator. Equations (2.2.1), (2.2.2), and (2.2.3) show the aforementioned quantities.

$$v1_i = 10 \cdot \left[\delta_i(t_1) - \delta_i(t_0) \right] \tag{2.2.1}$$

$$v2_i = 10 \cdot \left[\delta_i(t_2) - \delta_i(t_1) \right] \tag{2.2.2}$$

$$a_i = 20 \cdot \left[\delta_i(t_2) - 2 \cdot \delta_i(t_1) + \delta_i(t_0) \right] \tag{2.2.3}$$

where δ_i is the rotor angle of machine i, and t_0, t_1, and t_2 are samples taken four cycles apart.

These quantities are used as predictors by the decision tree training algorithm. An objective function is needed to classify the target or dependent variable as stable or unstable. The

selected criterion was the angle difference between any two generators. If such difference exceeds 360 degrees four seconds after the clearing time, the target is labeled as unstable; the dependent variable is said to be stable otherwise. Such criterion was derived based on engineering judgment and simulations.

A large set of faults of various lengths were applied to a New England 39 bus test system to train the tree. In spite of the fact that the model used was relatively small, the computational burden required the use of a cluster of IBM RISC System/6000's at the Cornell National Supercomputer Facility (CNSF). Their results show predictive accuracies better than 91%. One of the authors' suggestions to improve the accuracy of the tree was the inclusion of various system operating points in the training set [26].

A major limitation for the application of this study is raised by the fact that the sampling window is initiated after the contingency is cleared, i.e., the decision tree process should be triggered immediately after a fault is removed. Therefore, it is not only necessary to transmit the generators' phasor information but also an outage detection signal from every line in the system, which renders the approach economically infeasible.

Similar approaches have been proposed by other authors [5, 29, 30, 34]. Diverse power systems models with significantly different topologies have been used to train and test the proposed methods; all showed accuracies good enough to make implementing the DTs feasible. There is also a wide range of suggested predictors and criteria to be used to classify the target (dependant variable).

In [30] the Zhenjiang power grid of China was used to train the DT (see Figure 2-12). Power flows and angle difference with respect to a slack bus were used as predictors. The target was classified as insecure if generators angles exceeded 500 degrees, or voltages dropped below 0.7 pu for more than 1 second, or frequency decreased to less than 49 Hz for more than 1 second.

In [5] the emphasis is set on voltage collapse; specifically, voltage collapse initiated by critical contingences selected based on AEP's[3] experience. The methodology is based on load flow analysis and it was tested in a 2414 bus model of AEP. DTs are constantly updated offline

[3] American Electrical Power (AEP) is one of the biggest utilities in the U.S. serving Arkansas, Indiana, Kentucky, Louisiana, Michigan, Ohio, Oklahoma, Tennessee, Texas, Virginia, and West Virginia.

using a 24 hour load forecasts to better represent the system operating conditions. Every hour a further potential update is done based on actual prevailing conditions. Predictors included current magnitudes, angle differences, MVar flows, the square of voltage magnitudes, and other parameters. The target was labeled as insecure if the load flow solution failed to converge; the system is said to be secure otherwise. Figure 2-13 shows PMU locations used to train and test the proposed scheme.

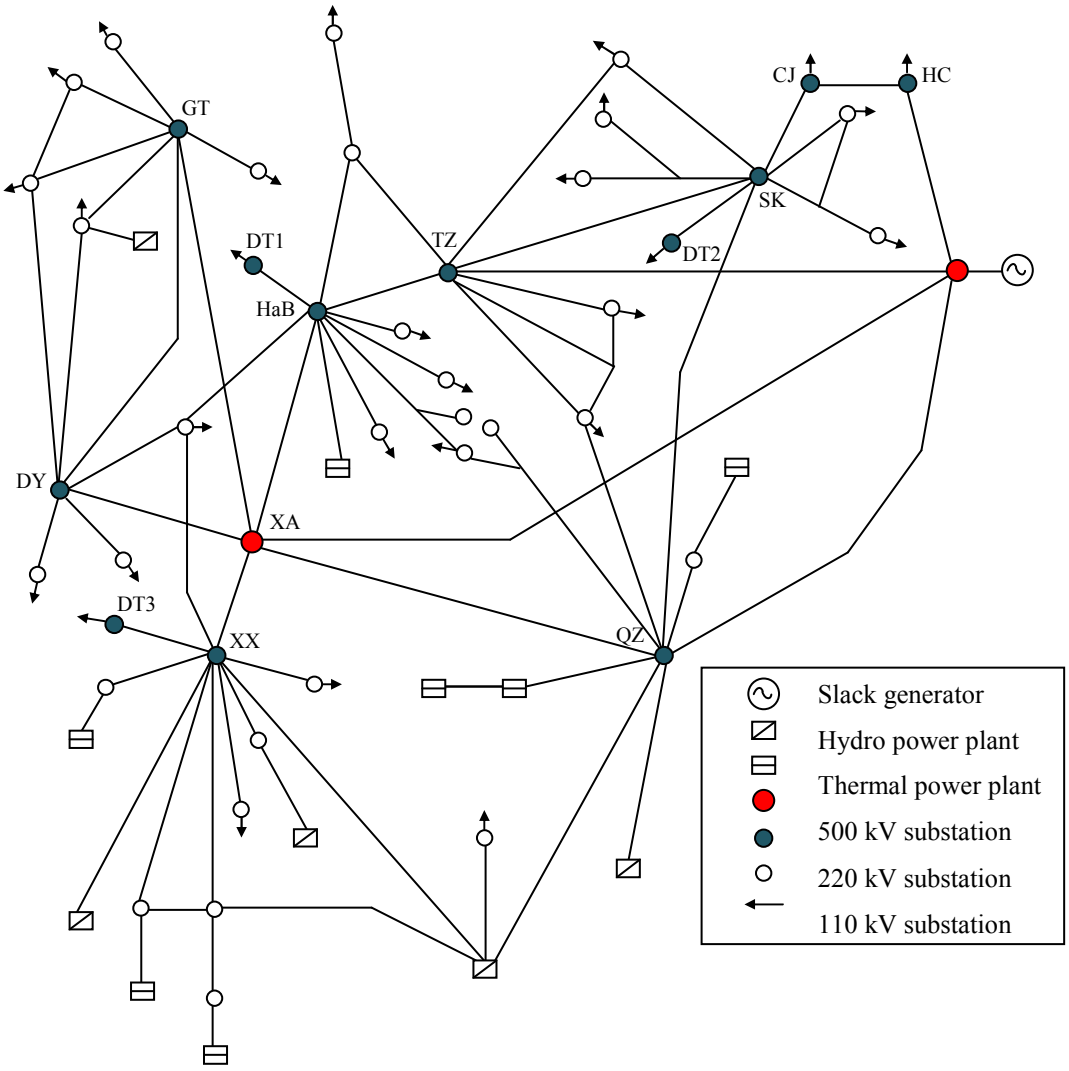

Figure 2-12. One line diagram of Zhenjiang power grid of China [30].

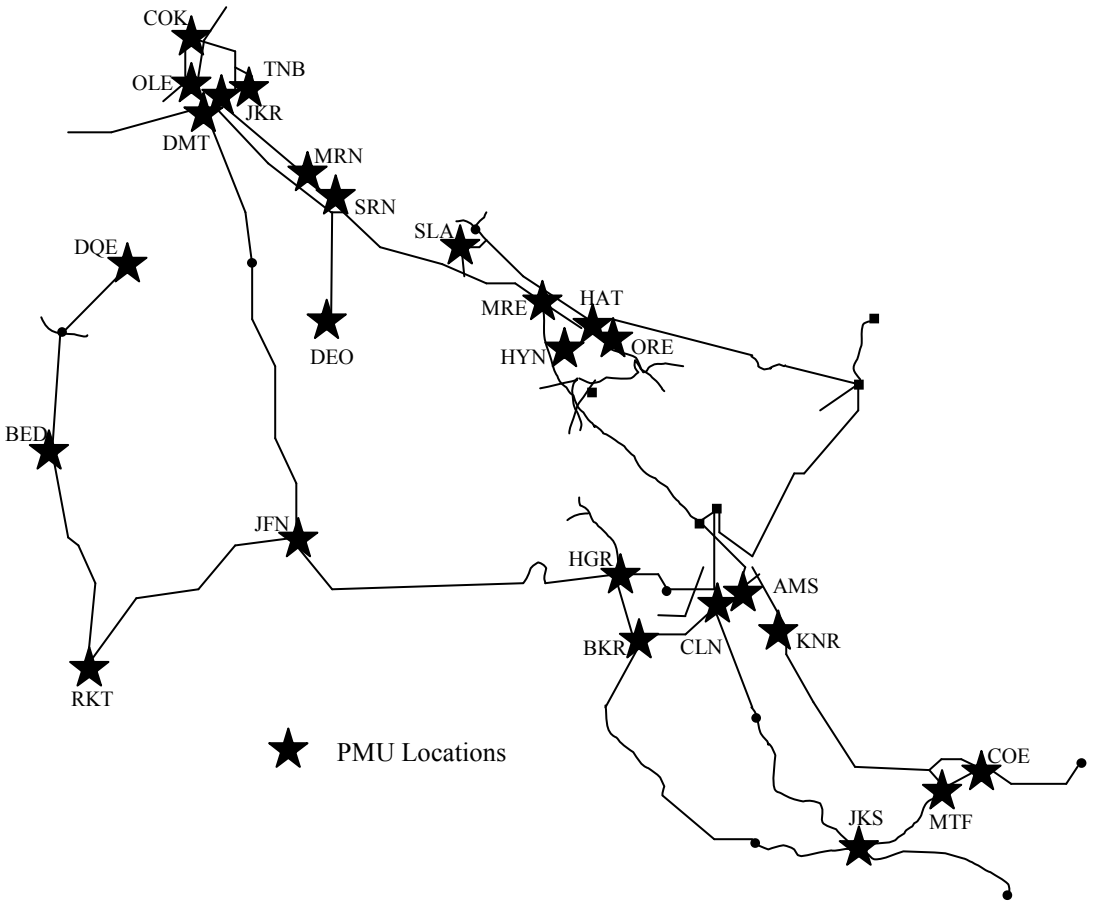

Figure 2-13. PMU locations in AEP system [5].

2.2.2 Data Mining and Protection Systems

Protection systems detect abnormal power system conditions and set off corrective actions in order to restore the system to a normal state. Protection relays are embedded on every stage of the power system: generation, transmission and distribution. Data Mining is particularly suited to stability related protection systems. In general, the design and implementation of such schemes entail a mixture of simulations, empirical experience, and engineering judgment. Data Mining methods can be used to analyze large databases and help in the complex process of deriving settings. In the following sub-section a couple of applications of Decision Trees to an Out-of-Step (OOS) relay and a Special Protection Scheme (SPS) are discussed.

In [27] Decision Trees were used to predict loss of synchronism on the Pacific AC Intertie (PACI). The principle behind the scheme is identical to that of an *R-Rdot* out-of-step

relay [53] and it is based on the fact that the apparent resistance seen by a relay decreases as the angle across the intertie increases. Faults also exhibit the same characteristic of small apparent resistance. The distinction between faults and power swings is achieved by measuring the rate of change of the resistance; in the case of the former it is almost instantaneous, whereas for the latter, it develops over a significantly longer period of time.

An objective function is needed to classify the control signal (target or dependent variable) as a trip or no-trip. The criterion used was a 120 angle difference across the intertie. If the angle difference exceeds the limit the target is classified as a trip; the target is a no-trip otherwise. This criterion was based on empirical evidence and engineering judgment. The predictors used were the aforementioned apparent impedance and its rate of change.

Simulations were performed in a 176 bus system using single and double contingencies of various lengths. A plot of the trajectories followed by *R-Rdot* measurements taken during simulations can be found in [27] (see figure 2 in [27]). The objective of the DT is to identify patterns in the figure in order to predict loss of synchronism. The figure conveys the complexity involved in isolating patterns; a further reason why Data Mining is tailored made for these studies. The trained DT had a very good accuracy failing to predict 4 trip signals and with 1 false trip out of 1600 cases.

Decision Trees have also been successfully applied to the design of Special Protection Schemes (SPS). Hydro-Quebec has established new operation rules, based on DTs, to set an SPS called RPTC (French: Rejet de Production et Teledelestage de Charges) [33]. The scheme pursues to maintain stability by rejecting generation, shedding load, or a combination of both. The scheme reacts to the loss of lines (LOD: Line Opening Detection) or compensation banks (SCB: Series Compensation Bypass). After such events, predefined amounts of load or generation are shed.

Traditionally, deterministic techniques based on worst-case scenarios were used to compute the settings of the RPTCs. The caveat is the lack of optimality; more than necessary load may be shed, or generation rejected. In general, this is an intrinsic disadvantage of any control action based on worst-case scenarios.

Under the new protocol, in order to derive settings for the RPTCs, a wide range of systems states are generated using snapshots of real operating conditions from historical data. In [33], faults were simulated using dynamic simulations and a total of 236 attributes were recorded. Power transfers, capacity margins, voltages and spinning reserve constitute some of the potential attribute candidates. The DT built reduced the average of over-tripped units per case from 2.7 (current practice) to 1.62 (new approach).

Other Data Mining applications to protective relays have also been proposed: transformer protection [54], controlled system islanding [28], and distributed generation islanding [55].

Chapter 3 Data Mining - Decision Trees

Data Mining[4] is defined as the process of "mining" or extracting knowledge from data; the goal is to extract rules or knowledge from regularity patterns exhibited by the data. The field of Data Mining is not associated with a single specific algorithm but it is rather a conglomerate of methods that include among others: parameter associations, correlation analysis, k-Nearest neighbor, neural networks, genetic programming, cluster analysis, classification and regression trees, outlier analysis, etc. Some of these methods are regularly applied in power systems. For example, an outlier analysis algorithm plays a vital role in power system state estimation. The author has worked on a robust outlier detection method based on projection statistics, a cluster analysis technique, to identify and down weight outliers [56].

The advocated Data Mining method used in this dissertation is known as Decision Trees (DTs). Such selection is motivated by the abundance of successful applications of DTs to power systems. A review of several of these applications can be found in Chapter II. Further motivation is provided by the natural simplicity of DTs, its intuitive representation of knowledge discovery, and its outstanding classification accuracy.

The chosen methodology to grow DTs is known as CART (Classification and Regression Trees) [57]. The main focus, due to the nature of our problem, is on classification trees. Despite the fact that no DT algorithm has been found to be superior to all others under any possible situation, CART is the facto technique to grow DTs[5].

In the following sections the principles and algorithms used to build DTs are thoroughly described. The purpose is to present an implementation oriented view of DTs. The chapter is accompanied by a Matlab version of CART; the code can be found in Appendix A.

[4] In the literature, Data Mining is also known as "Knowledge Discovery from Data" (KDD).
[5] A major competitor is the Quinlan Iterative Dichotomizer *C4.5*. Its latest version C5 has become quite similar to the implementation proposed in CART.

3.1 Overview of Decision Trees

A Decision Tree is a form of inductive learning. Given a data set, the objective is to build a model that captures the mechanism that gave rise to the data, i.e., we are not trying to model the data itself but the underlying mechanism that gave rise to the data[6]. The process of constructing the model is a "supervised learning"[7] problem since the training is supervised by an outcome variable called the target.

Decision Trees are grown through a systematic method known as *recursive binary partitioning* [57]; a "divide-and-conquer" approach where successive questions with yes/no answers are asked in order to partition the sample space. Figure 3-1 shows a schematic view of a Decision Tree. The process begins with a "root" node that encloses the learning sample L; the data set that summarizes past experiences. The objective is to recursively partition the sample space L in some clever way so as to extract the knowledge exhibited in data regularity patterns.

At each node t the sample is split into two subsets t_L and t_R, the left and right child respectively. The splitting process is iterated until a terminal node is reached, i.e., a node where no further split is possible. A classification decision is made at such terminal nodes. To classify new data, a route down the tree, from the root node to a terminal node, is found by successively comparing attributes to the DT splitting values.

As shown in Figure 3-1, at each splitting node the sample is typically partitioned into two sub-sets. However, it is possible to have multi-way splits (more than two children). Multi-way splits tend to fragment the data too quickly which leads to a less efficient split on the next level. Furthermore, algorithms to determine binary splits are computationally more efficient and, since any multi-way split can be implemented through a sequence of binary splits, their use is in general discouraged.

[6] The fact that our purpose is to model the mechanism that gave rise to the data is what enables the constructed model to transcend the particular data set used to grow the DT and to evaluate new unforeseen data.
[7] In unsupervised learning methods, the different classes are defined during the analysis. The robust outlier detection method based on cluster analysis and projection statistics [56] is an example of unsupervised learning.

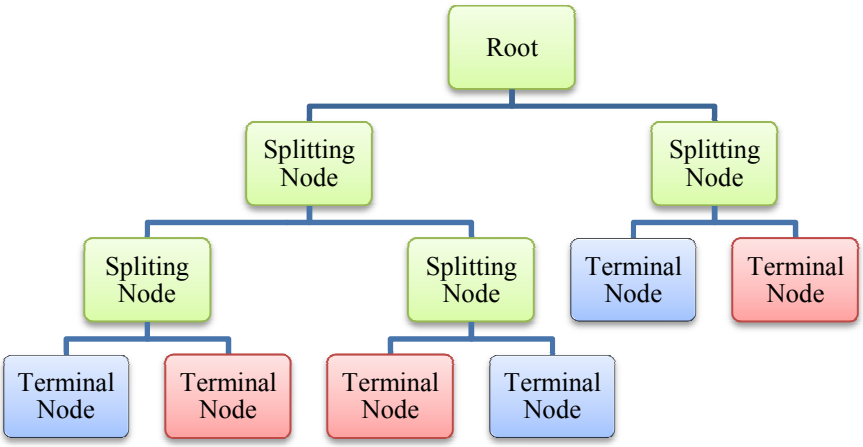

Figure 3-1. Classification Trees – After a sequence of successive sample partitions a classification decision is made at terminal nodes. The red and blue colors represent different classes.

The learning sample L is composed by a set of measurement vectors $X=\{x_1,x_2,\ldots,x_m\}$. Each column of a measurement vector x_i is known as an attribute. Attributes can be of two types: categorical or numerical. Categorical attributes take a finite set of values and do not have an intrinsic order, for example: loading = {low, medium, high}. On the other hand, numerical attributes take values in the real line and therefore have a natural order.

Being a supervised learning method, the class of each vector must be known prior to the data mining process. Therefore, each measurement vector x_i must be classified by the modeler into a set of mutually exclusive classes $C=\{C_1, C_2, \ldots, C_J\}$.

To conclude, the learning sample L is a matrix with m rows (the number of measurement vectors) and $n+1$ columns (the number of attributes on each measurement vector plus the target). Table 3-1 shows a typical arrangement of a learning sample.

Table 3-1. Learning sample matrix with n attributes and m measurement vectors.

	Target	**Attr 1**	**Attr 2**	**Attr 3**	**…**	**Attr n**
x_1	C_j	numerical	categorical	…	…	…
x_2	…	…	…	…	…	…
…	…	…	…	…	…	…
x_m	…	…	…	…	…	…

3.2 Growing Decision Trees

The process of growing DTs is concerned with the following:

- How to choose the splitting attributes?
- How to decide whether a node should be labeled as terminal or if further splitting is necessary?
- How to assign a class to each terminal node?

The process begins at the root node which encloses the learning sample L. The idea is to partition the space into disjoint subsets in a clever way so as to increase the "purity" of such subsets. In our context, purity is understood as a measurement of class homogeneity. Homogeneous nodes that include only one class C_j achieve maximum purity, whereas heterogeneous nodes with an equal proportion of classes $C_{0,\ldots,}C_J$ have minimum purity.

A split is said to be optimal when it maximizes the purity of the descendent nodes. In order to compare potential splitting attributes and threshold values a "goodness of split" criterion needs to be defined. For this purpose, it is convenient and mathematically equivalent to define optimality in terms of node impurity rather than "purity", that is, an optimal split can be equivalently defined as the split that minimizes node impurity. Several functions to measure node impurity have been proposed in the literature: Gini Index, Entropy Impurity, Towing, CHAID, etc. However, empirical results suggest that the choice of a particular impurity function has little effect in the selection of a final tree [57]. The main characteristics of these indices are discussed in the next subsection. The most commonly used index is called "Gini Impurity index" and it is the impurity function used in this dissertation. The Gini Impurity index is defined by:

$$i(t) = 1 - \sum_{j}^{J} p^2\left(C_j \mid t\right) \tag{3.2.1}$$

where $p(C_j|t)$ is an estimator of the probability that a case belongs to class C_j given that it falls into t.

Then, the goodness-of-split criterion of a split s at node t is defined to be the decrease in impurity achieved by split s,

$$\Delta i(s,t) = i(t) - \left[p_L \cdot i(t_L) + p_R \cdot i(t_R) \right] \tag{3.2.2}$$

where $i(t)$ is impurity measurement at node t computed using equation (3.2.1), p_L and p_R are the proportion of cases that fall into the left and right child respectively, and $i(t_L)$ and $i(t_R)$ are the left and right child impurity measurements.

The optimal split $s_{optimal}$ is defined to be the split that maximizes the decrease in impurity in equation (3.2.2). The question now is how to find such optimal split[8], that is, how to select the best strategic splitting attribute and its corresponding threshold. The philosophy under CART's algorithm is quite simple: perform an exhaustive search over all attributes and all possible splitting values.

Consider the set of all attributes $A=\{a_1,a_2,...,a_n\}$ (recall that attributes represent columns of the learning sample L; see Table 3-1). Each attribute $a \in A$ is iteratively selected one at a time. If the selected attribute is numerical, then there are an infinite number of possible splitting values. It is customary, though completely arbitrary, to select the midpoint between two adjacent values as the splitting threshold. If the selected attribute is categorical, then there are a finite number of potential splitting thresholds and they are defined to be the set of unique categories in A.

Let us define $S_a=\{s_1,s_2,...\}$ to be the set of potential splitting values of attribute a. The optimal split s_a of attribute a is the one that maximizes the decrease in impurity expressed by equation (3.2.2). Finally, $s_{optimal}$ at node t is the split that maximizes the decrease in impurity $\Delta i(s,t)$ over all attributes $a \in A$ and splitting values $s \in S_a$.

Following this systematic procedure, the tree is grown by recursively finding optimal splits and partitioning each node into two children. A criterion to declare terminal nodes needs to be defined, that is, how to decide when should the splitting process be stopped. Several stopping

[8] Note that existence is guaranteed since optimality was defined as a relative concept; the split is optimal compared to other attempted splitting values. A proof of uniqueness can be found in 57. *Classification and regression trees*. 1984, Belmont, Calif: Wadsworth International Group. x, 358 p.

rules have been proposed in the literature but none has proved to be competent [57]. For example, an heuristic stopping rule could be to label a node terminal if no significant decrease in impurity can be achieved, i.e., stop if $\Delta i\left(t, s_{optimal}\right) > \varepsilon$. However, an appropriate selection of the threshold ε is not a trivial task. A small ε is bound to render trees that are too large which leads to an over-fitting problem. If ε is too big, the splitting process may be stopped too soon. Furthermore, an intermediate split may not significantly reduce $\Delta i\left(t, s\right)$ and yet the classification accuracy may be drastically improved in subsequent partitions.

The proposed solution to the stopping rule problem is not to have a stopping rule at all. CART's algorithm initially grows a tree as large as possible. A node is considered to be terminal if it has achieved zero impurity (maximum class homogeneity, only a unique class remains) or if the total number of measurement vectors x_i at node t is less than some predetermined value n_{min}[9].

Ones a maximum sized tree T_{max} has been grown, the tree is selectively pruned upwards. Branches are systematically pruned based on a cost-complexity criterion. The pruning algorithm is thoroughly discussed in section 3.1.4.

Finally, as stated previously, a classification decision is made at terminal nodes. Class C_j is assigned to terminal node t if $p(C_j|t)$ is the largest,

$$p\left(C_j \mid t\right) = \max_i \left(p\left(C_i \mid t\right)\right) \tag{3.2.3}$$

3.2.1 Pseudo-algorithm: growing T_{max}.

To conclude, the algorithm to grow T_{max} can be summarized by the following steps:

1) The learning sample L is an m by $n+1$ matrix[10], where m is equal to the number of vector measurements and n is the number of attributes.
2) Determine the minimum number of elements n_{min} to be allowed in a terminal node (typically, 5 or 10).

[9] The selection of n_{min} is arbitrary. Typical values suggested in the literature are 5 or 10. However, it is possible to set n_{min} equal to one and grow a maximum sized tree.
[10] The extra column is the target or dependent variable.

3) From the set of all attributes $A=\{a_1,a_2,\ldots,a_n\}$ select attribute $a \in A$.

If a is of type numeric, sort it in ascending order and define the set $S_a=\{s_1,s_2,\ldots\}$ of potential splitting values for attribute a as the midpoint between two adjacent measurements:

$$S_a(k) = \frac{x_a(k+1) + x_a(k)}{2} \tag{3.2.4}$$

where $x_a(k)$ represents attribute a of vector measurement k, with $k=1,\ldots,m$.

If a is of type categorical, define the set $S_a=\{s_1,s_2,\ldots\}$ of potential splitting values for attribute a as the set of unique categories in a.

4) For each splitting value $s \in S_a$ partition the learning sample L at node t into two disjoint subsets t_L and t_R; these are the left and right child of node t.

If a is numerical, then,

$$t_L = \{x_a(k) \ if \ x_a(k) \le s(k)\} \tag{3.2.5}$$

$$t_R = \{x_a(k) \ if \ x_a(k) > s(k)\} \tag{3.2.6}$$

If a is of type categorical,

$$t_L = \{x_a(k) \ if \ x_a(k) = s(k)\} \tag{3.2.7}$$

$$t_R = \{x_a(k) \ if \ x_a(k) \ne s(k)\} \tag{3.2.8}$$

5) Compute the decrease in impurity achieved by split s at node t using equation (3.2.2):

$$\Delta i(s,t) = i(t) - \left[p_L \cdot i(t_L) + p_R \cdot i(t_R) \right] \tag{3.2.2}$$

where $i(t)$ is the impurity measurement at node t, p_L and p_R are the proportion of cases that fall into the left and right child respectively, and $i(t_L)$ and $i(t_R)$ are the left and right child impurity measurements.

The impurity function is known as the Gini index:

$$i(t) = 1 - \sum_{j}^{J} p^2\left(C_j \mid t\right)$$

The left and right proportions can be computed by:

$$p_L = \frac{n\left(t_L\right)}{n\left(t\right)} \tag{3.2.9}$$

$$p_R = \frac{n\left(t_R\right)}{n\left(t\right)} \tag{3.2.10}$$

where $n(t)$ is the total number of vector measurements at node t and $n(t_L)$, $n(t_R)$ are the total number of vector measurements that fall to the left and right node respectively.

The estimator $p(C_j|t)$ is calculated by,

$$p\left(C_j \mid t\right) = \frac{p\left(C_j, t\right)}{p\left(t\right)} \tag{3.2.11}$$

where $p(C_j,t)$ is defined as the resubstitution estimator of the probability that a case belongs to class C_j and that it falls to node t, and $p(t)$ is the estimator of the probability that a case falls into node t,

$$p\left(C_j, t\right) = \pi\left(C_j\right) \cdot \frac{n_t\left(C_j\right)}{n\left(C_j\right)} \tag{3.2.12}$$

$$p\left(t\right) = \sum_{j}^{J} p\left(C_j, t\right) \tag{3.2.13}$$

The function $n_t(C_j)$ denotes the number of class C_j cases at node t, $n(C_j)$ is the total number of cases that belong to class C_j, and $\pi(C_j)$ is the so called prior probability and it is either provided by the modeler or estimated from the data,

$$\pi\left(C_j\right) = \frac{n\left(C_j\right)}{n} \tag{3.2.14}$$

6) The optimal split $s_{optimal}$ at node t is defined as the split that maximizes the decrease in impurity $\Delta i(s,t)$ in equation (3.2.2) over all attributes $a \in A$ and splitting values $s \in S_a$.
7) Repeat steps 2 through 5 until no more splits are possible, i.e., terminal nodes contain a single class C_j or the minimum number of elements in a node n_{min} has been achieved.
8) A classification decision is made at terminal nodes. Class C_j is assigned to terminal node t if $p(C_j|t)$ is the largest,

$$p\left(C_j \mid t\right) = \max_i \left(p\left(C_i \mid t\right)\right) \tag{3.2.3}$$

The Matlab code, *growTmax()*, can be found in Appendix A.1.2.

3.2.2 Experiment 1: an example with simulated data

For this experiment, a sample of 200 measurement vectors with two attributes, a_1 and a_2, was generated using a mixture of bivariate uniform random variables. There are two classes labeled as a 1 or a 0; red and blue dots respectively. Figure 3-2 shows a scatter plot of the learning sample L.

The maximum sized tree is grown using the function *growTmax()*; see code attached in Appendix A. Figure 3-3 shows the maximum sized tree T_{max}. Green nodes represent splitting nodes and terminal nodes are color coded, blue and red, to identify the different class assignment. Matlab's cursor can be used to display detailed node information. For example, node number 4, a splitting node, tests if attribute is less a_2 than or equal to 0.8323. Measurement vectors x_i falls into the left child if $a_2 \leq 0.8323$; x_i falls into the right child otherwise. Node number 9 is a terminal node and cases that fall into this node are classified as class 0. The cursor also displays information regarding the total number of cases in the learning sample that belong to each class.

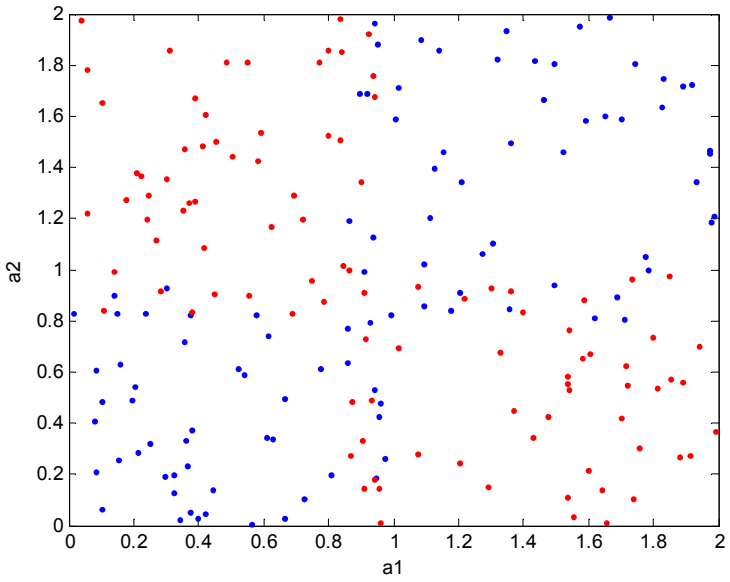

Figure 3-2. Scatter plot of attributes a_1 and a_2. The red and blue colors indicate different classes.

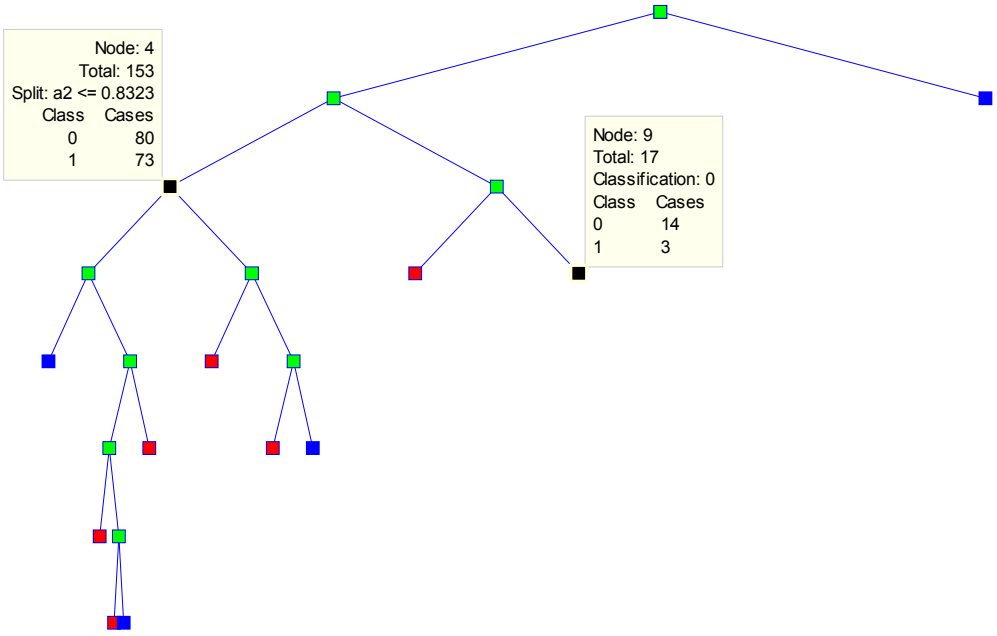

Figure 3-3. Maximum sized tree T_{max}. Green nodes represent splitting nodes. Terminal nodes are color coded to identify the final classification. Matlab's cursor can be used to display node information.

3.2.3 Splitting Rules

The impurity function to assess node heterogeneity used throughout this dissertation is known as the *Gini Impurity* index:

$$i(t) = 1 - \sum_{j}^{J} p^2(C_j \mid t) \tag{3.2.1}$$

where $p(C_j|t)$ is the estimated probability of a case belonging to class C_j given that it falls into node t. The index can be interpreted as the estimated probability of misclassification given that a classification C_j is made randomly from the class distribution present at node t. It can also be interpreted as a sum of variances of Bernoulli trials.

The function achieves a maximum (maximum impurity) when the classes C_j are equally mixed. If only one class is present at node t, the impurity is equal to zero. It is a continuous and strictly concave function.

In the case of a two class problem, the Gini Impurity function can be reduced to:

$$i(t) = 2 \cdot p(C_0 \mid t) \cdot p(C_1 \mid t) \tag{3.2.15}$$

Proof: since there are only two mutually exclusive classes, the sum of their probabilities must add up to one, i.e.,

$$1 = p(C_0 \mid t) + p(C_1 \mid t) \tag{3.2.16}$$

Taking the square of each side,

$$1^2 = \left(p(C_0 \mid t) + p(C_1 \mid t)\right)^2 \tag{3.2.17}$$

$$1 = p^2(C_0 \mid t) + p^2(C_1 \mid t) + 2 \cdot p(C_0 \mid t) \cdot p(C_1 \mid t) \tag{3.2.18}$$

By simple algebraic manipulation,

$$1 - p^2(C_0 \mid t) - p^2(C_1 \mid t) = 2 \cdot p(C_0 \mid t) \cdot p(C_1 \mid t) \tag{3.2.19}$$

The left term is the Gini index as expressed by equation (3.2.1) and it is equivalent to equation (3.2.15).

Another commonly used impurity function is the so called *Entropy Impurity* [58]:

$$i_{entropy}(t) = -\sum_{j} p_{C_j} \cdot \log\left(p_{C_j}\right) \tag{3.2.20}$$

$$p_{C_j} = \frac{n\left(C_j\right)}{n(t)} \tag{3.2.21}$$

Entropy Impurity is a basic concept in information theory. It is the main splitting criterion used by the algorithm C4.5; a major competitor of CART. Figure 3-4 shows a plot of the Gini and Entropy functions for a two class problem. To facilitate the comparison between the indices, both impurity functions have been properly scaled. It can be seen that the behavior of both functions is quite similar and empirical results [57] suggest that the splitting criterion used has little influence in the selection of a final decision tree.

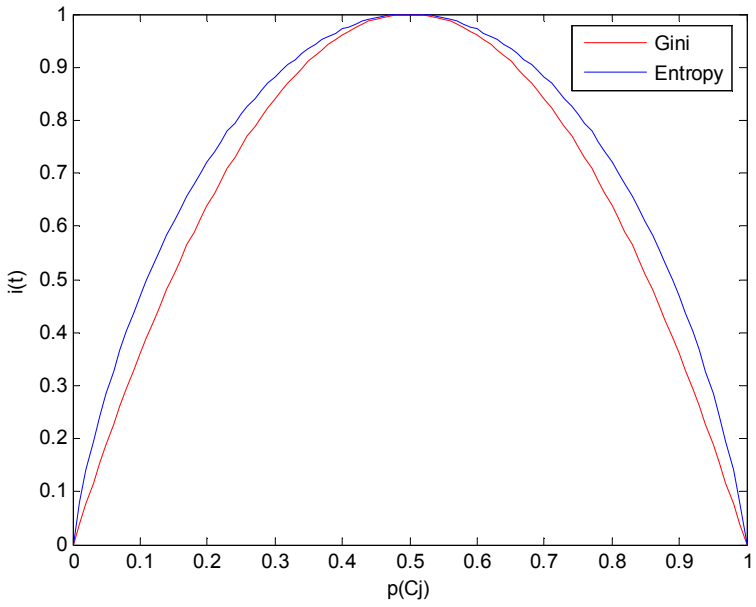

Figure 3-4. Comparison between Gini and Entropy impurity functions. Both functions reach a maximum when classes are equally mixed and achieve a minimum when only one class is present.

Finally, a third commonly used splitting criterion for multiclass problems is the *Towing* index [57]. The Towing index is not an impurity measurement per se; it does not measure node heterogeneity.

$$i_{towing}(t) = \frac{n(t_L) \cdot n(t_R)}{4} \cdot \sum_{j}^{J} \left(p(C_j \mid t_L) - p(C_j \mid t_R) \right)^2 \tag{3.2.22}$$

The main objective of the Towing index is class separation. The function tries to associate groups of classes and pursues to maximize the difference between the probability that class C_j goes to t_L and the probability that it falls to t_R. In essence, every multi-class split is treated as a two-class problem and a conglomeration of classes is made to partition the set into two super-classes. An advantage of this method is that it can reveal class resemblances by grouping classes with similar characteristics. The Towing index can be shown to be mathematically equivalent to the Gini index in the case of a two-class problem.

3.2.4 Competitors and Surrogate Splits

Competitors of $s_{optimal}$ at node t can easily be determined during the exhaustive search for the primary split. The first competitor of $s_{optimal}$ is the attribute which is second best at maximizing the decrease of impurity in equation (3.2.2). The idea is to save, in order of merit, a list with the top five splitting attributes.

Surrogates, on the other hand, attempt to maximize the predictive association with the primary split. The objective of a surrogate is to mimic the primary split by partitioning the sample space in a way such that the actual measurements that fall into the left and right node are as similar as possible to those of the primary split. The concept of a surrogate is very close to the notion of correlation in a linear regression.

Let s' be a potential split that partitions node t into t'_L and t'_R. Let $n(C_j,LL)$ be the number of cases that belong to the intersection $t'_L \cap t_L$, that is, the number of cases in t that both $s_{optimal}$ and s' send to the left child. Then the probability that a case falls into $t'_L \cap t_L$ is,

$$p\left(t'_L \cap t_L\right) = \sum_j \pi\left(C_j\right) \cdot \frac{n\left(C_j, LL\right)}{n\left(C_j\right)} \tag{3.2.23}$$

The probability that both splits, $s_{optimal}$ and s', send a case in node t to the left child is,

$$p_{LL}\left(s_{optimal}, s'\right) = \frac{p\left(t_L \cap t'_L\right)}{p\left(t\right)} \tag{3.2.24}$$

The probability $p_{RR}(s_{optimal},s')$ that $s_{optimal}$ and s' send a case in node t to the right child is computed in a similar way.

Then, the probability that s' correctly predicts the actions of $s_{optimal}$ is,

$$p\left(s_{optimal}, s'\right) = p_{LL}\left(s_{optimal}, s'\right) + p_{RR}\left(s_{optimal}, s'\right) \tag{3.2.25}$$

The surrogate split $s_{surrogate}$ of the optimal split $s_{optimal}$ is defined as the split that most accurately predicts the actions of $s_{optimal}$, i.e., the split that maximizes equation (3.2.25),

$$p\left(s_{optimal}, s_{surrogate}\right) = \max_{s'}\left(p\left(s_{optimal}, s'\right)\right) \tag{3.2.26}$$

Surrogates can be used in situation when the measurement of the primary split is missing. They also reveal attribute associations and potential masking effects[11]. The fact that an attribute is not used in a final classification tree T_k does not mean that it is not relevant in the decision process. Surrogates with high predictive association identify attributes that, despite not being used in the tree, can efficiently extract the information exhibited in data regularity patterns.

3.3 Minimal Cost-Complexity Pruning

As stated previously, several stopping rules were initially proposed in an attempt to guide the tree growing process in the search for a right sized tree, that is, a tree with optimal classification accuracy. However, such approach proved to be extremely inefficient since the

[11] Unlike linear regressions, highly correlated variables present no challenge for decision trees. "Multi-collinearity" can significantly bias the estimation of the variance of highly correlated regressors.

growing algorithm was either stopped to soon or it rendered an overgrown tree which is likely to over-fit the data.

CART's breakthrough was to devoid the tree growing methodology from any kind of stopping rule. The first step is to grow a maximum sized tree T_{max} following the procedure described in section 3.1.2. Needless to say, this maximum sized tree is likely to suffer from an over-fitting problem. The underlying hypothesis is that a right sized tree can be discovered by subsequently removing the less reliable branches of the tree.

The systematic algorithm to develop a sequence of smaller size subtrees $\{T_1 > T_2 > \ldots > T_{root}\}$ is known as minimal cost-complexity pruning. Cost-complexity is understood as a function that measures the tradeoff between error rates and tree sizes. For a fixed complexity (tree size), the objective is to find the subtree T_k with the lowest misclassification rate, i.e., the tree T_k that minimizes the cost-complexity function.

Let us define $R^*(T)$ as the probability that tree T will misclassify a new sample drawn from the same distribution of the learning sample L; $R^*(T)$ is the so called "true misclassification rate".

An estimator of $R^*(T)$ can be constructed as the proportion of cases in L misclassified by T. Such estimator is known as the resubstitution estimator $R(T)$. This estimator tends to be highly inaccurate and over optimistic due to the double job attributed to the learning sample; L is used both to grow T and to assess its misclassification rate. Since all DT algorithms, directly or indirectly, attempt to minimize $R(T)$, it can be shown[12] that $R(T)$ decreases after any partition of the space, i.e., for a split s at node t,

$$R(t) \geq R(t_L) + R(t_R) \tag{3.3.1}$$

Therefore, $R(T)$ decreases as the number of splits increases. In fact, if T is grown until each terminal node contains only one case, then the misclassification rate $R(T)$ is equal to zero. This phenomenon is somewhat similar to the effect that adding any variable has on the estimated

[12] A proof can be found in 57. *Classification and regression trees*. 1984, Belmont, Calif: Wadsworth International Group. x, 358 p..

R^2 of a linear regression model. It is well known that the estimated R^2 increases regardless of the explanatory power of the added variable.

Let us define *r(t)* to be the resubstitution estimator of the probability of misclassification given that a measurement vector x_i falls into node t,

$$r(t) = 1 - \max_j \left(p\left(C_j \mid t\right)\right) \tag{3.3.2}$$

where $p(C_j|t)$ is the resubstitution estimator of the probability of having class C_j given that we are at node t.

Then, define *R(t)* as,

$$R(t) = r(t) \cdot p(t) \tag{3.3.3}$$

Then the resubstitution estimator *R(T)* of the true misclassification rate $R^*(T)$ for a tree T can be computed by,

$$R(T) = \sum_{t \in T_{\text{Terminal}}} R(t) \tag{3.3.4}$$

where $T_{Terminal}$ is the set of terminal nodes of tree T. Despite the fact that *R(T)* is not a good estimator of the true misclassification rate $R^*(T)$, it is a natural criterion to compare subtrees of the same size.

Let us define the complexity of tree T to be the number of terminal nodes $n(T_{Terminal})$ of tree T. Then the cost-complexity function $R_\alpha(T)$ for tree T is defined as the sum of the misclassification rate and a cost penalty for complexity,

$$R_\alpha(T) = R(T) + \alpha \cdot n\left(T_{\text{Terminal}}\right) \tag{3.3.5}$$

The parameter α determines the cost assigned to tree complexity. If α is set equal to zero, no penalty cost is assigned to tree size and, since *R(T)* decreases with the number of splits, the minimal complexity-cost tree T will be equal to T_{max}. As α increases, there will be an inflection point α_1 where the cost-complexity of subtree T_1 will be lower than the cost-complexity of T_{max}. The pruning algorithm works by removing the weakest branch of T_{max} and deriving subtree T_1.

As α keeps increasing, another inflection point α_2 will be reached and the weakest branch of T_1 is pruned. This process is iterated until a minimum sized tree T_{root} , a tree with only one splitting node and two childrenand, is attained. A sequence of minimal cost-complexity subtrees $\{T_1 > T_2 > \ldots > T_{root}\}$ is achieved in this manner.

Still one question remains, which subtree in the sequence is the "right" sized tree. A natural selection criterion would be to use the true misclassification error rate $R^*(T)$. The right sized tree T_k is the subtree in the sequence with minimum error rate. An estimator of $R^*(T)$ known as V-fold cross validation is thoroughly described in section 3.1.5.

3.3.1 Pseudo-algorithm: cost-complexity pruning.

To conclude, the algorithm for cost-complexity pruning can be summarized by the following steps:

1) Starting from a fully grown tree T_{max}, if for any node t,

   $$R(t) = R(t_L) + R(t_R)$$

   then, prune t_L and t_R. The resultant subtree T_1 is saved.[13]

2) For every splitting node t in subtree T_k , compute the function $f(t)$:

$$f(t) = \frac{R(t) - R(t_D)}{n(t_D) - 1} \tag{3.3.6}$$

   were t_D is the set of descendant terminal nodes of node t, $n(t_D)$ is the number of terminal nodes that descend from node t, and $R(t)$ and $R(t_D)$ are computed using equations (3.3.3) and (3.3.4) respectively,

$$R(t) = r(t) \cdot p(t) \tag{3.3.3}$$

$$R(T) = \sum_{t \in T_{\text{Terminal}}} R(t) \tag{3.3.4}$$

3) The weakest link is defined as the node $t_{weakest}$ such that,

[13] It can be shown that $R(t) \geq R(t_L) + R(t_R)$

$$f\left(t_{weakest}\right) = \min_{t \in T_k}\left(f\left(t\right)\right) \tag{3.3.7}$$

4) Prune off the descendants of $t_{weakest}$ and repeat step 2 until the minimum sized tree is achieved, i.e., the root node and two terminal nodes.

The Matlab code, *CostComplexityPruning()*, can be found in Appendix A.1.3.

3.3.2 Experiment 1: continued

In section 3.1.2 a maximum sized tree T_{max} was grown using a learning sample generated through a mixture of bivariate uniform random variables. Using the function *CostComplexityPruning()*, a sequence of minimal cost-complexity subtrees is created.

Figure 3-5 shows the sequence of subtrees $\{T_1 > T_2 > T_3 > T_4 > T_5 > T_{root}\}$ generated by cost-complexity pruning. It can be seen that as the penalty cost α assigned to tree size increases, the subtree that minimizes the cost-complexity function has fewer terminal nodes. The algorithm works by subsequently pruning the weakest, less reliable, branches in the tree.

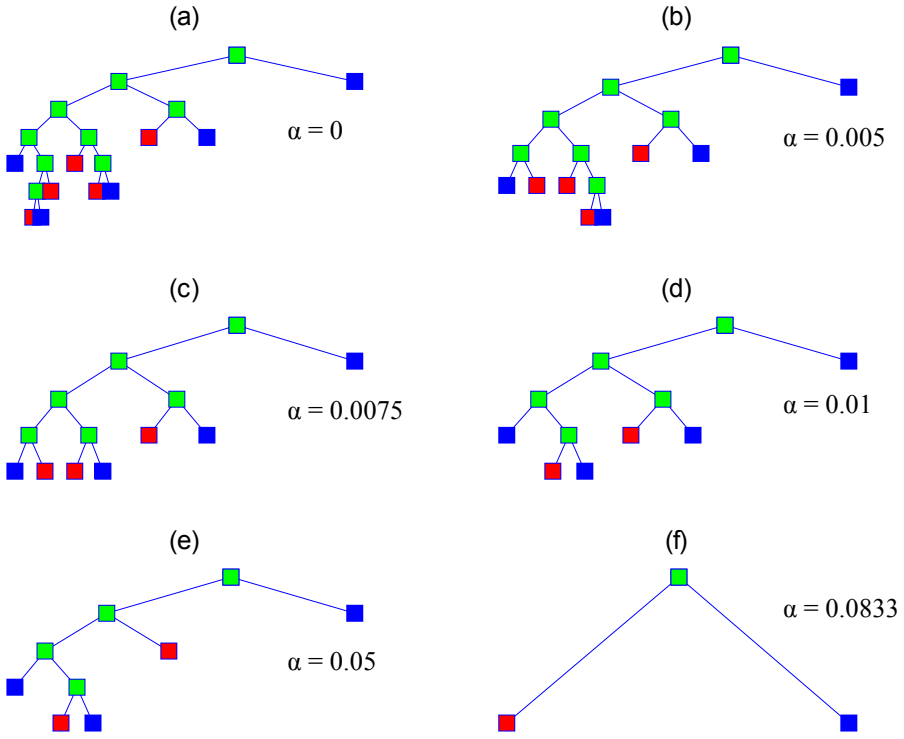

Figure 3-5. Sequence of minimal cost-complexity subtrees.

3.4 Optimal Subtree: V-fold Cross Validation

The pruning algorithm presented in section 3.3 generates a sequence of minimum sized subtrees in terms of cost-complexity. The objective now is to select the "right" sized tree; the optimal subtree. A natural criterion would be to select the subtree with the lowest true misclassification rate $R^*(T_k)$. As discussed previously, the resubstitution estimator $R(T_k)$ decreases as the size of the tree increases. Therefore, if $R(T)$ were to be used as the selection criterion, the largest sized tree would always be chosen.

An "honest"[14] estimator $\hat{R}(T_k)$ of the true misclassification rate $R^*(T_k)$ can be obtained either by:

- Using an independent test sample.
- V-fold cross-validation.

An independent test sample is constructed by partitioning the learning sample L into a new training sample L_1 and a test sample L_2. Care must be taken to ensure that L_2 is an IID sample of L. The trees T_{max} and the sequence of subtrees $\{T_1 > T_2 > ... > T_{root}\}$ are grown using L_1. Then the cases in L_2 are used to assess the misclassification rate of each subtree.

Let us define the probability that a vector measurement x_i belonging to class C_j is classified into class C_i by subtree T_k as,

$$Q^{TS}(C_i \mid C_j) = \frac{n^{(L_2)}(C_i \mid C_j)}{n^{(L_2)}(C_j)} \tag{3.4.1}$$

where $n^{(L_2)}(C_j)$ is the number of cases belonging to class C_j in L_2, and $n^{(L_2)}(C_i \mid C_j)$ is the number of cases that were misclassified as class C_i given that their true class is C_j.

The expected cost of misclassification for class C_j is[15],

[14] Concepts like unbiasedness, efficiency, and consistency require probabilistic assumptions. Since we have not yet explicitly made any probabilistic assumption the word "honest" is used.

[15] Equation (3.4.2) can be easily weighted by a misclassification cost. With unity misclassification cost, equation (3.4.2) is simply the proportion of class C_j test cases misclassified by T_k.

$$R^{TS}\left(C_j\right) = \sum_i Q^{TS}\left(C_i \mid C_j\right) \tag{3.4.2}$$

Then, the expected misclassification cost for tree T_k is[16],

$$R^{TS}\left(T_k\right) = \sum_j R^{TS}\left(C_j\right) \cdot \pi\left(C_j\right) \tag{3.4.3}$$

where $\pi\left(C_j\right)$ is the prior probability and it is determined by the modeler or estimated as the ratio of the number of cases in L_2 with class C_j over the total number of cases in L_2.

The optimal tree $T_{optimal}$ is the tree that minimizes equation (3.4.3), i.e.,

$$R^{TS}\left(T_{optimal}\right) = \min_k\left(R^{TS}\left(T_k\right)\right) \tag{3.4.4}$$

The major disadvantage of using a test sample is that the sample size used to train the DT is reduced. Typically, one third of the measurement vectors are set aside to be used as an independent test sample. If sample size is a concern, the method known as V-fold cross validation is preferred.

With V-fold cross validation, the complete learning sample L is used to grow a maximum sized tree T_{max} and its sequence of minimal cost-complexity subtrees $\{T_1 > T_2 > \ldots > T_{root}\}$. Then, the learning sample L is divided into V subsets $\{L_1, \ldots, L_V\}$ with an approximately equal number of cases in each one of them; random selection or a stratified selection[17] can be used. Typically, a value of 10 is used for V, that is, L is partitioned into 10 subsets.

Let us define the v^{th} learning sample $L^{(v)}$ as,

$$L^{(v)} = L - L_v \qquad v = 1, \ldots, V \tag{3.4.5}$$

then, the learning sample $L^{(v)}$ contains 9/10 of the total cases in L.

For each learning sample $L^{(v)}$ a maximum sized tree $T^{(v)}{}_{max}$ is grown and the sequence of minimal cost-complexity subtress $\{T^{(v)}{}_1 > T^{(v)}{}_2 > \ldots > T^{(v)}{}_{root}\}$ is determined. Since the subset L_v has

[16] With unity misclassification cost, equation (3.4.3) is the total proportion of test cases misclassified by T_k.
[17] In stratified cross-validation, the folds are strategically designed so that the class distribution of each subset L_v resembles the distribution of the original sample L.

been excluded from the growing process it can be used as an independent test sample for tree $T^{(v)}$. However, keep in mind that the main objective is to estimate the true misclassification rate $R^*(T_k)$ for the original sequence of subtrees $\{T_1 > T_2 > \ldots > T_{root}\}$ grown with the complete learning sample L.

Depending on the stability of the tree, the sequence of subtress grown using $L^{(v)}$ may or may not be similar to the sequence of subtress grown with L. The proposed solution is to use the subtree in $\{T^{(v)}_1 > T^{(v)}_2 > \ldots > T^{(v)}_{root}\}$ that better resembles the cost-complexity of T_k. Recall, from section 3.1.4, that inflection points determined when braches were pruned. Therefore, for tree T_k the complexity parameter α has a range between $\alpha_k \leq \alpha < \alpha_{k+1}$. Compute the geometric midpoint α'_k of the interval by,

$$\alpha'_k = \sqrt{\alpha_k \cdot \alpha_{k+1}} \tag{3.4.6}$$

Then the equivalent cost-complexty subtree of T_k is defined as the subtree $T^{(v)}_{(\alpha'_k)} \in \left\{T^{(v)}_1 > \ldots > T^{(v)}_{root}\right\}$ whose complexity parameter range $\alpha^{(v)}_k \leq \alpha^{(v)} < \alpha^{(v)}_{k+1}$ encloses α'_k. The misclassification rate $R^{CV}\left(T^{(v)}_{(\alpha'_k)}\right)$ is then used as an equivalent measurement of the error rate $\hat{R}\left(T_k\right)$ of subtree T_k.

Let $n^{(v)}\left(C_i \mid C_j\right)$ be the number of cases that are misclassified as C_i by tree $T^{(v)}_{(\alpha'_k)}$ given that they belong to class C_j. Then set,

$$n\left(C_i \mid C_j\right) = \sum_v^V n^{(v)}\left(C_i \mid C_j\right) \tag{3.4.7}$$

Finally, an estimator of the true misclassification rate for tree T_k is obtained using the same equations used for the independent test sample.

$$Q^{CV}\left(C_i \mid C_j\right) = \frac{n\left(C_i \mid C_j\right)}{n\left(C_j\right)} \tag{3.4.8}$$

$$R^{CV}\left(C_j\right) = \sum_i Q^{CV}\left(C_i \mid C_j\right) \tag{3.4.9}$$

$$R^{CV}\left(T_k\right) = \sum_j R^{CV}\left(C_j\right) \cdot \pi\left(C_j\right) \tag{3.4.10}$$

The optimal subtree $T_{optimal}$ is the tree with the lowest misclassification cost.

$$R^{CV}\left(T_{optimal}\right) = \min_k \left(R^{CV}\left(T_k\right)\right) \tag{3.4.11}$$

3.4.1 Experiment 1: continued.

In section 3.3 a sequence of subtrees $\{T_1 > T_2 > T_3 > T_4 > T_5 > T_{root}\}$ was created through cost-complexity pruning. The objective is to identify the right sized tree in the sequence. The true misclassification rate $R^*(T)$ is a natural criterion to select the optimal subtree, that is, choose the subtree T_k with the lowest error rate. As stated previously, $R(T)$ is not a good estimator of $R^*(T)$. Therefore, the method known as V-fold cross validation is used to estimate $R^*(T)$ for each subtree.

Figure 3-6 shows a plot of the estimated misclassification rate $R^{CV}(T)$ for each subtree in the sequence. As indicated in the plot, the optimal subtree T_5 has seven terminal nodes and a misclassification rate of 0.19; a schematic of subtree T_5 is shown in Figure 3-7. Decision Trees typically exhibit a characteristic similar to that shown in Figure 3-6. Initially the error rate declines significantly as the number of splits increases. Then a valley of minimal misclassification rate is reached. Finally, as the number of terminal nodes keeps increasing, the error rate begins to increase due to the so called over-fitting problem; very large trees are likely to make predictions beyond what is warranted by the data.

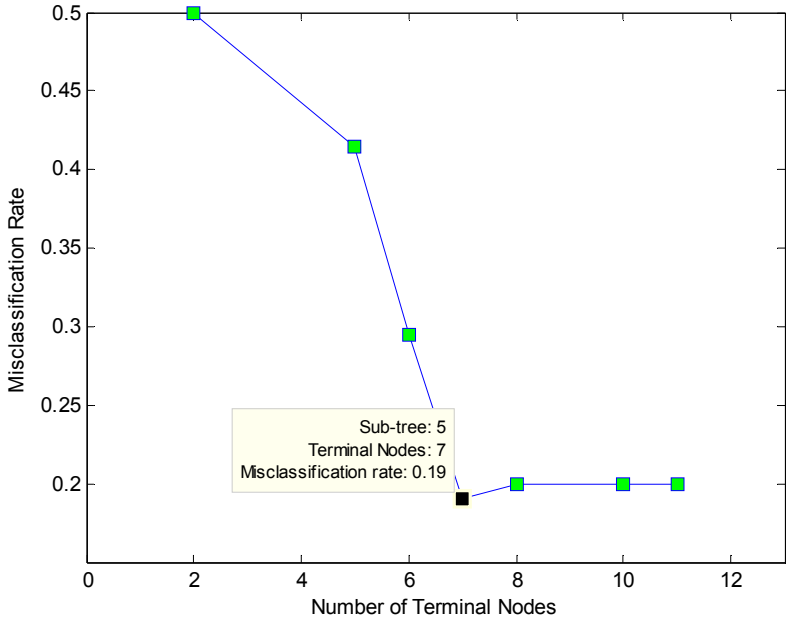

Figure 3-6. V-fold cross-validation misclassification error rates.

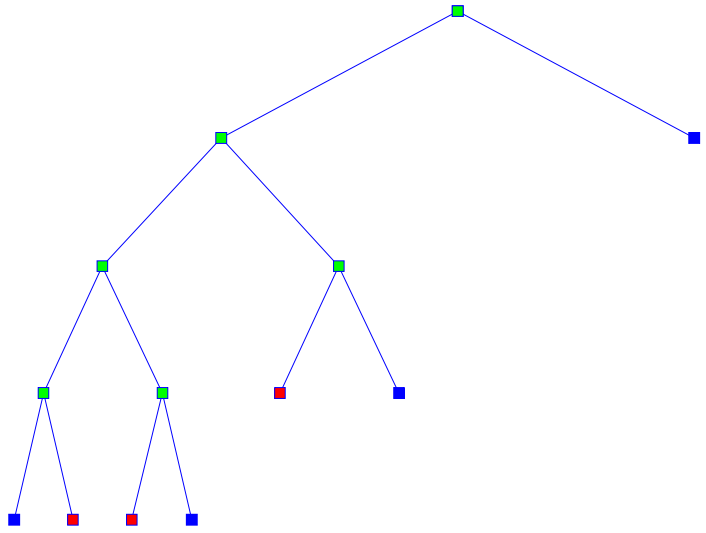

Figure 3-7. Optimal subtree T_5. The tree achieves the minimum error rate.

In order to compare and validate the misclassification rates obtained with V-fold cross validations a Monte Carlo experiment was designed. The derived error probability of 0.19 for T_5 implies that, in the long run, we expect to misclassify 19% of the cases presented to the tree. Using the same stochastic mechanism, 10000 replications with a sample size of a 100 were

generated and dropped down each subtree in the sequence. The error rates were recorded and they represent the true misclassification rate $R^*(T)$.

Figure 3-8 depicts a comparison between the misclassification rates estimated using V-fold cross validation $R^{CV}(T)$, the true error rate $R^*(T)$ estimated using Monte Carlo simulations, and the resubstitution estimator $R(T)$. As expected, $R(T)$ is not an honest estimator of $R^*(T)$ and it decreases as the number of terminal nodes increases. On the other hand, the estimates $R^{CV}(T)$ obtained with V-fold cross validation are right on target. It can be shown that V-fold cross validation has a negative bias and they always underestimate $R^*(T)$ [57]; $R^{CV}(T)$ is a conservative estimator of $R^*(T)$.

Table 3-2 shows the results obtained from the simulation. The values in parenthesis represent standard deviations. Figure 3-9 depicts a histogram with the simulated error probabilities for tree T_5. It resembles a Gaussian distribution with mean 0.1885 and a standard deviation of 0.038.

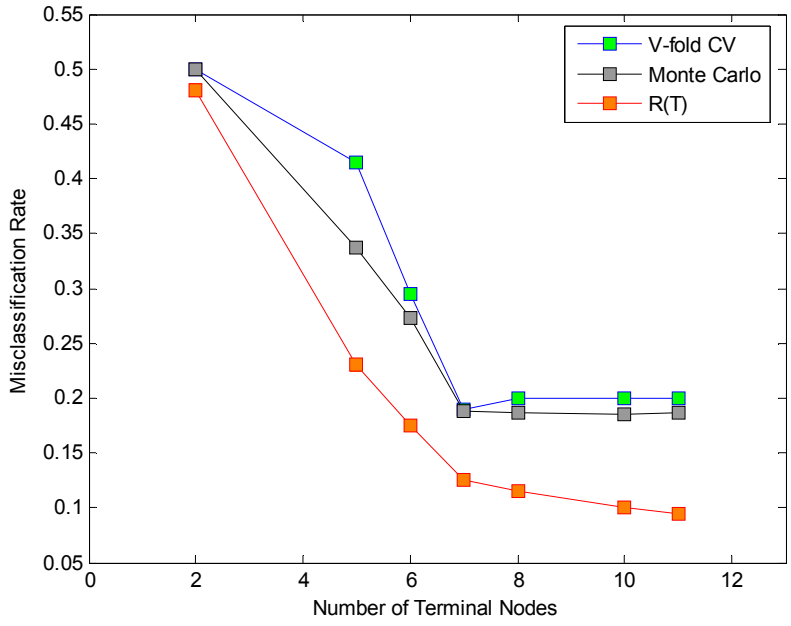

Figure 3-8. Comparison between the true misclassification rate $R^*(T)$, the V-fold cross validation estimate $R^{CV}(T)$, and the resubstitution estimate $R(T)$.

Table 3-2. Comparison between $R^*(T)$, $R^{CV}(T)$ and $R(T)$.

Terminal Nodes	Monte Carlo $R^*(T)$ Mean	Std	V-fold Cross Validation $R^{CV}(T)$	$R(T)$
2	0.5001	(0.018)	0.5	0.48
5	0.3372	(0.043)	0.415	0.23
6	0.2733	(0.036)	0.295	0.175
7	0.1885	(0.038)	0.19	0.125
8	0.1867	(0.039)	0.2	0.115
10	0.1859	(0.038)	0.2	0.1
11	0.1864	(0.039)	0.2	0.095

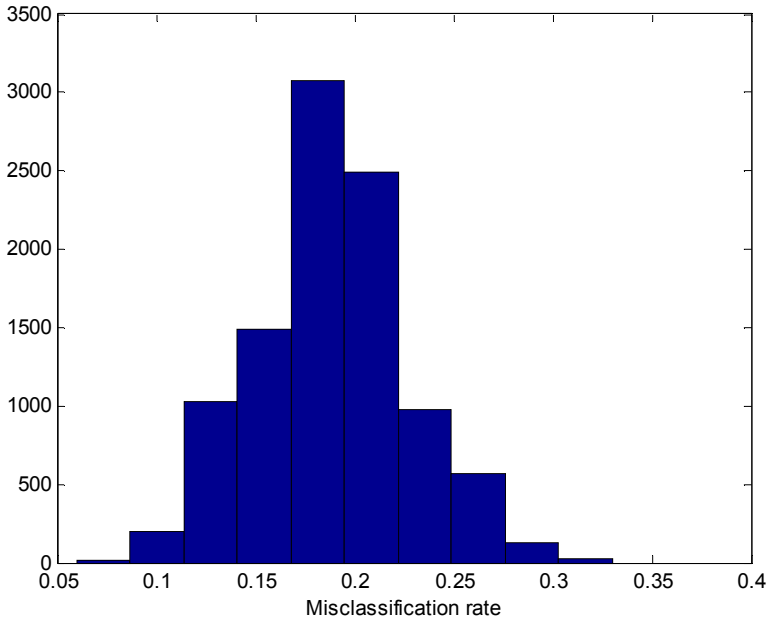

Figure 3-9. Monte Carlo misclassification rate of optimal subtree T_5.

Chapter 4 Methodology

Traditionally, protection systems have been biased towards dependability. System topology and good stability margins justified such design. However, it was argued in Chapter 1 that due to the manner in which power systems have evolved, this philosophy needs to be reviewed and that, under stressed system conditions, a favorable bias towards security is beneficial.

In this chapter the methodology to implement an adaptive security-dependability protection system is presented. First, it is recognized that there are some critical locations in the power grid where adaptive relaying is most beneficial. A systematic procedure to identify and rank the critical locations of a power system is presented in section 4.1.

The adaptive philosophy of protection systems acknowledges that relays may change their characteristics in order to tailor their operation to the prevailing system conditions. The methodology proposed in this chapter aims to reduce the likelihood of hidden failures and potential cascading events by adjusting the security/dependability balance of protection systems. The design of the security/dependability adaptive voting scheme is presented in section 4.2. The methodology is based on Wide Area Measurements (WAMs) and Data Mining. The advocated algorithm to grow Decision Trees is known as CART and it is thoroughly described in chapter 3.

4.1 Methodology to Identify Critical Locations

A hidden failure was defined as a permanent defect on a relay system that will cause the incorrect removal of a circuit element as a direct consequence of another event [13]. An analysis of NERC outages reports indicates that hidden failures are involved in over 70% of cascading outages. The threat that hidden failures pose is due to the intrinsic high risk associated with them. Typically, hidden failures are prone to manifest themselves under stressed system conditions [14] and therefore their consequence tends to be rather noteworthy.

Significant research effort has been employed in developing technology to detect hidden failures and prevent them from causing unwanted operations. However, hidden failures in relays are low probability events so it is difficult to economically justify deploying systems to protect every relay in the system from hidden failures. Attention and resources must be concentrated on areas in which the severity of an unwanted disconnection due to a hidden failure is relatively high. These areas are defined as the critical locations of the power system.

The purpose of this section is to develop a systematic procedure to identify and rank the critical locations of a power system.

4.1.2 Overview of Critical Locations.

In our context, critical locations are those locations in the power grid where a false trip caused by a protection hidden failure will be most detrimental for the system. In order to rank critical locations, an index of severity is developed to assess the consequence of hidden failures in protective equipment.

An index of severity to identify critical location based on dynamic simulations was proposed in [59]. Such index was defined as a function of the region of vulnerability, the amount of load lost due to load shedding, and generation lost due to generation rejection schemes or relay action. The list of study cases to be considered was selected using human expertise. Stability, in particular angle stability, was assessed by visual inspection of rotor angle plots.

For small systems with few test cases the approach taken in [59] is feasible and competent to identify critical locations. However, large power systems present a dimensionality curse. The study of hidden failures is, to some extent, analogous to a *N-2* contingency analysis. The main difference is that in *N-2* studies the occurrence of each contingency is considered to be caused by independent events, whereas for hidden failures the two events clearly exhibit dependence; only faults within the region of vulnerability can cause the manifestation of hidden failures.

Typically, *N-2* analysis performed by utilities only involve a reduced set of "credible" contingencies (see NERC Transmission Planning Standard [60]). The number of simulations required for an exhaustive *N-2* study can be computed by,

$$Simulations = \frac{N!}{(N-k)! \cdot k!} \tag{4.1.1}$$

where N is the total number of circuit elements in the system and k is the number of elements being removed, in this case, $k = 2$.

Figure 4-1 shows a plot of the number of simulations needed as a function of the number of circuit elements in the system. The plot exhibits an exponential trend and, as it can be seen, taking into consideration a thousand elements in the system gives rise to half a million simulations. The California model used in this dissertation has more 4000 buses, 4000 transmission lines, 1500 transformers, and 1100 generators; needless to say, this translates into quite a few simulations.

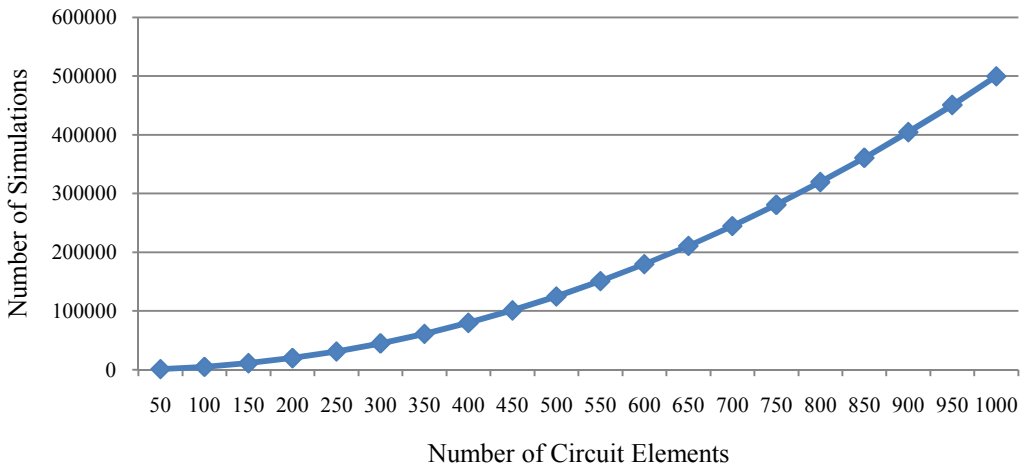

Figure 4-1. Required number of simulations as a function of the number of circuit elements.

To deal with this dimensionality curse a two step systematic procedure is proposed. A schematic of the advocated method is shown in Figure 4-2. The identification and ranking of critical locations is based on two indices: the Static Index and the Dynamic Index. The Static Index is founded on load flow analysis and it is intended as a fast contingency screening algorithm. It classifies cases into two sets: non-severe cases and harmful ones. The purpose of the Dynamic Index is to rank critical locations based on a severity assessment of the disturbance. The dynamic index is only computed on those cases determined to be harmful by the Static Index.

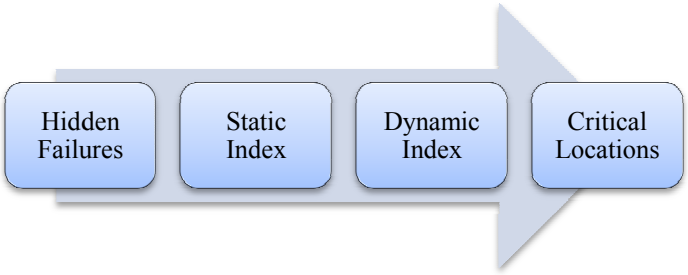

Figure 4-2. Methodology to identify the critical locations of the power system.

A systematic procedure is useful to confirm engineering judgment and intuition regarding critical locations, and to uncover others that are not obvious at first sight. In this dissertation, the main focus is hidden failures on relays protecting high voltage transmission lines. However, the proposed methodology can be used to identify critical locations on any protection relay.

4.1.3 Static Index

The main objective of the Static Index is to study a comprehensive list cases using load flow analysis[18]. For our purposes, a case is defined as a double contingency, a real fault followed by the removal of two transmission lines; the faulted element and the line whose protective relay has a hidden failure. A list of cases to be studied is created using an exhaustive procedure. Due to the dependence exhibited by hidden failure the number of simulations needed is a function of the system topology, operating condition, and the region of vulnerability of the protective equipment.

$$Simulations = f\left(\text{Topology}, \text{Protection System, Operating Conditions}\right) \qquad (4.1.2)$$

To better illustrate the vast number of cases to be analyzed, consider the one line diagram shown in Figure 4-3. Let us assume that there is a hidden failure in the relay protecting line number 1 connected between bus A and bus B. The region of vulnerability of such relay is denoted by dashed rectangles. For this particular example, it is assumed that the region of vulnerability extends only onto adjacent lines. However, it is possible, depending on the characteristics of the protective equipment, the system topology, and prevailing operating conditions, for the region of vulnerability to extend beyond adjacent buses[19].

Any fault lying within the region of vulnerability will cause the manifestation of the hidden failure and the unwanted disconnection of line number 1. An exhaustive list of all possible cases is created by assuming the occurrence of a fault inside the region of vulnerability; the list is shown in Table 4-1. It should be easy to appreciate the significantly vast number of combinations to be studied in a large power system.

[18] It is well known that the computational burden of load flow analysis is considerably less than that of dynamic simulations.

[19] A third zone impedance relay is an example of a protective relay whose region of vulnerability extends beyond adjacent buses (see Chapter 1, section 1.6).

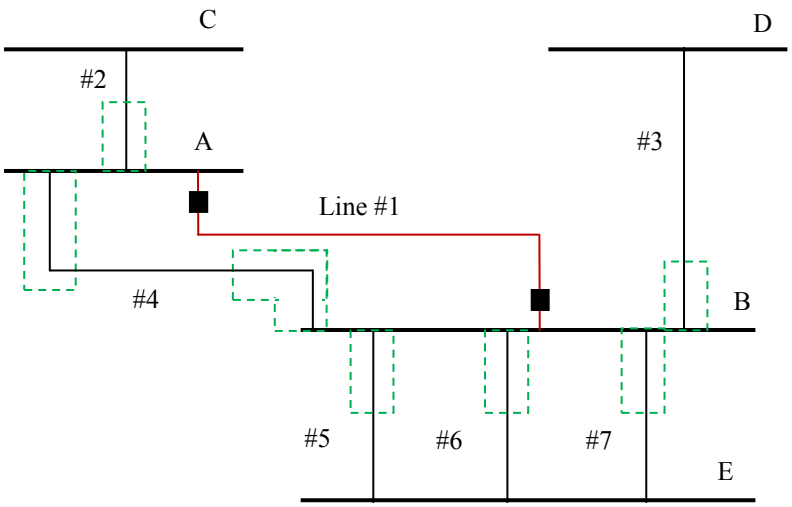

Figure 4-3. One line diagram. Line #1 has a hidden failure and its region of vulnerability is indicated by dashed rectangles. Any fault within the region of vulnerability will expose the hidden failure.

Table 4-1. Exhaustive list of cases for a hidden failure in line #1.

Case	Type	Line
1	Fault Hidden Failure	Line #2 Line #1
2	Fault Hidden Failure	Line #3 Line #1
3	Fault Hidden Failure	Line #4 Line #1
4	Fault Hidden Failure	Line #5 Line #1
5	Fault Hidden Failure	Line #6 Line #1
6	Fault Hidden Failure	Line #7 Line #1

The flow diagram of the Static Index algorithm is shown in Figure 4-4. An exhaustive list of cases is created by contemplating the possibility of having a hidden failure on each protective relay. As stated previously, for each case two transmission lines are removed from the system; the faulted line and the line whose protective relay has a hidden failure. The new operating point is then determined by solving a load flow. The idea is to compare the pre-fault operating conditions and the post-fault operating conditions. If the impact of a particular combination of

fault and hidden failure has a negligible consequence on the system, the contingency is said to be "non-severe". Under such circumstances, the binary output of the static index is set to zero and no further actions are taken concerning that particular case. On the other hand, if the disturbance has significantly deteriorated the system state, it is labeled as "harmful" and the binary output of the static index is set to one. A condensed list of potentially harmful cases is assembled with all cases classified as ones. The dynamic index is then used to rank such list.

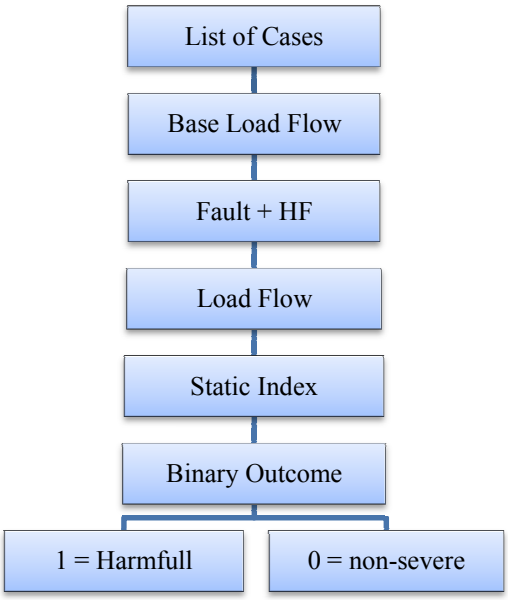

Figure 4-4. Static Index flow diagram.

A criterion needs to be defined in order to make the distinction between "non-severe" and "harmful" contingencies. The assessment is based on line loading conditions and bus voltage range.

In general, three factors determine line loading capabilities:

- Thermal limit.
- Voltage drop limit.
- Steady-state stability limit.

There is a well known relationship between line length and the loadability limiting factor. Such characteristic was first empirically determined by St. Claire [61]; reason why the curves are known as St. Claire's curves. The beauty of St. Claire's curves is that they are virtually

independent of voltage level, that is, they can be applied to any transmission line regardless of voltage level. The understanding of line loadability was later enhanced by analytical studies [62, 63].

The heat produced due to I^2R losses expands conductors and causes transmission lines to sag. If an overload is sustained for a certain period of time, the sagging line will eventually reach the minimum clearance to ground causing a line-to-ground fault. The available window of time under such overloaded condition depends on the pre-contingency current, ambient temperature, wind velocity, and the total clearance to ground. Thermal limits are usually the limiting factor for short lines; lines lengths of 50 miles or less.

To maintain an adequate quality of power at the delivery point, a maximum voltage drop across the line of 5% is typically allowed. Voltage drop is the limiting factor for medium length lines; lines lengths between 50 and 200 miles. In long lines, above 200 miles, their thermal limits tend to exceed the network requirements of power flow through the line. In general, the thermal limit is actually set by the weakest link on the line; line terminating equipment, breakers, substations, wave traps, etc. The limiting factor for long lines is the so called steady-state stability limit.

There is a maximum power that can be transferred through a transmission line. The relationship is known as the power angle curve[20] and it is a function of the angle across the transmission line,

$$P = \frac{E_s \cdot E_r}{X} \sin(\delta) \tag{4.1.3}$$

The maximum power transfer is achieved when $\delta = 90^\circ$. The steady-state stability limit is defined in terms of a desired stability margin,

$$\text{Stability Margin \%} = \frac{P_{\max} - P_{\text{limit}}}{P_{\max}} \cdot 100 \tag{4.1.4}$$

[20] The power angle curve was presented in Chapter 1; see equation (2.1.1). Initially, as the angle across the line δ increases, the increase in current I dominates over the decrease in the midpoint voltage V_m and an increase in power is observed. Beyond $\delta = 90$, the decrease in V_m dominates over the increase in I, and the power transmitted is therefore reduced. Under such condition the system becomes unstable.

then, with a stability margin of 35% the maximum angle allowed across the line is 40 degrees.

Utilities perform detailed studies to determine the maximum line loadability according to the limiting factors described above. Typically, maximum loading thresholds are included in power system models.

The power system must also be able to maintain acceptable voltages at all buses. A voltage collapse occurs when voltage drops in a progressive and uncontrollable manner. Typically, bus voltages should remain between 0.93 and 1.05 pu. The upper voltage limit depends on the rated voltage of the line; in general, most 500 kV lines are operated around 1.05 pu. The lower voltage limit is of particular interest if under voltage load shedding (UVLS) protection schemes are implemented.

To conclude, the criterion to distinguish between "non-severe" and "harmful" contingencies is based on line loading conditions and bus voltages. The thresholds used are reported in Table 4-2. It should be noted that these parameters were specified to achieved an appropriate valance in the screening process; the list of cases classified as class one should be as condensed as possible, but no harmful case should be labeled as non-severe.

Table 4-2. Static Index parameters' thresholds.

Parameter	Limit
Line Loadability	110%
Bus voltages[21]	0.93 to 1.05
Maximum voltage drop across a line	0.05
Maximum bus voltage change	0.07 pu
Convergence	Yes/No

Contingency Ranking with Load Flow Analysis

Due to the manner in which the static index was formulated (binary outcome) it is not possible to rank cases in order of severity. However, several indices have been proposed in the

[21] Special limits were used in buses with under voltage load shedding and in 500 kV buses. Buses with limit violations in the base case where not considered in the analysis.

literature to rank contingencies using load flow analysis. An overload index proposed in [64] is defined as,

$$i(P) = \sum_{i}^{N} \left(\frac{P_i}{P_{MAX_i}} \right)^2 \tag{4.1.5}$$

where N is the total number of transmission lines in the system, P_i is the real power flowing through transmission line i, and P_{MAXi} is the maximum rated real power flow through transmission line i. The index is known to have potential "masking" effects; the index may give a higher rank to a case with multiple lines loaded close to their limits than a case with a single overloaded line.

Similarly, an index for voltage performance can be implemented as,

$$i(Q) = \sum_{i}^{N} \left(X_i \cdot P_i^2 \right) \tag{4.1.6}$$

where X_i is the reactance of branch transmission line i, and P_i is the real power flowing through branch i.

4.1.4 Dynamic Index

An exhaustive list of disturbances plus hidden failures is screened using the Static Index. A condensed list is assembled with all cases classified as ones. Such list is further scrutinized with the aid of dynamic simulations. In this section a severity index, called the Dynamic Index, is defined.

Figure 4-5 shows the flow diagram of the Dynamic Index. As stated previously, only cases classified as class one by the static index are contemplated. For each case, a dynamic simulation is run for ten seconds. A shorter simulation time can greatly reduce the computational burden. In general, ten second simulations are customary for dynamic studies since it is long enough to account for second swing instability. Protection relays, such as under voltage load shedding and out-of-step relays, are built in the model. It is therefore possible for the original contingency and hidden failure to result in a cascading sequence of events. The severity of each

case is assessed using a rotor angle coherency-based index: the Integral Square Generator Angle (ISGA).

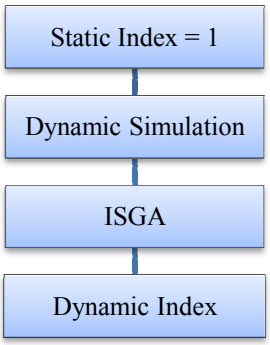

Figure 4-5. Flow diagram of the Dynamic Index.

The ISGA index was first proposed in [26] as an index to predict angle instability. It is defined as the weighted sum of the difference between generator rotor angles and the center of angle,

$$ISGA = \frac{1}{T \cdot S_T} \int_0^T \sum_i^N S_i \cdot \left(\delta_i(t) - \delta_{COA}(t) \right)^2 \tag{4.1.7}$$

where S_i is the machine rated power, δ_i is the generator rotor angle, S_T is the sum of S_i over all machines, T is the total simulation time, and is δ_{COA} the center of angle.

$$\delta_{COA}(t) = \frac{\sum_i^N S_i \cdot \delta_i(t)}{\sum_i^N S_i} = \frac{\sum_i^N S_i \cdot \delta_i(t)}{S_T} \tag{4.1.8}$$

The center of angle (COA) is the electrical analogous of the center of mass. In mechanics, the center of mass of a body is a function of the position and mass of the particles that compose the system and it is frequently used to describe the system's response to external forces. It turns out that, just like the center of mass, the electrical center of angle is a convenient reference for generators' angles. In power systems, the center of angle is a function of the rotor angles of the machines ("position") and the rotational moment of inertia of the machines ("mass"). In equation (4.1.8) the machines' rated VA base S_i is used instead of the moment of

inertia J_i. In general, J_i is not a parameter readily available in power system models. It can be easily computed by,

$$J = \frac{2 \cdot H \cdot S}{\omega_s^2} \tag{4.1.9}$$

where H is the machine's inertia constant, S is the machine's rated VA base, and ω_s^2 is the rated angular velocity in mechanical radians per second. However, for simplicity and without any loss of generality, S is used instead of J to compute the center of angle.

The ISGA[22] is a coherency-based index and it is tailored made to rank critical locations. In essence, the ISGA score measures the electro-mechanical oscillations incurred by generators in the system due to the applied disturbance and hidden failure. Stable cases have a relatively small ISGA score while unstable events have the largest scores. The score facilitates the distinction between stable and unstable cases at a glance. The main attractive characteristics of the index that make it suited to rank critical locations are:

- The index is proportional to the size of the machine loosing synchronism. The deviation of machine i from the center of angle $(\delta_i(t) - \delta_{COA}(t))^2$ is weighted by the size of the machine, S_i. Therefore, large machines going out of step are greatly penalized by the index. If two cases, case A and case B, loose a single machine, but the rated power of the machine going out of step is in case A is larger than the one in case B, $S_A > S_B$, then by construction $ISGA_A > ISGA_B$.
- Since deviations from the center of angle $(\delta_i(t) - \delta_{COA}(t))^2$ are squared, small departures from the COA are down weighted while large deviations are magnified.
- The index is proportional to the total number of generators that loose synchronism. The algorithm sums over all generators, therefore having more generators going out-of-step implies a larger score.

The top ISGA scores identify the critical locations of the power system. Results show that the ISGA is very effective in distinguishing stable cases from unstable cases and then

[22] It should be noted that the ISGA index is not a form of the kinetic energy of the network 26. Rovnyak, S., et al., *Decision trees for real-time transient stability prediction.* Power Systems, IEEE Transactions on, 1994. **9**(3): p. 1417-1426..

ranking the unstable cases. An extension to the dynamic index is proposed in [65] based on the transient energy function (TEF). The main objective is to create a sub-ranking of stable cases since the transient energy function proved to be adept at ranking the stable cases.

4.2 Methodology for a Security/Dependability Adaptive Scheme

Reliability in the context of power system protection comprehends two aspects, dependability and security. Dependability was defined in chapter 1 as *"the degree of certainty that a relay or relay system will operate correctly"*. Security *"relates to the degree of certainty that a relay or relay system will not operate incorrectly"*. In general, enhancing security implies an intrinsic loss of dependability and vice versa. Protection engineers try to achieve an optimal balance between these two conflicting concepts; this is why power systems protection is often recognized as an art.

Traditionally, protection systems have been biased towards dependability. System topology and good stability margins justified such design. An adequate transmission line redundancy entails a variety of alternative paths for power to flow. Power systems that exhibit sufficient transmission line redundancy can withstand losing a line due to lack of security without jeopardizing the systems operation; provided that lines have enough loading margins. Under this scenario, the consequence of not tripping when a fault occurs (lack of dependability) is far worse than tripping when it is not necessary (lack of security).

It was argued in chapter 1 that due to the manner in which power systems have evolved, this philosophy needs to be reviewed and that, under stressed system conditions, a favorable bias towards security can be beneficial. The adaptive philosophy of protection systems acknowledges that relays may change their characteristics in order to tailor their operation to the prevailing system conditions. The methodology proposed in this chapter aims to reduce the likelihood of hidden failures and potential cascading events by adjusting the security/dependability balance of protection systems. When the power system is in a "safe" state, a bias towards dependability is desired. Under such conditions, not clearing a fault with primary protection has a greater impact on the system than a relay miss-operation due to lack of security. However, when the power system is in a "stressed" state, unnecessary line trips can greatly exacerbate the severity of the

outage, contribute to the geographical propagation of the disturbance, and may even lead to cascading events and subsequent blackout. Under such states, it would be desirable to alter the reliability balance in favor of security.

A conceptual overview of the security/dependability adaptive voting scheme was given in chapter 1 and the schematic shown in Figure 1-8 is repeated here for convenience. Critical locations were defined as those locations in the power grid where a false trip caused by a protection hidden failure is most detrimental for the system. The optimal location for the security-dependability adaptive scheme can be determined using the systematic procedure presented in section 4.1.

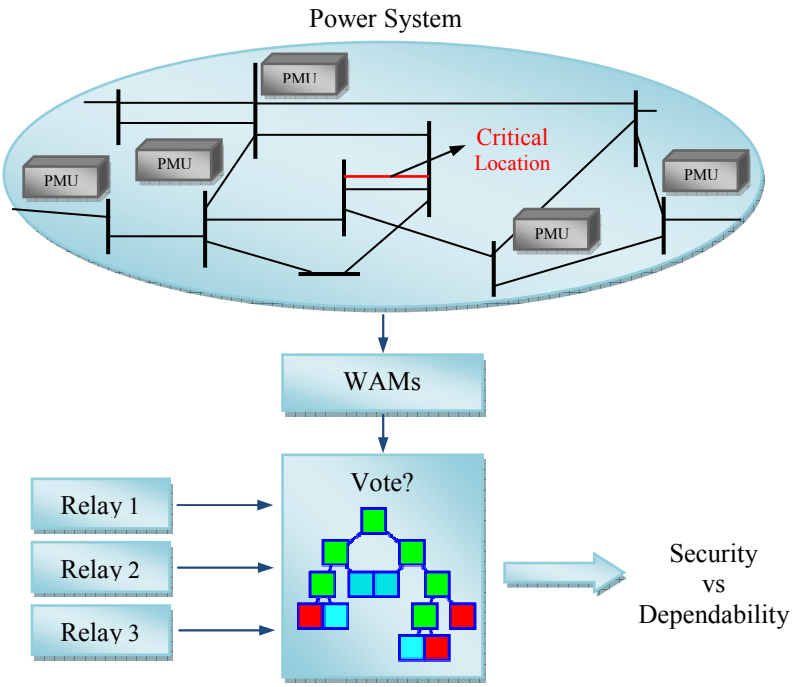

Figure 4-6. Conceptual schematic: adaptive security/dependability voting scheme.

The voting scheme consists of a set of three independent and redundant relays. Wide area measurements are obtained with the aid of PMUs. The underlying hypothesis is that phasor measurements at strategic buses provide enough information to discriminate the need for a bias towards security. These measurements are used to infer the state of the power system which is then classified as either "stressed" or "safe". If the system is found to be stressed, the proper course of action is to enable the voting scheme and therefore bias the protection system towards

security. On the other hand, if the system is found to be safe, the voting scheme is disabled and only one relay takes on the protective function, i.e., a favorable biased towards dependability.

The methodology to implement the adaptive voting scheme is concerned with the following:

- Where should PMUs be placed in order to infer the system state?
- How should a "stressed" or "safe" state be defined?
- What attributes are more adept for the purpose of classifying the system state?

The advocated methodology is based on Data Mining. A thorough description of Decision Trees can be found in chapter 3. Most of the effort is concerned with developing the learning sample *L*. Then, a classification tree is grown using CART's algorithm. The final Decision Tree provides the decision function to discriminate between "safe" and "stressed" states, i.e., to recognize the need for a dependability bias (disarm the voting scheme) or a bias towards security (arm the voting scheme). The splitting nodes of the DT pinpoint the desired location of PMU measurements.

4.2.1 Developing the Learning Sample *L*.

The flow diagram of the proposed procedure to build the learning sample *L* is shown in Figure 4-7. The main objective of the adaptive scheme is to alter its security/dependability balance to better suit prevailing system conditions. It is therefore necessary to generate a representative sample of different operating points[23].

Diverse operating points are generated through a combination of load scaling and load flow solutions[24]. Consider the four major control areas in California: Pacific Gas and Electric (PG&E), Southern California Edison (SCE), San Diego Gas and Electric (SDG&E), and Los Angeles Department of Water and Power (LADW&P). The idea is to systematically scale the system load by subsequently increasing and decreasing the load at each area. A combinatory of load scaling using two areas at the same time is also performed; for example, the load at PG&E

[23] In general, Data Mining methods require a large sample size for optimal results.
[24] Load flow solutions represent snapshots of the power system state.

may be increased by 3% while the load at SCE is decreased by 10%. It is argued that the proposed load scaling process induces enough variation in voltage phasors across the 500 kV network to mine patterns in the data. It is likely that the sample generated will include typical system states and some unrealistic ones. However, "unrealistic" conditions also provide valuable information since it is precisely those atypical and unexpected conditions the ones that tend to jeopardize the system. If available, historical information of daily load curves can further enhance the learning sample. Ultimately, the main objective is to induce as much variation in the operating points as possible.

Wide area measurements at all 500 kV buses are obtained with the aid of PMUs. A set of complex phasor voltages at every bus fully specifies the system state (assuming that the topology of the system is perfectly known). It is argued that key measurements at strategic 500 kV buses are enough to predict the appropriate security/dependability balance of the adaptive protection scheme. Typically, several attributes are included as potential predictors: voltage magnitudes, angle differences[25], MVar flows, current magnitudes, etc. Voltage magnitudes throughout the system tend to sit around 1 pu and therefore obscure the mining process; in general, the square of voltage magnitudes is a better predictor since bus voltages close to the normal value of 1 pu remain unchanged while deviation from it are magnified. Results show that an outstanding predicting attribute can be obtained by decomposing the current flowing through a transmission line into its real and imaginary parts. Current flows typically exhibit large deviations and are therefore good predictor candidates.

[25] As the angle difference between generators increases, the synchronizing power coefficient decreases. Therefore, large angles across the network are indicative of potential loss of synchronism if the system is subjected to a severe disturbance.

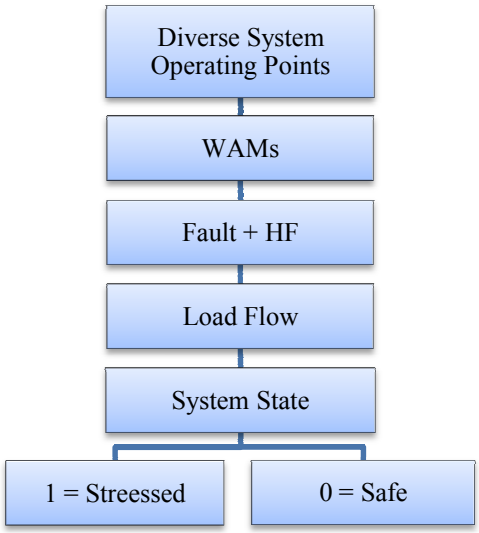

Figure 4-7. Flow diagram: developing the Learning Sample *L*.

For each operating point, a fault and a hidden failure are assumed to occur at the pre-determined critical location. Since Decision Trees constitute what is known as a supervised learning method, an objective function is defined to classify the system state prior the contingencies into two classes: "safe" or "stressed". The goal is to determine whether the manifestation of a hidden failure has a severe impact on the system or not. If the system prevailing conditions are such that it can withstand the materialization of a hidden failure at the critical location, then a favorable bias towards dependability is desired since the consequence of not tripping a fault due to lack dependability is far worse than over-tripping due to lack of security. On the other hand, if the system is stressed and the unnecessary removal of a critical line can potentially jeopardize the power system, then a bias towards security is preferred. It should be emphasized that the classification of the system state into "safe" and "stressed" is done with respect to the selected critical location, i.e., it is not a general statement regarding the system state and it may or may not apply to other disturbances in the system. For example, it is possible for a "safe" operating point (regarding the point of view of the adaptive scheme) to be in fact extremely hazardous for other critical locations.

The exact form of the objective function is a matter of engineering judgment and empirical evidence. The modeler may use a criterion similar to that of the static index presented in section 4.1. For the purpose of this dissertation, extreme contingencies were contemplated. A

natural criterion, due to the inherent high severity of the contingency, is to check the convergence of the load flow problem. Cases in which the load flow fails to converge are classified as "stressed". If a solution is achieved, then the system state is labeled as "safe". Note that the classification refers to the system prevailing conditions prior the contingencies and it is determined by evaluating the operating point, or lack of it, after the events.

The relationship between load flow divergence and system stability is well known. For example, *V-P* and *Q-V* curves are frequently used to study voltage stability. Such curves are plotted through a sequence of power flow solutions for different load levels; the load is increased until the load flow fails to converge which is indicative of instability.

A matrix structure of L is shown in Table 4-3. Each measurement vector x_i represents a system state. Based on engineering judgment and empirical evidence several attributes (columns of the matrix) are included.

Table 4-3. Learning sample *L*. Attributes: bus voltage angles, real and imaginary currents, and voltage square magnitudes.

	Class	θ_{GATES}	θ_{DIABLO}	θ_{MIDWAY}	…	Ir_{1106}	Ii_{1106}	…	Ii_{3850}
x_1	1	-3.91	2.57	-5.89	…	5.40	1.78	…	-0.09
x_2	0	-2.52	4.14	-3.99	…	3.97	1.83	…	0.16
x_3	1	-3.95	2.52	-5.89	…	5.14	1.69	…	-0.13
x_4	1	-3.68	2.84	-5.52	…	4.92	1.72	…	-0.08
...	…	…	…	…	…	…	…	…	…
x_{4150}	0	-3.00	3.61	-4.62	…	4.36	1.77	…	-0.06

Summary of the algorithm to develop *L*.

To conclude, the algorithm to develop the learning sample can be summarized by the following steps:

- Diverse system operating conditions are obtained through a systematic load scaling process.
- At each system state, several measurements are taken at all 500 kV buses in the system. Proposed attributes: voltage phasor angles, real and imaginary currents, and square voltage magnitudes.

- For each operating point a fault and a hidden failure are assumed to occur. A load flow solution is attempted. If it converges the system state prior to the contingencies is said to be "safe" and it is classified as a zero; otherwise, it is said to be "stressed" and it is classified as a one.

4.2.2 Training the Decision Tree.

The algorithm used to grow Decision Trees is known as CART and it is thoroughly described in Chapter 3. The advocated Data Mining method tackles two problems at the same time. First, and foremost, it provides an intuitive and simple model to predict the appropriate reliability balance of the adaptive protective scheme based on wide area measurements. Second, splitting attributes determine the locations where PMU are needed. Several PMU placement algorithms have been proposed in the literature [12, 66, 67]. In general, an observability function[26] is used to guide the PMU placement process [68]. The methodology proposed in this dissertation is itself an application oriented PMU placement algorithm.

A schematic of a Decision Tree is shown in Figure 4-8. As stated previously, splitting nodes determine the PMU placement; splitting nodes test if attribute a_i is less than or equal to an optimal splitting threshold s (see chapter 3). A classification decision is made at terminal nodes. If the system is classified as "stressed" a bias towards security is preferred, and the voting scheme is armed. On the other hand, if the system is classified as "safe", a bias towards dependability is desired and only one relay performs the protective action (the voting scheme is disarmed).

[26] The concept of "depth of observability" is proposed in [65]. Assuming that topology and network data is known, a single PMU can provide information regarding the local bus at which the PMU is placed and all other adjacent buses. All buses adjacent to the PMU are defined as depth one; the next bus away is said to be at depth two. PMU placement algorithm attempt to minimize the number of PMUs needed for a specified depth of observability.

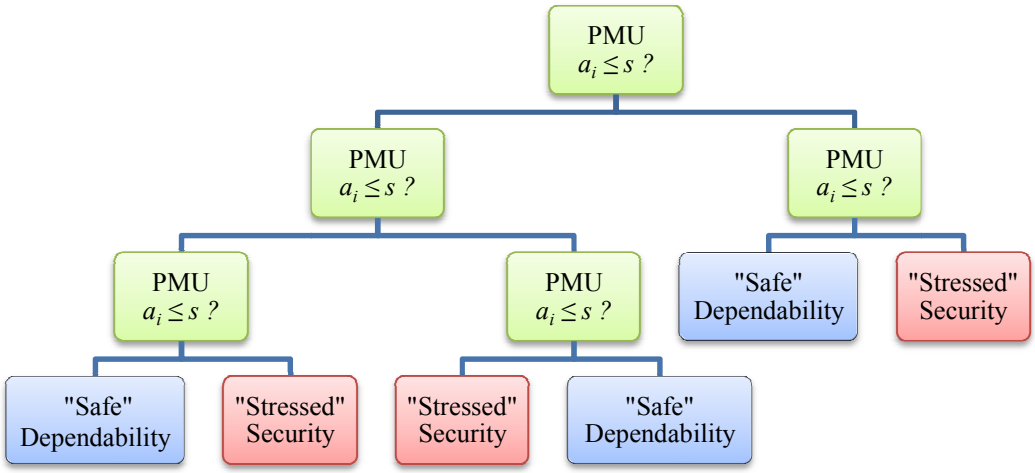

Figure 4-8. Schematic of a Decision Tree. Splitting nodes indicate PMU placement. A classification decision is made at terminal nodes. If the system is classified as "stressed", a security bias is preferred and the voting scheme is armed. If the system is classified as "safe", a bias towards dependability is desired and only one relay performs the protective action.

To illustrate the logic behind CART, consider the two three-dimensional contour plots of voltage angle[27] measurements shown in Figure 4-9; it is assumed that PMUs are placed at all 500 kV buses. The learning sample consists of more than 4000 of such planes; similar three-dimensional contour plots can be made with other attributes (currents, voltages, etc). The surface shaded in red represents an operating system condition classified as "stressed"; the blue shaded surface belongs to the "safe" class. The objective of the data mining algorithm is to increase class homogeneity, that is, it attempts to discriminate and isolate the two classes by subsequently partitioning the sample space using hyper-planes. In essence, optimal partitions are dictated by data regularity patterns.

[27] Note: voltage angles are all referenced to a specific bus, Pittsburg.

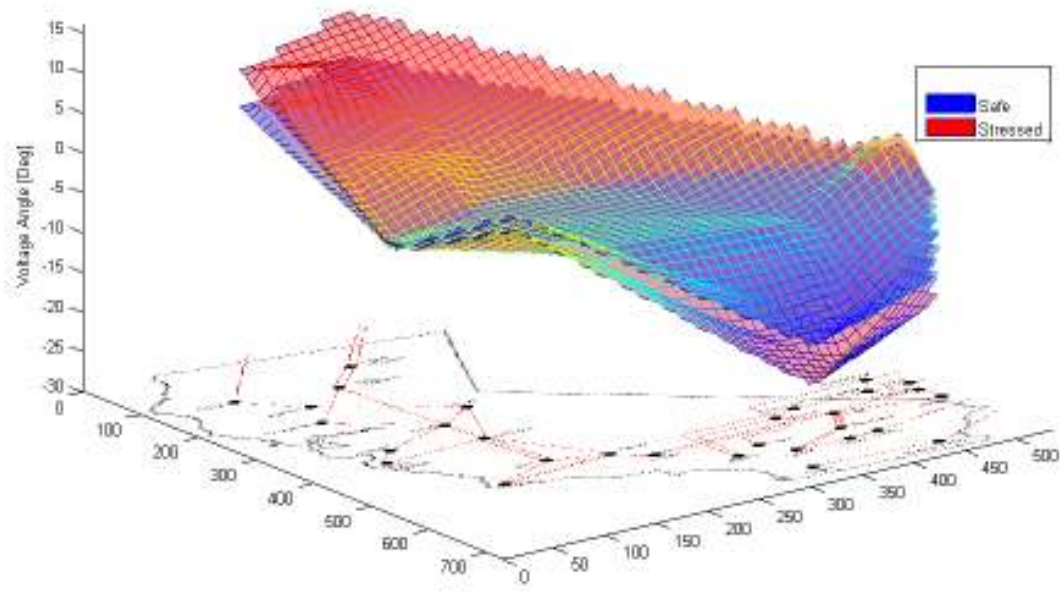

Figure 4-9. Three-dimensional contour of bus voltage angles in 500 kV buses in California.

Theoretically, and assuming a "ceteris paribus" clause, Decision Trees also suggest a course of action to modify the classification of the system state. Let us assume that a measurement vector x_i is dropped down the DT and the terminal node reached classifies the system as stressed. Let us also assume that the father of the terminal node (a splitting node) tests if the real current flowing through a particular line is less than some predefined threshold and that system conditions are such that the answer to the question is no. If the operator is able to reduce the real current flowing through the line without altering any of the other tested attributes ("ceteris paribus"), then the system can be brought back to a safe state. Needless to say, such ceteris paribus assumption is not likely to hold. However, the author suggest that future research efforts should be made to assess the potential that Decision Trees have to uncover a course of action by making an upside-down read of the path to a terminal node.

As a final remark, it is suggested that the modeler should first inspect scatter plots of the attributes included in the learning sample. CART's algorithm can craft new splitting decisions by using a linear combination of attributes. Consider the scatter plots shown in Figure 4-10. The stochastic data was generated using a bivarite Gaussian random process. On the scatter plot on the left, a linear combination of the attributes x_1 and x_2 is used to partition the sample space. On the scatter plot on the right, single attributes are used to partition the learning sample and, as a consequence, multiple splits are needed to grow the classification tree.

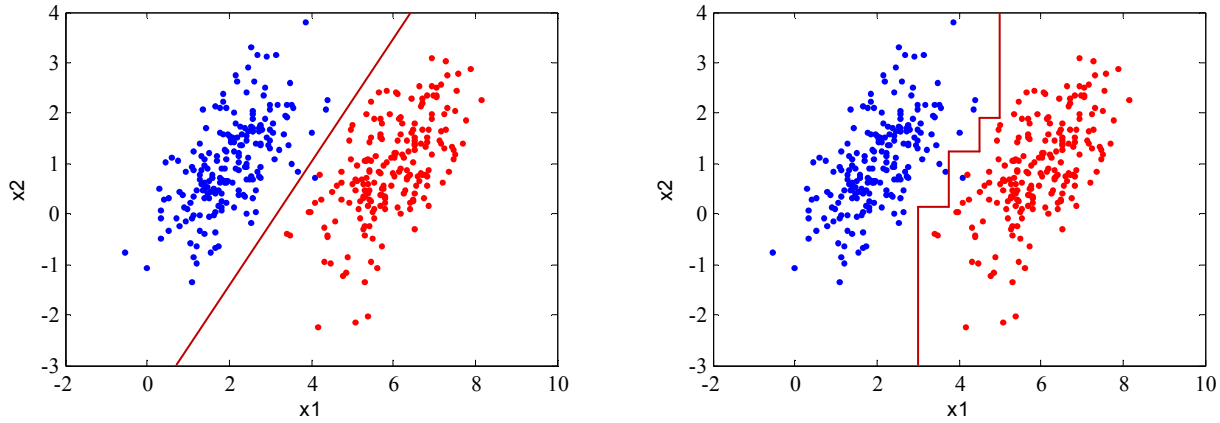

Figure 4-10. Bivariate Normal random variables. On the left, a linear combination of the attributes x1 and x2 is used to partition the sample space. On the right, single attributes are used to partition *L*, as a result, several splits are need to grow the classification tree.

Chapter 5 Simulation Results

Methodology to implement a security/dependability adaptive protection scheme was presented in Chapter 4. In this chapter, the methodology is put in practice using a highly detailed model of the California power system.

In section 5.1, critical locations in the power system are identified and ranked using the systematic procedure described in Chapter 4. The remainder of the chapter is concerned with growing decision trees to recognize the need to alter the reliability balance, security versus dependability, of the adaptive scheme. Two seasonal California models, heavy winter and heavy summer, are used to demonstrate the advocated methodology.

5.1 Critical Locations

Critical locations are defined as those locations in the power grid where a false trip caused by a protection hidden failure will be most detrimental for the system. The systematic procedure described in section 4.1 to identify and rank the critical locations of a power system is demonstrated using a highly detailed model of California. Hidden failures in relays protecting 500 kV lines were considered in the analysis; in particular, lines that constitute path 15 and path 26 of the California system as recommended by the advisory committee of the VT-CIEE research project.

An exhaustive list of study cases is created. In our context, a case is defined as a fault followed by two hidden failures. It is assumed that the region of vulnerability of any particular relay in the system extends only to adjacent lines; this renders a total of 501 cases.

The Static Index is used as a contingency screening tool. As stated in section 4.1.1, the static index is based on load flow analysis and its objective is to discriminate between non-severe and potentially harmful cases. If the impact of a particular combination of fault and hidden failure has a negligible consequence on the system, the contingency is said to be "non-severe". Under such circumstances, the binary output of the static index is set to zero. On the other hand, if the disturbance has significantly deteriorated the system state, it is labeled as "harmful" and the binary output of the static index is set to one. The parameters and thresholds used to assess the post-disturbance system state are shown in Table 5-1. A condensed list with 41 potentially harmful cases is assembled with all cases classified as ones.

Table 5-1. Static Index parameters' thresholds.

Parameter	Limit
Line Loadability	110%
Bus voltages	0.93 to 1.055
Maximum voltage drop across a line	0.05
Maximum bus voltage change	0.07 pu
Convergence	Yes/No

The condensed list of 41 cases is further scrutinized with the Dynamic Index. In order to assess the severity of each disturbance, an Integral Square Generator Angle (ISGA) index is proposed. As stated in section 4.1.2, the ISGA index is a coherency-based score constructed as a weighted sum of rotor angle deviations from the center of angle. The index allows the engineer to distinguish stable from unstable cases at a glance. The ISGA score is used to rank the study cases; the case with the largest score identifies the critical location of the power system. The complete list of ISGA scores is shown in Appendix B.

As an example, consider a partial list with four cases; the simulation results are shown in Table 5-2. The first case in Table 5-2, case number 350, has the largest score and it determines the optimal location to place the adaptive security/dependability protection scheme. Protection relays at anyone of the three 500 kV parallel lines connecting Midway-Vincent are the best candidates for an adaptive scheme. The ISGA score is comparatively large compared to other cases in the table. Figure 5-1 depicts a plot of generator's rotor angle excursions in area 24, Southern California Edison. The plot clearly indicates that the system splits apart with a group of coherent machines north from the path and another one in the south.

In the second case in the table, case 237, due to the applied disturbance more than 2000 MW of generation are removed from the system. Figure 5-2 shows two large generators drifting away from the system. The rest of the generators in the system remained coherent. The third and fourth cases in the table show non-severe, stable cases. The ISGA score for case number 269 is slightly larger than case 115. The difference between the plots shown in Figure 5-3 and Figure 5-4 is subtle, but after careful inspection, it can be seen that in case 269 the generators undergo larger and longer sustained oscillations, which renders a larger ISGA score.

To conclude, the Midway-Vincent path was determined to be the system critical location. A schematic of the backbone 500 kV transmission lines in California is shown in Figure 5-5. The figure highlights the optimal placement for the security/dependability adaptive scheme. The critical location suggested by the proposed procedure was confirmed, based on practical experience, by the advisory committee of the VT-CIEE research project. Human expertise and engineering judgment continue to be invaluable tools in the operation of the power system. However, the proposed systematic procedure can verify intuition and uncover other critical locations not obvious at first sight.

Table 5-2. ISGA score of four different cases.

CASE	FAULT	Bus From	Bus To	ISGA
350	F	MIDWAY	VINCENT	6721.188
350	HF	MIDWAY	VINCENT	
350	HF	MIDWAY	VINCENT	
237	F	GATES	DIABLO	4316.469
237	HF	DIABLO	MIDWAY	
237	HF	DIABLO	MIDWAY	
269	F	DIABLO	MIDWAY	9.7647
269	HF	MIDWAY	VINCENT	
269	HF	MIDWAY	VINCENT	
115	F	TABLE MT	VACA-DIX	7.7235
115	HF	ROUND MT	TABLE MT	
115	HF	TABLE MT	TESLA	

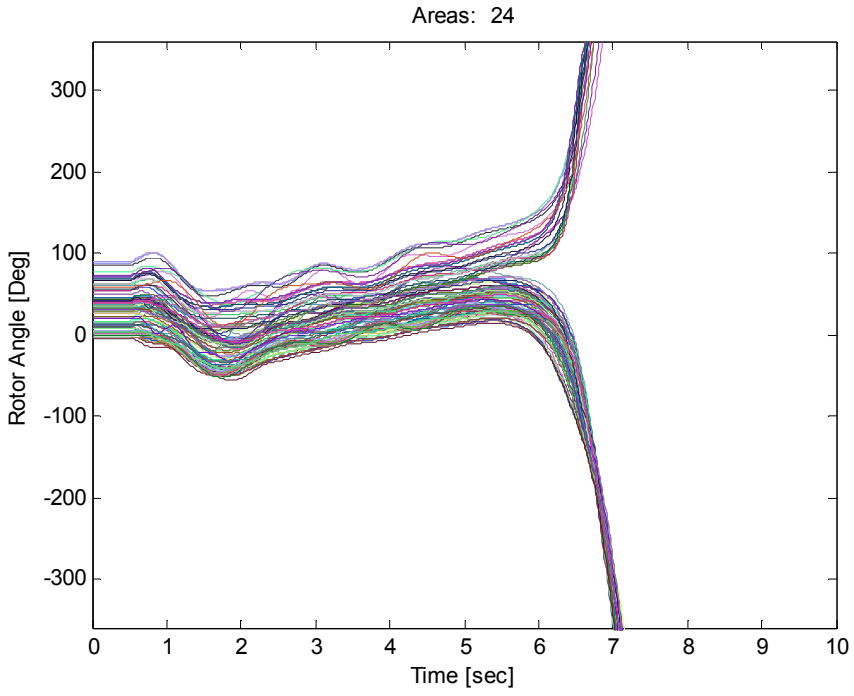

Figure 5-1. Generator rotor angles of study case number 350. ISGA score: 6721

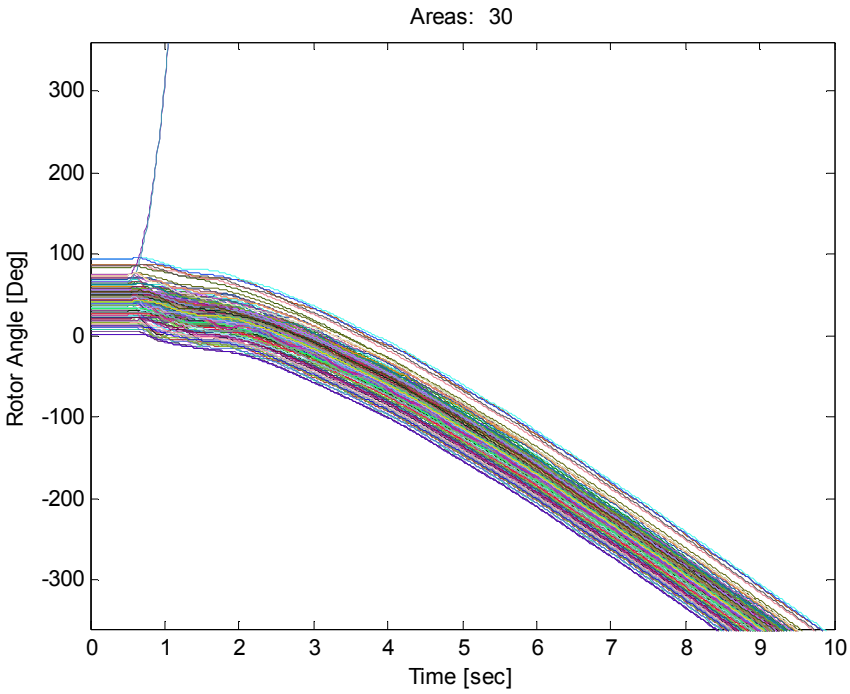

Figure 5-2. Generator rotor angles of study case number 237. ISGA score: 4316.

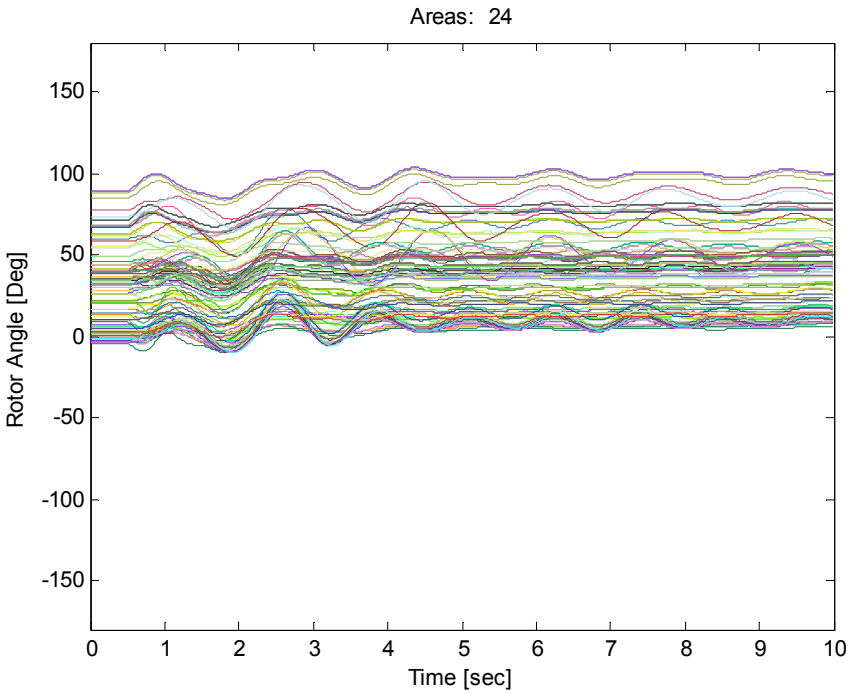

Figure 5-3 Generator rotor angles of study case number 269. ISGA score: 9.76.

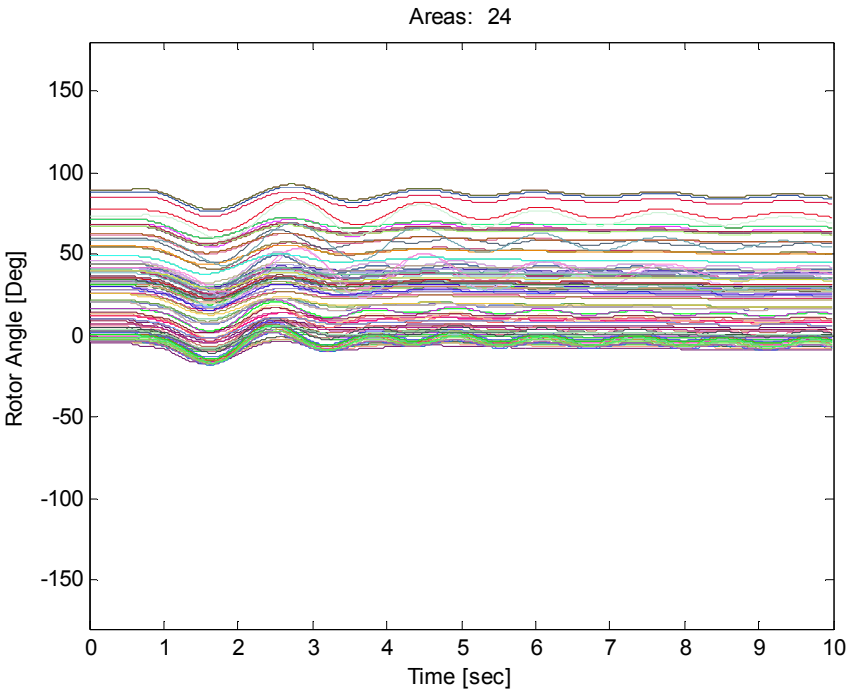

Figure 5-4. Generator rotor angles of study case number 115. ISGA score: 7.72.

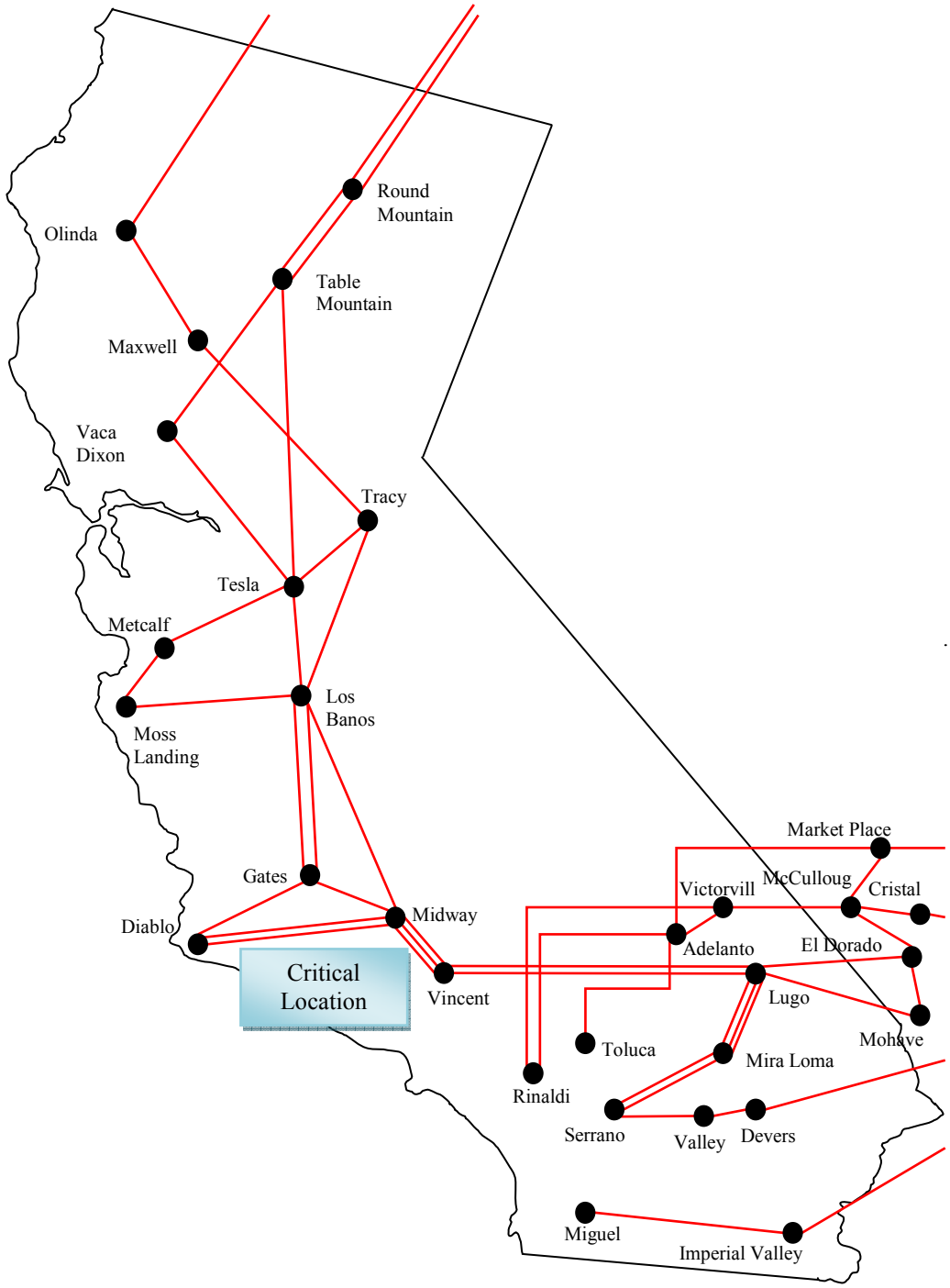

Figure 5-5. Schematic: 500 kV buses and lines in California. Midway-Vincent is determined to be the system critical location, i.e., the location where an adaptive security/dependability scheme is most beneficial.

5.2 Adaptive Security/Dependability Protection Scheme

In the advocated methodology, decision trees are trained off-line to be used as an on-line application. An accurate model of the power system is therefore crucial for an optimal performance of decision trees. The proposed methodology is tested using two seasonal models of the power system of California: heavy winter and heavy summer. The decision tree logic should be updated whenever significant changes are made to the system model.

In general, load composition[28], generation dispatched, amounts of imported power, peak load, topology, and scheduled maintenance, vary from season to season. Utilities develop various seasonal models to reflect such characteristics. Figure 5-6 shows a comparison of the dispatched generation between heavy winter and heavy summer models by area. The total power generated inside California almost doubles in heavy summer. Figure 5-7 shows a comparison of the real power consumption; again the difference between seasons is much accentuated.

In the following sections, decision trees are grown for each seasonal model. DTs define the logic to adjust the security/dependability balance of the adaptive protection scheme and also determine the required PMU placement.

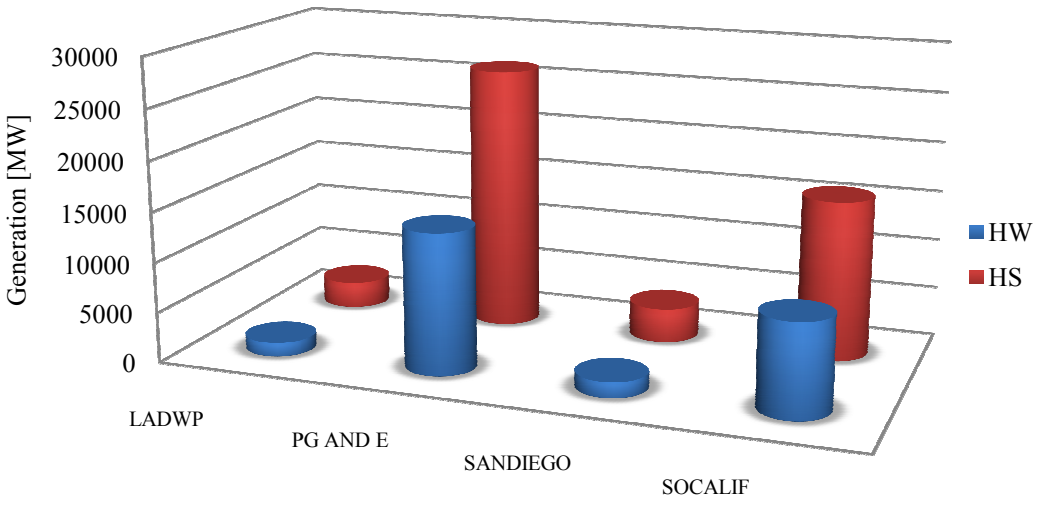

Figure 5-6. Comparison of generation dispatched between heavy winter and heavy summer model.

[28] Load composition is only adjusted in dynamic models. In load flow analysis, load is typically modeled as a sink of constant real and reactive power.

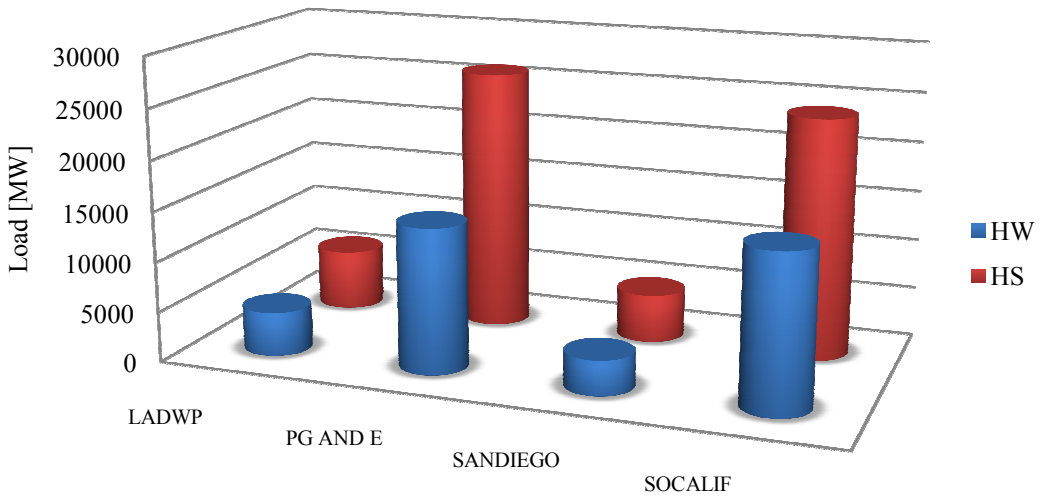

Figure 5-7. Comparison of load consumption between heavy winter and heavy summer model.

5.2.1 Decision Tree: Heavy Winter Model

Due to the features exhibited by the heavy winter model, a single hidden failure at the critical location, though economically costly, does not jeopardize the power system. However, results show that two hidden failures can potentially initiate a cascading sequence of events that lead to a system collapse. For simplicity, it is assumed that two out of the three parallel lines have defective protective relays, i.e., hidden failures. As a result, a fault within the region of vulnerability completely severs the transmission corridor. Analogously, this scenario could be conceived as having scheduled maintenance on one of the transmission lines and a single hidden failure. Under such circumstance, a status input variable should be included in the learning sample.

The learning sample is developed using the procedure described in Section 4.2.1. Loading conditions were systematically modified to generate 4150 different system operating points; out of those 4150 system states, 2514 cases were classified as one and 1636 as zero. Cases classified as one represent "stressed" system conditions and, under such circumstances, a favorable bias towards security is desired, i.e., the voting scheme should be armed. Cases classified as zero identify "safe" system states and therefore, a bias towards dependability is preferred; the voting scheme should be disarmed and a single relay performs the protective function.

To infer the system state, it is assumed that PMUs are placed at all 500 kV buses in the system. Figure 5-5 shows a schematic of 500 kV buses and lines in California. The learning sample consists of 132 attributes: voltage angles and the rectangular decomposition into real and imaginary of the current flowing through 500 kV transmission lines; angles are measured in degrees and currents in per unit. Table 5-3 depicts the learning sample *L*; it has 4510 measurement vectors (rows) and 132 attributes[29] (columns). Further attributes were initially considered. However, optimal results were obtained with the attributes proposed in Table 5-3.

Table 5-3. Learning Sample: Heavy Winter Model

	Class	θ_{GATES}	θ_{DIABLO}	θ_{MIDWAY}	**…**	Ir_{1106}	Ii_{1106}	**…**	Ii_{3850}
x_1	1	-3.91	2.57	-5.89	…	5.40	1.78	…	-0.09
x_2	0	-2.52	4.14	-3.99	…	3.97	1.83	…	0.16
x_3	1	-3.95	2.52	-5.89	…	5.14	1.69	…	-0.13
x_4	1	-3.68	2.84	-5.52	…	4.92	1.72	…	-0.08
...	…	…	…	…	…	…	…	…	…
x_{4150}	0	-3.00	3.61	-4.62	…	4.36	1.77	…	-0.06

The main hypothesis in this dissertation is that a few strategic PMU measurements are sufficient to recognize the need to adjust the security/dependability balance of the adaptive protection scheme. Regularity patterns in the data are mined using CART's algorithm to grow a decision tree; the Matlab code can be found in Appendix A. The results were validated using a commercial implementation of CART by Salford Systems [69].

As discussed in Chapter 3, the particular choice of an impurity function tends to have little effect on the final tree. In this dissertation, the Gini impurity index is used. A sequence of minimal cost-complexity subtrees is produced by the decision tree algorithm; the subtree sequence $\{T_{max}, T_1, T_2, T_3, T_4, T_5\}$ is shown in Figure 5-8.

[29] Ir and Ii represent the real and imaginary decomposition of the current flowing through a transmission line respectively.

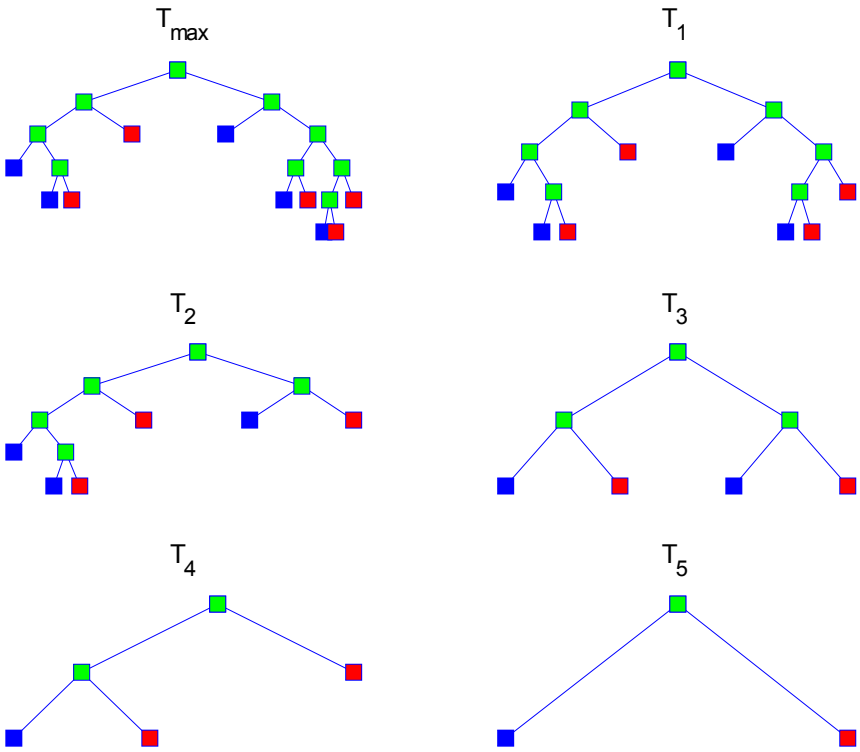

Figure 5-8. Sequence of subtrees generated through cross-complexity pruning. Subtree T_2 in the sequence is proposed as the final classification tree.

The selection of a right sized tree is made based on classification accuracy and tree complexity. In general, a parsimonious principle is invoked: simple models are preferred over complex ones. In the case of Decision Trees, simplicity is associated with tree size, which is measured as the total number of terminal nodes. For our particular application, parsimony has a practical interpretation: fewer nodes imply fewer PMU units deployed which in turn reduces investment costs. Typically, PMU devices have a negligible cost. The major investment is associated with the communication network infrastructure needed to transmit PMU data.

A plot of the estimated misclassification rate for each subtree is shown in Figure 5-5. The estimator used is known as V-fold cross-validation and it is thoroughly discussed in Chapter 3. Subtree T_2 (see Figure 5-8) is selected as the final decision tree since it attains the best balance between classification accuracy and tree complexity; a detailed description of the tree is shown in Figure 5-10. The tree has 6 terminal nodes and an estimated misclassification rate of approximately 1%. Note that the subtree with minimum misclassification rate is T_1 with an error

rate of 0.9%. However, subree T_2 achieves similar accuracy with one less PMU (a detailed description of subtree T_1 can be found in Appendix B).

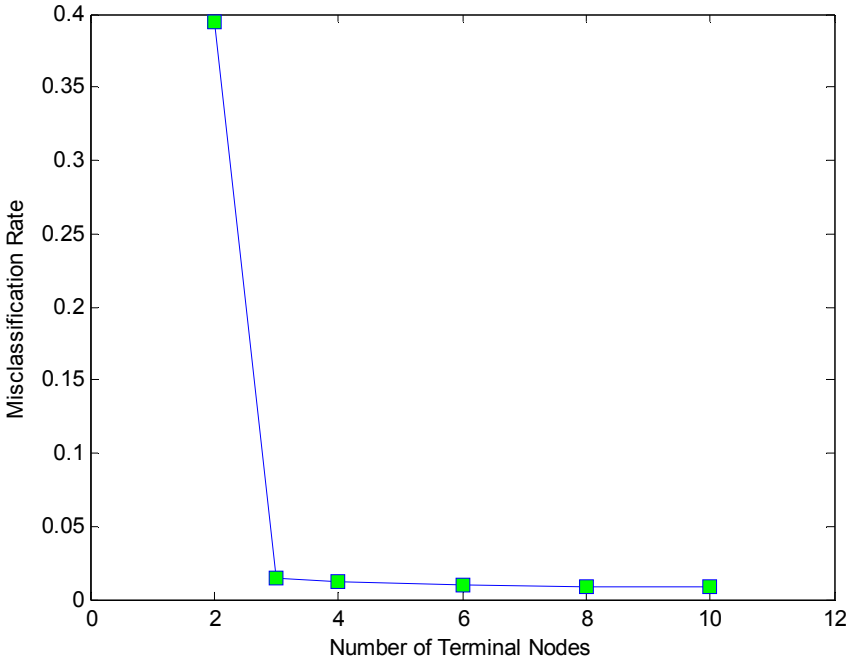

Figure 5-9. Cross-validation estimation of the misclassification rate.

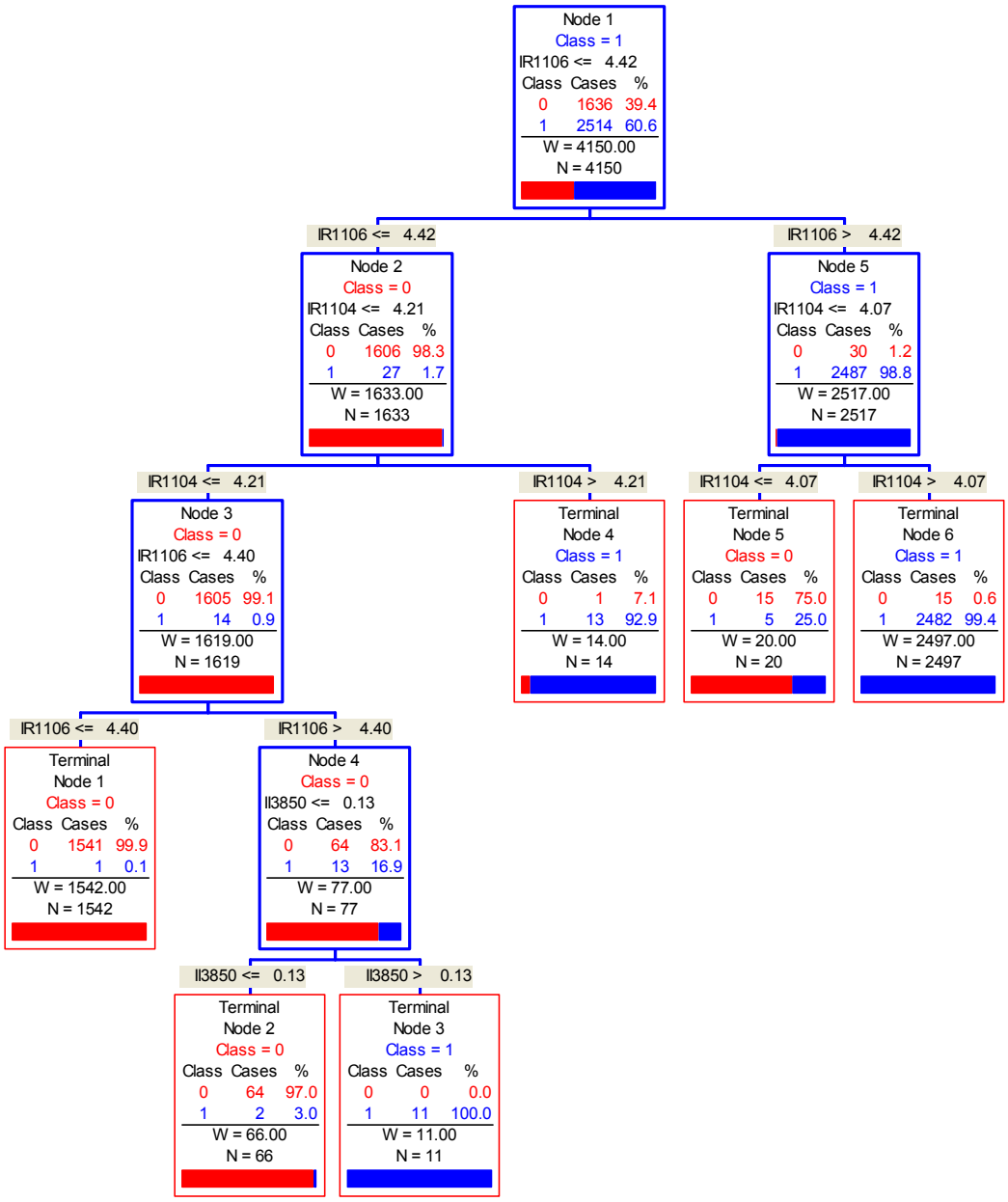

Figure 5-10. Detailed description of the proposed decision tree[30].

[30] Schematic obtained using a commercial implementation of CART by Salford Systems.

Partitioning the sample space

As stated in Chapter 3, the goal of the data mining algorithm is to extract rules or knowledge from regularity patterns exhibited by the data. Decision Trees recursively partition the sample space with hyper-planes to uncover knowledge. In order to better illustrate the underlying idea, consider the plot shown in Figure 5-11. The figure depicts a plot of all the contemplated attributes in the learning sample. Blue dots in the plot represent measurements taken under a "safe" system state (class labeled as zero). Under such circumstances a bias towards dependability is desired. PMU measurements taken under "stressed" conditions are colored in red (class labeled as one); on those situations a biased towards security is beneficial. In order to develop decision rules to adjust the security/dependability balance of the protection scheme, the goal is to discriminate between blue dots (class 0) and red dots (class 1) in the figure by subsequently partitioning the learning sample with planes.

Figure 5-12 shows a two-dimensional plot of the first split in tree; is Ir1106 $\leq$ 4.4249? The attribute Ir1106 represents the real current flowing through line 1106 in the model; a 500 kV transmission line connected between Tesla and Los Banos. The increase in homogeneity achieved by the first split is outstanding. If a unique PMU where to be used to adjust the security/dependability balance, it would have an error rate of approximately 4%. It can be observed in the figure that several blue dots lie above the splitting line and some red dots below the line. Subsequent partitions in the following branches of the tree are able to further reduce the misclassification rate to about 1%. The complete sequence of partitions of the decision tree can be found in Appendix B.

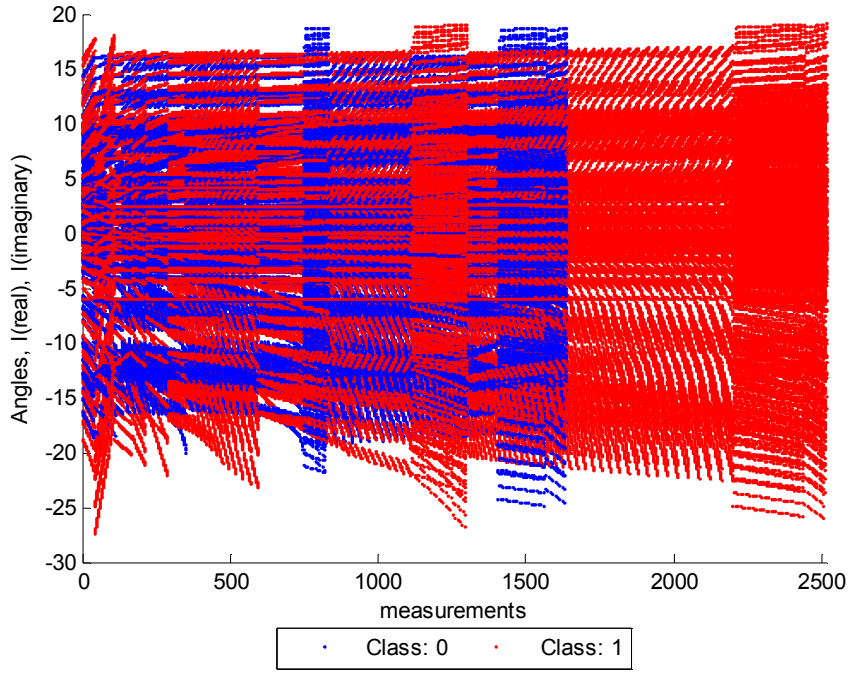

Figure 5-11. Plot of all attributes in the learning sample *L*.

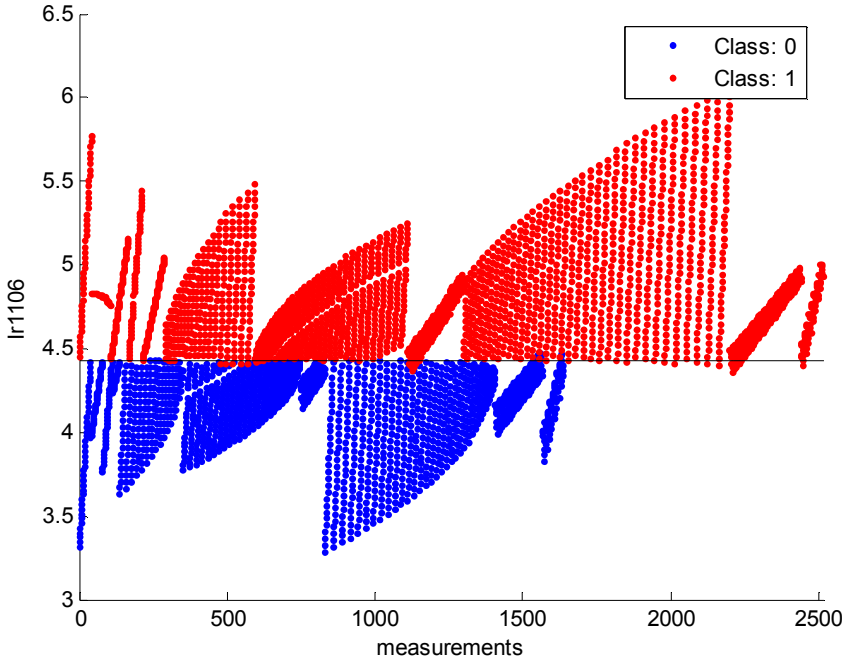

Figure 5-12. First partition of the sample space; optimal attribute: real current flowing between Tesla – Los Banos.

PMU Placement: Primary Splits and Surrogates

Splitting nodes of the decision tree (see Figure 5-10) indicate the desired location of PMUs. Table 5-4 summarizes the attributes used to partition the sample space. PMUs are required at the following locations:

- Los Banos: note that current flows through two different transmission lines need to be measured.
- Devers.
- Pittsburg: system reference.

To increase robustness and reliability, further PMUs may be deployed to measure surrogate attributes. The objective of a surrogate is to maximize the predictive association with the primary split. Surrogates attempt to mimic the partition achieved by the primary split and are therefore handy in cases where the information of the primary split is missing; failure in the communication link, PMU malfunction, etc. Table 5-5 shows the best surrogate of each primary split. The higher the predictive association, the better the surrogate mimics the primary split.

The schematic shown in Figure 5-13 depicts the final PMU placement. Primary splits are shown in green and surrogates in blue. As expected, a wide area perspective of the system is needed for an optimal performance of the decision tree.

Table 5-4. Splitting attributes of the Decision Tree.

Attribute	PMU measurement
Ir1106	Real Current: Tesla – Los Banos
Ir1104	Real Current: Tracy – Los Banos
Ii3850	Imaginary Current: Palo Verde - Devers

Table 5-5. List of surrogates. The predictive association measures how well the surrogate mimics the primary split.

Node	Primary Split	Surrogate	Predictive Association
1	Ir1106 ≤ 4.42	Ir1104 ≤ 4.16 (Tesla – Los Banos)	0.93
2	Ir1104 ≤ 4.21	Angle Round MT ≤ 16.88	0.64
3	Ir1104 ≤ 4.07	Ii1115 ≤ -2.02 (Gates - Diablo)	0.75
4	Ir1106 ≤ 4.4	Ir1104 ≤ 4.15	0.52
9	Ii3850 ≤ 0.13	Ir87 ≤ 5.04 (Victorville - McCulloug)	0.55

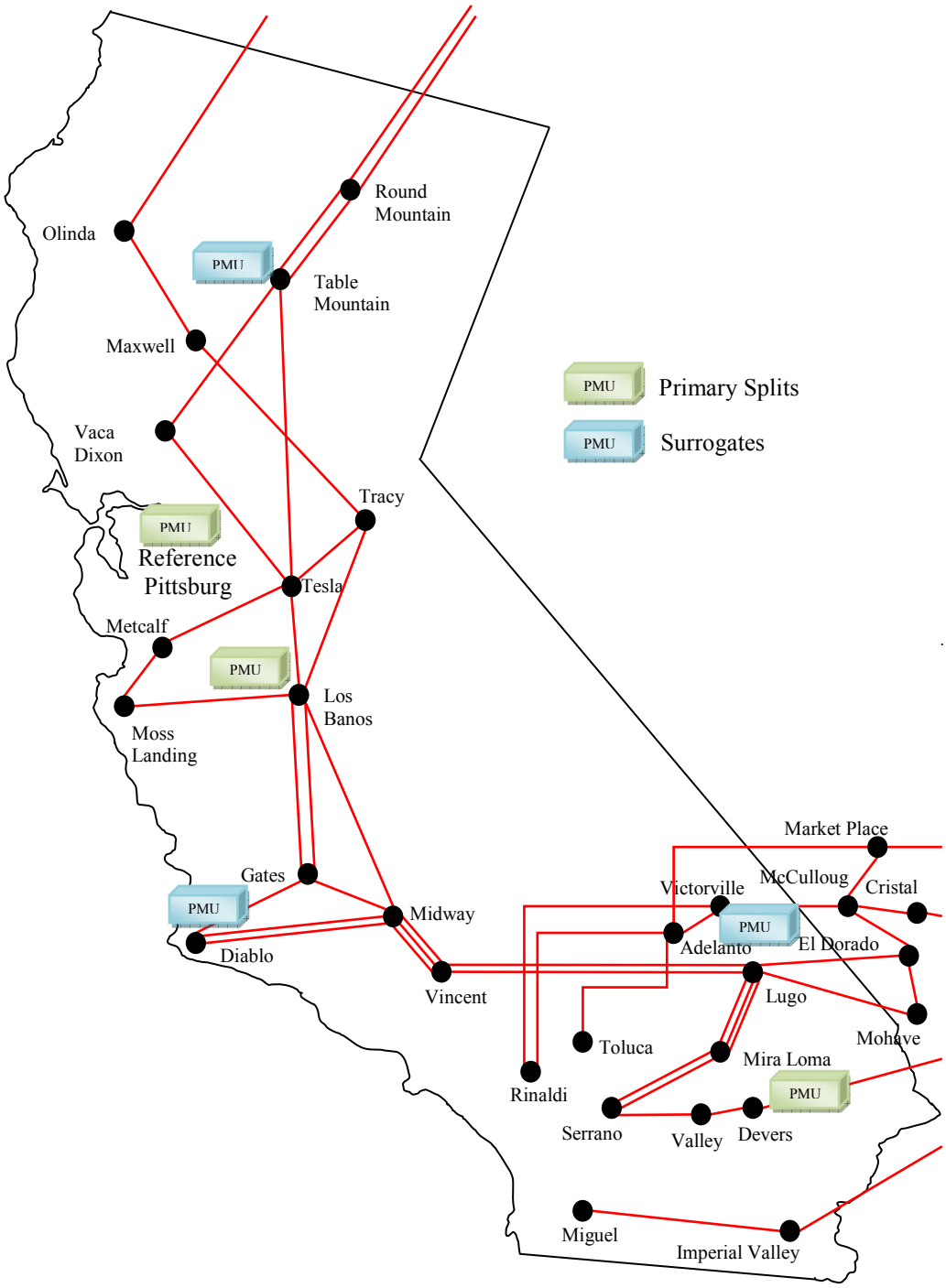

Figure 5-13. PMU placement for Heavy Winter Decision Tree. PMUs in green represent primary splits. PMUs in blue represent surrogates.

5.2.3 Out of Sample Testing: Heavy Winter

In order to test the performance of the decision tree with out-of-sample data, further test cases can be created by simulating circuit element outages. The objective is to induce additional system operating points to assess the robustness of the tree to topology changes. The out-of-sample data consists of 660 system operating conditions obtained by simulating outages in:

- Generators delivering more than 200 MW.
- Loads consuming more than 200 MW.
- Transmission lines: 230 kV and 500 kV.

Each of these outages were simulated under diverse loading conditions. The results of the test are summarized in Table 5-6, Table 5-7, Table 5-8, and Table 5-9. Out of the 660 cases, 14 cases were misclassified by the decision tree; an error rate of approximately 2%. Out of those 14 cases, only 2 "stressed" states were misclassified as class zero. This results show an outstanding performance of the decision tree. As stated previously, if the system undergoes significant departures from the model assumptions, a new decision tree should be trained. The proposed out-of-sample test only attempts to assess tree robustness under small departures.

Table 5-6. Out of sample test: generator outage.

	Classified class 0	**Classified class 1**
True class: 0	30	5
True class: 1	0	45

Table 5-7. Out of sample test: load outage.

	Classified class 0	**Classified class 1**
True class: 0	117	1
True class: 1	0	50

Table 5-8. Out of sample test: 230 kV lines outage.

	Classified class 0	Classified class 1
True class: 0	132	0
True class: 1	0	132

Table 5-9. Out of sample test: 500 kV lines.

	Classified class 0	Classified class 1
True class: 0	62	6
True class: 1	2	78

5.2.4 Decision Tree: Heavy Summer Model

As portrayed in Figure 5-6 and Figure 5-7, the prevailing condition in the heavy summer model is more stressed than in heavy winter. The power consumed doubles and therefore transmission line loading increases significantly. In order to demonstrate the methodology, a single hidden failure at the critical location is considered. The learning sample is developed using the procedure described in Section 4.2.1. Loading conditions were systematically modified to generate 11367 different system operating points; out of those 11367 system states, 5363 cases were classified as one and 6004 as zero. Cases classified as one represent "stressed" system conditions and under such circumstances a favorable bias towards security is desired, i.e., the voting scheme is armed. Cases classified as zero identify "safe" system states and a bias towards dependability is preferred, i.e., the voting scheme is disarmed and a single relay performs the protective function.

Following the same procedure as in the heavy winter model, in order to infer the system state it is assumed that PMUs are placed at every 500 kV bus in the system. The learning sample consists of 132 attributes: voltage angles and the rectangular decomposition into real and imaginary of the current flowing through 500 kV transmission lines; angles are measured in

degrees and currents in per unit. The learning sample L has 11367 measurement vectors (rows) and 132 attributes (columns).

The sequence of minimal cost-complexity subtrees $\{T_{max}, T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9\}$ is shown in Figure 5-15. The selection of a right sized tree is made based on classification accuracy and tree complexity. Figure 5-14 shows a plot of the estimated misclassification rate for each subtree. Subtree T_6 (see Figure 5-15) is selected as the final decision tree since it attains the best balance between classification accuracy and tree complexity; a detailed description of the tree is shown in Figure 5-16. The tree has 6 terminal nodes and an estimated misclassification rate of approximately 1%.

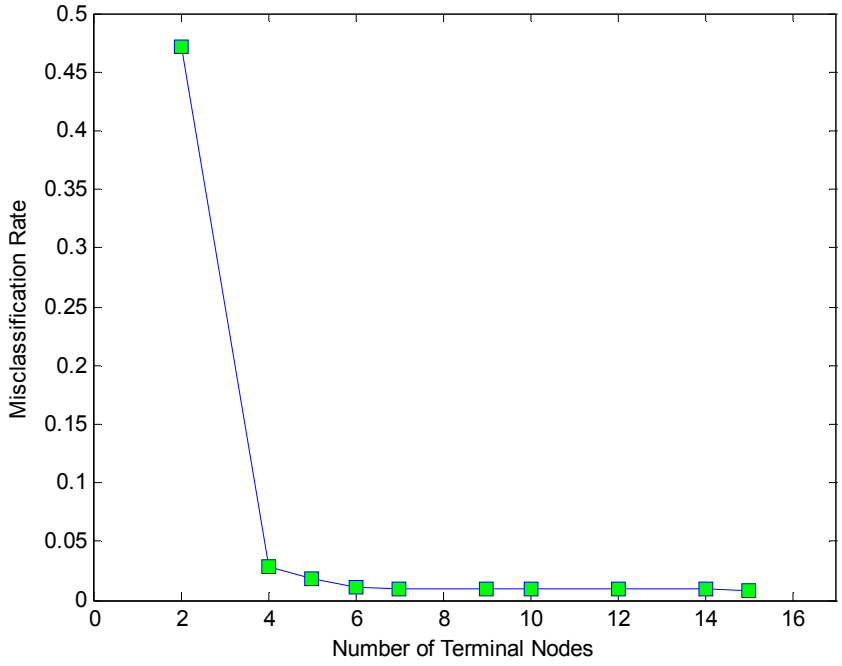

Figure 5-14. Misclassification rate for the sequence of subtrees.

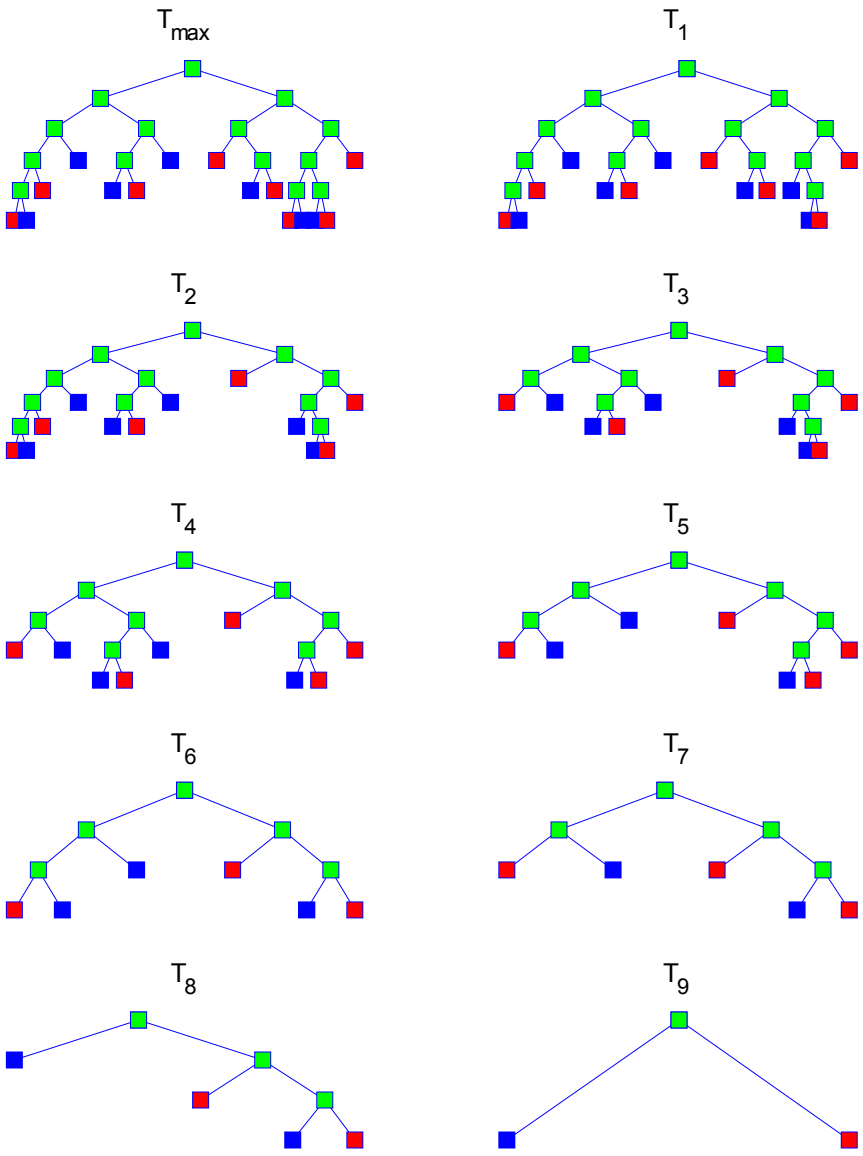

Figure 5-15. Sequence of subtrees generated through cost complexity pruning. Subtree T_6 in the sequence is proposed as the optimal classification tree.

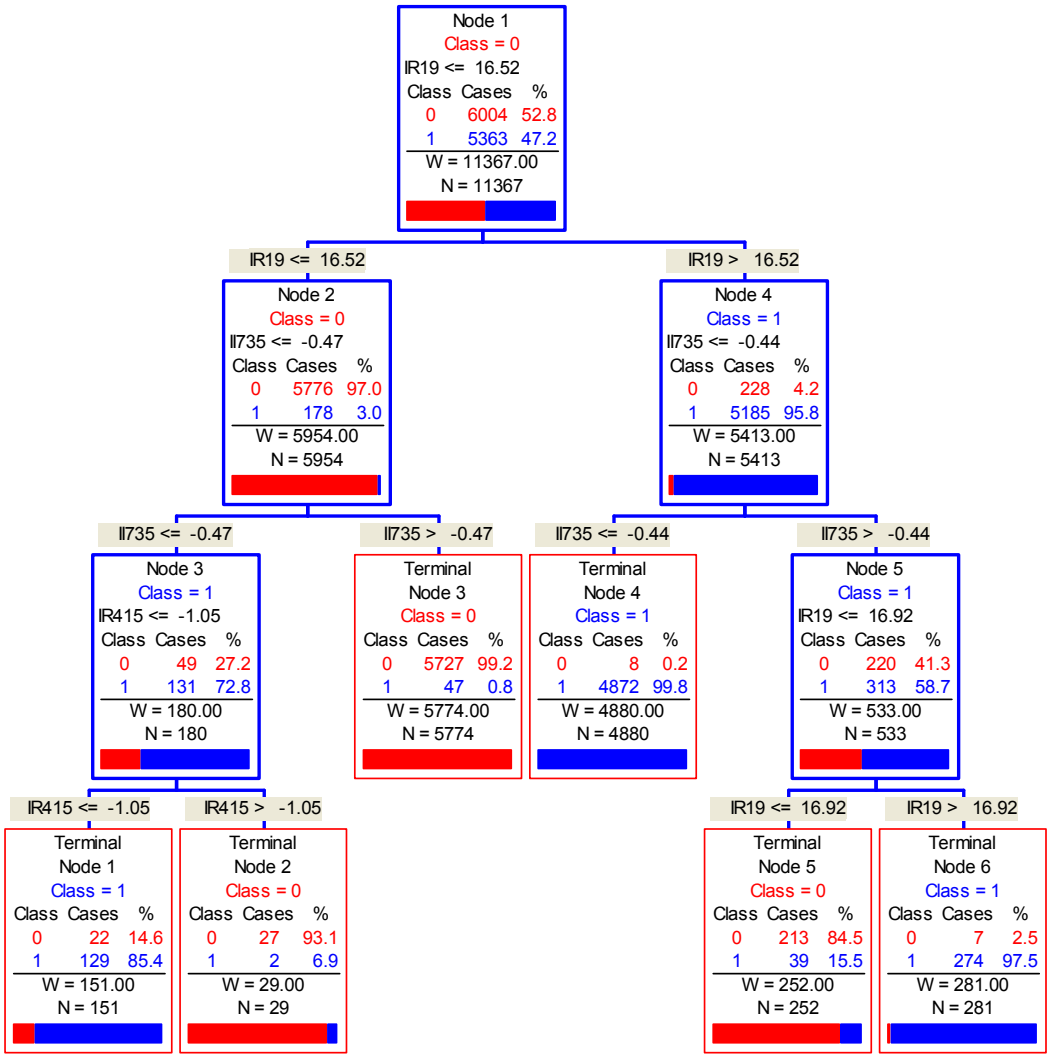

Figure 5-16. Detailed tree description of T_8. The tree has a misclassification rate of approximately 1%.

Partitioning the sample space

Decision rules to recognize the need to adjust the security/dependability balance of the protection scheme are developed by a sequence of partitions of the sample space. Figure 5-17 shows a two-dimensional plot of the first split in tree; is Ir19 $\leq$ 16.52? The attribute Ir19 represents the real current flowing through line number 19 in the model; a 500 kV transmission line connected between Devers and Palo Verde. Blue dots in the figure represent "safe" states (class 0, bias towards dependability) and red dots "stressed" conditions (class 1, bias towards security).

The first split achieves an outstanding increase in homogeneity. If a unique PMU where to be used to adjust the security/dependability balance, it would have an error rate of just 4.7%. It can be observed in the figure that several blue dots lie above the splitting line and some red dots below the line. Subsequent partitions further down the tree are able to reduce the misclassification rate to approximately 1%. The complete sequence of partitions of the decision tree can be found in Appendix B.

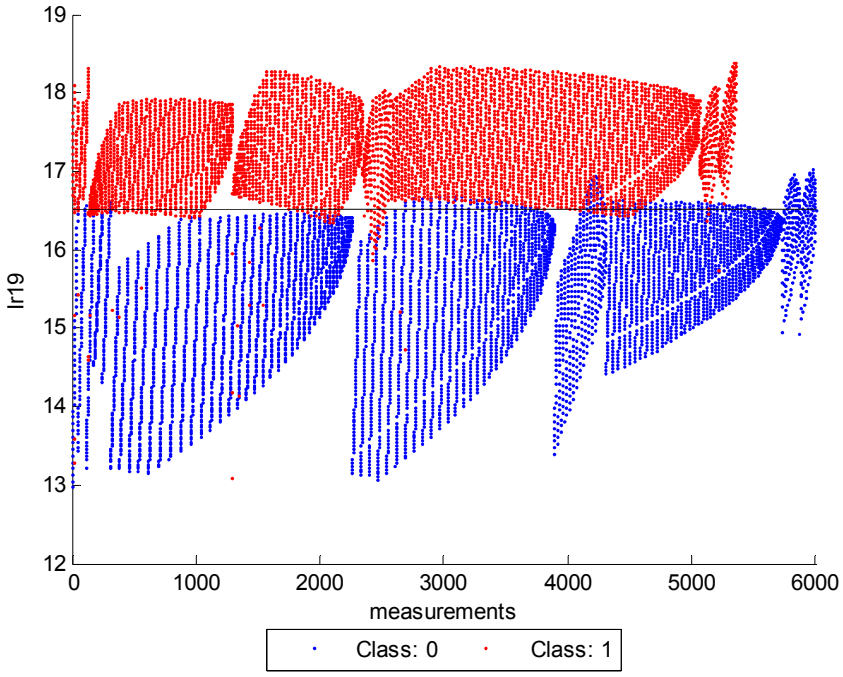

Figure 5-17. Split at the root node of T_8.

PMU Placement: Primary Splits and Surrogates

Splitting nodes of the decision tree (see Figure 5-16) indicate the desired location of PMUs. Table 5-4 summarizes the attributes used to partition the sample space. PMUs are required at the following locations:

- Devers: note that current flows through two different transmission lines need to be measured.
- El Dorado.
- Pittsburg: system reference.

To increase robustness and reliability further PMUs may be deployed to measure surrogate attributes. As stated in Chapter 3, the objective of a surrogate is to mimic the partition achieved by the primary split. Table 5-5 shows the best surrogate of each primary split. The higher the predictive association, the better the surrogate mimics the primary split. The schematic shown in Figure 5-18 depicts the PMU placement. Primary splits are shown in green and surrogates in blue.

Table 5-10. Splitting attributes of the Decision Tree.

Attribute	PMU measurement
Ir19	Real Current: Palo Verde – Devers
Ii735	Imaginary Current: Devers – Valley SC
Ir415	Real Current: El Dorado - McCullough

Table 5-11. List of surrogates. The predictive association measures how good the surrogate mimics the primary split.

Node	Primary Split	Surrogate	Predictive Association
1	Ir19 ≤ 16.52	Ir472 ≤ -4.98 (Mohave – El Dorado)	0.93
2	Ii735 ≤ -0.47	Ii1033 ≤ 1.38 (Diablo - Midway)	0.17
3	Ii735 ≤ - 0.4.4	Ii1033 ≤ 1.38 (Diablo - Midway)	0.72
4	Ir415 ≤ - 1.05	Ii1022 ≤ 1.53 (Moss Landing – Los Banos)	0.79
7	Ir19 ≤ 16.92	Ir472 ≤ -5.26 (Mohave – El Dorado)	0.78

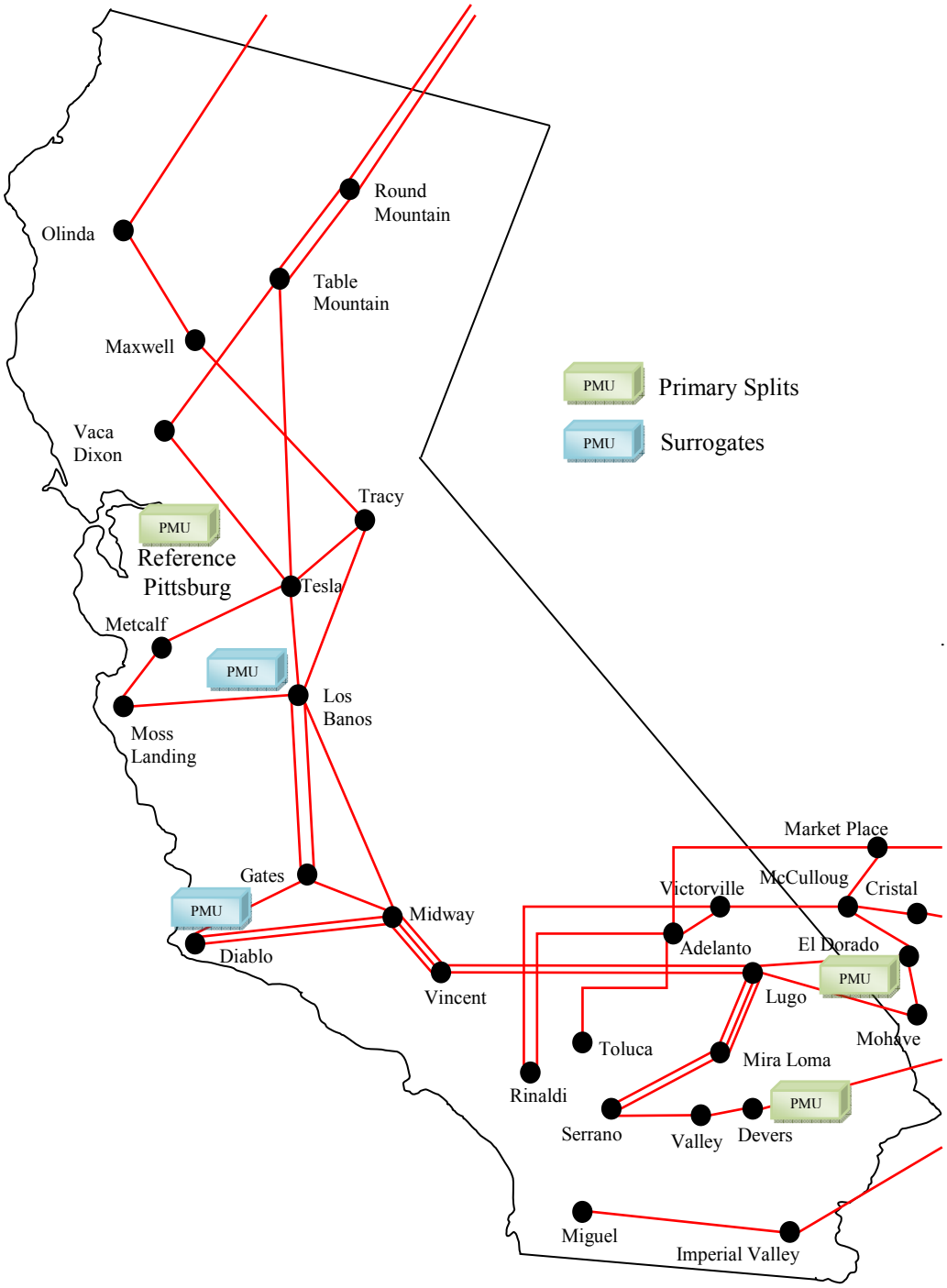

Figure 5-18. PMU placement for Heavy Summer Decision Tree. PMUs in green represent primary splits. PMUs in blue represent surrogates.

5.2.5 Out of Sample Testing: Heavy Summer

In order to test the robustness of the decision tree to small departures, test cases, not included in the learning sample, are created by simulating outages. As stated previously, the objective is to assess robustness against topology changes. The out-of-sample data consists of 1138 system operating conditions obtained by simulating outages in:

- Generators delivering more than 200 MW.
- Loads consuming more than 200 MW.
- Transmission lines: 230 kV and 500 kV.

Each of these outages were simulated under diverse loading conditions. The results of the test are summarized in Table 5-12, Table 5-13, Table 5-14, and Table 5-15. Out of the 1137 cases, 49 cases were misclassified by the decision tree; an error rate of approximately 4.3%. The tree has an adequate performance when subjected to topology changes.

Table 5-12. Out of sample test: generator outage.

	Classified class 0	**Classified class 1**
True class: 0	107	2
True class: 1	6	112

Table 5-13. Out of sample test: load outage.

	Classified class 0	**Classified class 1**
True class: 0	154	0
True class: 1	7	37

Table 5-14. Out of sample test: 230 kV lines outage**.**

	Classified class 0	**Classified class 1**
True class: 0	278	0
True class: 1	25	284

Table 5-15. Out of sample test: 500 kV lines.

	Classified class 0	**Classified class 1**
True class: 0	62	6
True class: 1	3	54

5.2.6 Conclusion

The proposed methodology was put into practice using two seasonal, highly-detailed, models of California. Simulation results confirm the initial hypotheses:

- Decision Trees can be used to uncover regularity patterns associated with power system operating points. The reliability balance of a critical protection system is adjusted to suit prevailing system conditions. The misclassification rate of both seasonal decision trees, heavy winter and heavy summer, is approximately 1%. Out-of-sample tests indicate that the decision trees grown are robust to small departures from topology assumptions.
- Strategically placed PMUs are sufficient to infer prevailing system conditions. PMUs must be placed at four 500 kV buses. Aided with three extra PMUs, robustness to missing attributes can be achieved through surrogates. The overall PMU placement, contemplating the decision trees grown using both seasonal models, heavy winter and heavy summer, is shown in Figure 5-19.

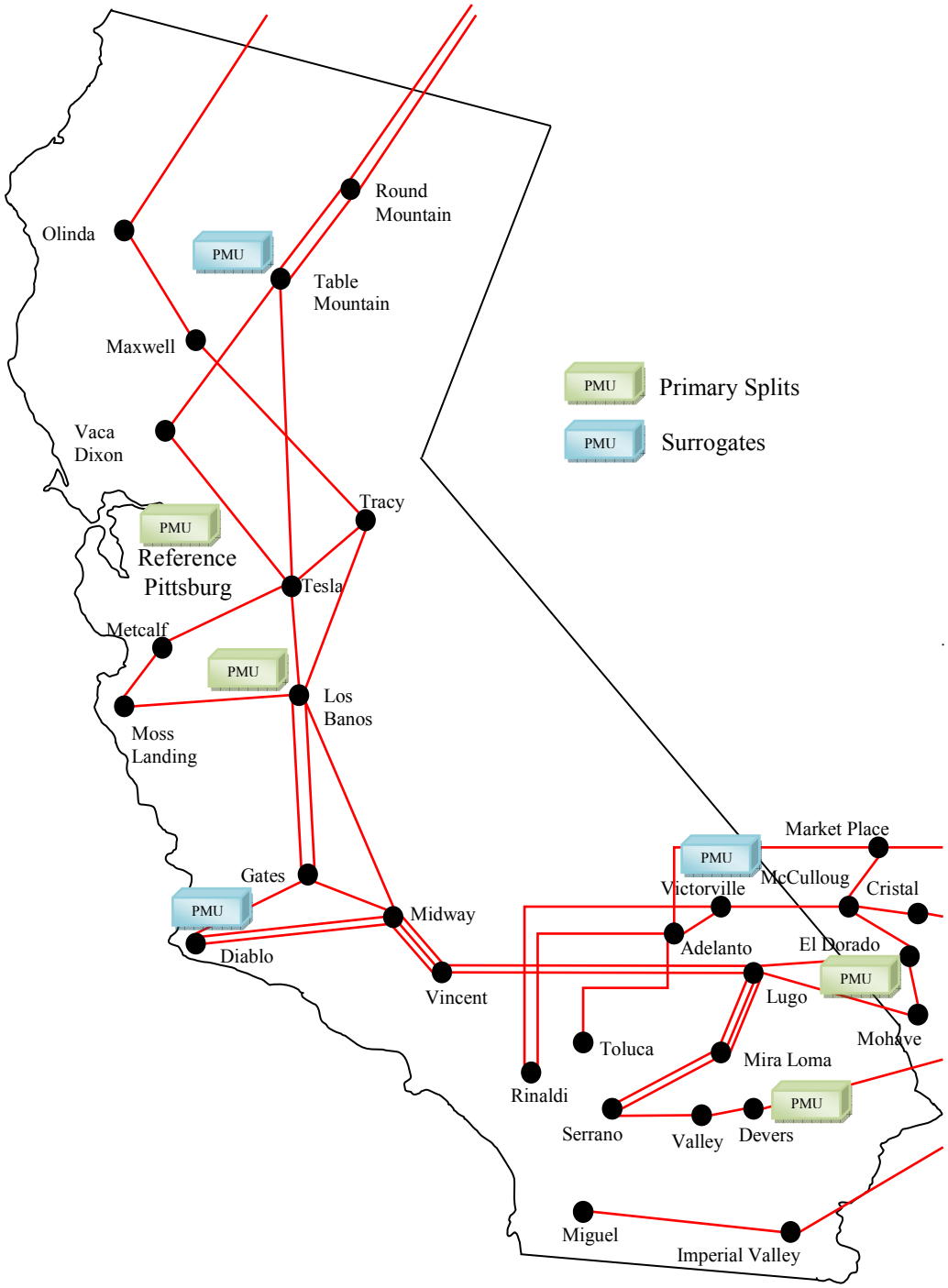

Figure 5-19. Overall PMU placement contemplating seasonal decision trees: heavy winter and heavy summer.

Chapter 6 Conclusions and Future Research

Conventional relays react in a predetermined and fixed manner and are typically biased towards dependability. These dormant sentinels protect the grid with purely local information and are not able, by design, to adjust to prevailing system conditions. Experience shows that such rigid relay settings may become unreliable or suboptimal under abnormal stressed conditions. Hidden failures in protection relays can have catastrophic consequences. By their own nature, hidden failures are prone to manifest themselves under stressed system conditions, and therefore, their consequence is rather significant. An analysis of NERC reports indicate that hidden failures are involved in over 70% of power system blackouts [13]; testimony of the crucial role played by protection relays in a reliable power system.

It is argued in this dissertation that embracing the paradigm shift of adaptive protection leads to a new realm of opportunities. The concept of adaptive relaying can be traced back to the origins of digital relays, and yet, very few adaptive schemes have been designed and implemented. Three principles summarize the adaptive philosophy advocated in this dissertation:

1. ***Adaptive Relaying:*** is defined as the ability of relays to change their settings, operation, or logic to adapt to prevailing system conditions [8].
2. ***Critical Locations:*** sites in power system where embracing adaptive schemes would be most beneficial.
3. ***Adaptability Scope:*** WAMs should not directly intervene with high speed protection.

The main objective of this dissertation is to propose methodology to implement a security/dependability adaptive protection scheme. Motivation, hypothesis, conclusions, and main contributions are discussed in the following sections.

6.1 Conclusions

The methodology proposed in this dissertation aims to reduce the likelihood of hidden failures and potential cascading events by adjusting the security/dependability balance of protection systems. Aided with Wide Area Measurements, the scheme tailors the security/dependability balance to better suit prevailing system conditions. When the power system is in a "safe" state, a bias towards dependability is desired. Under such conditions, not clearing a fault with primary protection has a greater impact on the system than a relay miss-operation due to lack of security. However, when the power system is in a "stressed" state, unnecessary line trips can greatly exacerbate the severity of the outage, contribute to the geographical propagation of the disturbance, and may even lead to cascading events and subsequent blackouts. Under such states, it is desirable to alter the reliability balance in favor of security.

The main hypothesis in this dissertation is that few, strategically placed, PMU measurements are sufficient to recognize the need to alter the security/dependability balance of the adaptive protection scheme. Simulation results on a highly detailed 4000 bus model of California confirm the premise. Patterns associated with different system states can be uncovered with the aid of Wide Area Measurements and Data Mining algorithms. The proposed method, Decision Trees, has proved to be highly adept to the task of mining knowledge in non-linear systems. As a further advantage, DTs provide an intuitive description of the uncovered knowledge. The systematic procedure used by DTs to make induction inferences also resembles the thinking process of engineers.

The optimal location for the adaptive protection scheme was derived using a systematic procedure to identify and rank critical location in power systems. The critical location was confirmed, based on practical experience, by the advisory committee of the VT-CIEE research project.

Simulation results show that the proposed adaptive scheme has a misclassification rate of 1%. Approximately half of the operating conditions considered in the learning sample were classified as "stressed". Therefore, assuming a hidden failure on a traditional protection relay, if a disturbance were to occur within its region of vulnerability, the consequence of it would be

rather noteworthy in 50% of the cases. The adaptive scheme is able to reduce the proportion of system states in which a single hidden failure has a significant impact, and in a sense, system states that are prone to manifest a hidden failure, to less than 1%.

A natural question would be to ask what happens in situations in which the scheme misclassifies a case, i.e., what is the consequence of the cases that account for the 1% error rate. The advocated adaptive protection scheme is susceptible to two types of errors:

- Type I: fail to vote when a bias towards security would be desirable. This circumstance characterizes the current protection practice, that is, a single protective relay typically biased towards dependability. Therefore, this error, though potentially extremely harmful, does not go in detriment of any existing practice.
- Type II: vote when a bias towards dependability would be preferred. A customary practice to increase security is to implement a voting scheme in which three relays continuously vote, regardless of prevailing conditions. Therefore, under this type of error, the scheme again reduces to current practices.

To conclude, the scheme presents a "win-win" situation. When it correctly predicts the appropriate security/dependability balance, which does, according to simulations, approximately 99% of the times, it reduces the likelihood of the manifestation of a hidden failure under stressed conditions, potentially preventing a cascading sequence of events. When it errors, no harm is done since it responds in the same manner as ordinary protection relays. The proposed scheme is an improvement over current practices.

6.2 Contributions:

Throughout this dissertation, emphasis was made on the importance of embracing the paradigm shift offered by adaptive protection. The general principles that constitute the philosophy of adaptive protection were delineated in Chapter 1. In Chapter 2, the potential benefits offered by adaptive schemes were illustrated by analyzing the role that traditional protection relays played on several power system blackouts.

The main core of this dissertation is concerned with developing methodology to implement an adaptive protection scheme that can alter its security/dependability balance to suit prevailing system conditions. The optimal location for the advocated adaptive scheme is determined using a systematic procedure, tailored made for large power systems, to identify and rank critical locations in power systems. The design of the adaptive scheme is based on Wide Area Measurements and Data Mining. Decision Trees were proved to be adept at extracting knowledge from regularity patterns exhibited in the learning sample, rendering a simple logic to adjust the reliability balance of the protection scheme. The required PMU measurements are also determined by the DT, i.e., the number of devices needed is minimized by the data mining algorithm; an application oriented PMU placement.

To conclude, the following is a summary of the main contributions of this dissertation.

- A systematic methodology to identify and rank *Critical Locations* in large power systems.
- Methodology to implement a security/dependability voting scheme based on *Wide Area Measurements* and *Decision Trees*.
- A Matlab implementation of CART.
- An application oriented PMU placement method.

6.3 Future Research

A natural extension of the proposed methodology would be to include historical daily load curves in the learning sample. The load flow model can be tuned to match true snapshots of the power system. The systematic load scaling procedure used to generate diverse operating points may also be improved by implementing an economic dispatch algorithm. Due to the intrinsic characteristics of load flow analysis, it is implicitly assumed that the variations in load are matched by rescheduling the amount of imported generation[31]. Due to the manner in which the load variations were formulated, an inertial re-dispatch, where all machines in the system increase their generation proportionally to their inertia, does not effectively represent the system behavior. However, the economic dispatch should render sensible system states.

Another simple, yet highly appealing, extension of the work presented in this dissertation, would be to replace the load flow analysis with dynamic simulations. Such transition would allow the modeler to contemplate dynamic attributes, such as the rate of change in angle differences, which can potentially enhance the classification accuracy. Despite the fact that Decision Trees are trained offline, the computational power needed may be a strong deterrent.

The proposed methodology can be tailored to explore the application of DTs to other adaptive protection systems. As stated previously, the induction inferences made by DTs resemble the thinking process of engineers, and therefore, their use can be extended to other protection systems. Consider, for example, the intelligent load shedding scheme described in Chapter 2, section 2.1.3. Traditional UFLS schemes detect the onset on frequency decay and, ones the frequency reaches a threshold value, predetermined quantities of load are shed. It was argued that under particular system conditions ("stressed"), it is too costly and inefficient to wait for a drop in frequency. During the major disturbance in the area regulated by the Union for the Coordination of Transmission of Electricity (UCTE, now called European Network of Transmission System Operators for Electricity – ENSO-E) of 2006, load shedding schemes performed as designed but their effectiveness was undermined by the generation lost precisely due to the decay in frequency. If corrective actions had been taken during the eight seconds that

[31] The model used in this dissertation has two equivalent swing generators that represent the interaction between California and the rest of the world. Both swing generators "pick up" the changes in load.

took the frequency to drop 1 Hz, the severity of the disturbance could have been greatly reduced. A wide area perspective of the system is bound to be a critical factor in the design of an intelligent load shedding scheme. The learning sample should include several attributes: inter-tie status and loading conditions, load consumption at key delivering points, voltage angles at backbone buses, machines' output power and reserve margins, a frequency measurement, ACE, etc. Under diverse operating points, dynamic simulations can be used to assess the need for immediate corrective actions. Decision Trees can recognize patterns in the data and derive the decision logic to shed load.

Finally, the successful application of DTs to classify system states with respect to a predefined critical location, suggests the extension of this work into real time security assessment. Significant research effort is being made towards visualization techniques for power system operators. The challenge is how to present to the operator, in a concise and effective manner, the vast amount of information provided by Wide Area Measurements. A major advantage of Decision Trees is their intuitive presentation of discovered knowledge. It was suggested in Chapter 4, that an upside-down interpretation of a path to a terminal node reveals a strategic course of action to modify prevailing conditions so as to return to a desired state. Therefore, the DTs would not only be able to classify the estimated state but also potentially suggest a course of action to the operator.

References

1. CAISO. www.caiso.com.
2. NASPI-NET. www.naspi.org.
3. UCTE, *Final Report: System Disturbance on 4 November 2006*. 2006.
4. NERC, P.S.O.T.F., *Final Report on the August 14, 2003 Blackout in he United States and Canada*. 2004.
5. Ruisheng, D., et al., *Decision Tree-Based Online Voltage Security Assessment Using PMU Measurements.* Power Systems, IEEE Transactions on, 2009. **24**(2): p. 832-839.
6. NYISO, *Blackout August 14, 2003 Final Report*. 2005.
7. UCTE, *Final Report of the Investigation Committee on the 28 September 2003 Blackout in Italy*. 2004.
8. Horowitz, S.H., A.G. Phadke, and J.S. Thorpe, *Adaptive transmission system relaying.* Power Delivery, IEEE Transactions on, 1988. **3**(4): p. 1436-1445.
9. IEEE, *Standard for Relays and Relay Systems Associated with Electric Power Apparatus*. 1989.
10. NERC, *Addressing the directives issued by FERC, in Order 706 relative to the approved Cyber Secuirty Standards CIP-002-1 through CIP-009-1*. 2008-06.
11. Kundur, P., *Power system stability and control*. 1994, New York ; London: McGraw-Hill. XXIII-1176 p.
12. Phadke, A.G., J.S. Thorp, and SpringerLink (Online service), *Synchronized phasor measurements and their applications*. 2008, Springer: New York. p. x, 247 p.
13. Tamronglak, S., *Analysis of Power System Disturbances due to Relay Hidden Failures*, in *ECE*. 1994, Virginia Tech: Blacksburg.
14. Elizondo, D., *Hidden Failures in Protection Systems and its Impact on Power System Wide Area Disturbances*, in *ECE*. 2000, Virginia Tech: Blacksburg.
15. *Northeast Power Failure: November 9 and 10, 1965*. 1965, Federal Power Commission.
16. Wang, H. and J.S. Thorp, *Optimal Locations for Protection System Enhancement: A Simulation of Cascading Outages.* Power Engineering Review, IEEE, 2001. **21**(7): p. 67-67.
17. Bae, K. and J.S. Thorp, *A stochastic study of hidden failures in power system protection.* Decision Support Systems, 1999. **24**(3-4): p. 259-268.
18. Fang, Y., et al. *Effects of Protection System Hidden Failures on Bulk Power System Reliability*. in *Power Symposium, 2006. NAPS 2006. 38th North American*. 2006.
19. Horowitz, S.H. and A.G. Phadke, *Power system relaying*. 2nd ed ed. 1995, Taunton, Somerset, England

New York: Research Studies Press ;

Wiley. xiv, 319 p.

20. EEI, *Edison Electric Institute Survey of Transmission Investment*. 2005.
21. EIA. *Energy Information Administration: www.eia.doe.gov*.
22. Ng, S.C. and E.A. Udren. *Business Case for Protection Reliability Improvement and Summary of Industry Survey of Protection Practices*. in *iPCGRID*. 2009. San Francisco, CA.
23. www.energy.ca.gov. *Energy Almanac, California Energy Commission*.

24. DOE. *Improving the Reliability of the U.S. Electric Grid*. 2009.
25. Witten, I.H. and E. Frank, *Data mining : practical machine learning tools and techniques*. 2nd ed. Morgan Kaufmann series in data management systems. 2005, Amsterdam ; Boston, MA: Morgan Kaufman. xxxi, 525 p.
26. Rovnyak, S., et al., *Decision trees for real-time transient stability prediction.* Power Systems, IEEE Transactions on, 1994. **9**(3): p. 1417-1426.
27. Rovnyak, S.M., C.W. Taylor, and Y. Sheng, *Decision trees using apparent resistance to detect impending loss of synchronism.* Power Delivery, IEEE Transactions on, 2000. **15**(4): p. 1157-1162.
28. Ruisheng, D., et al. *Decision tree assisted controlled islanding for preventing cascading events*. in *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*. 2009.
29. Voumvoulakis, E.M., A.E. Gavoyiannis, and N.D. Hatziargyriou. *Decision trees for dynamic security assessment and load shedding scheme*. in *Power Engineering Society General Meeting, 2006. IEEE*. 2006.
30. Zhiyong, L. and W. Weilin. *Phasor Measurements-Aided Decision Trees for Power System Security Assessment*. in *Information and Computing Science, 2009. ICIC '09. Second International Conference on*. 2009.
31. Martigne, H., et al. *Statistical method to determine operating rules in the event of generator dropout on EDF French Guyana Grid*. in *Power Tech Proceedings, 2001 IEEE Porto*. 2001.
32. Lobato, E., et al. *Decision Trees Applied to Spanish Power Systems Applications*. in *Probabilistic Methods Applied to Power Systems, 2006. PMAPS 2006. International Conference on*. 2006.
33. Huang, J.A., et al. *Application of data mining techniques for automat settings in emergency control at Hydro-Qudbec*. in *Power Engineering Society General Meeting, 2003, IEEE*. 2003.
34. Khatib, A.R., et al. *Real-time estimation of security from voltage collapse using synchronized phasor measurements*. in *Power Engineering Society General Meeting, 2004. IEEE*. 2004.
35. Centeno, V., *Adaptive out-of-step relaying with phasor measurements* in *Electrical Engineering*. 1995, Virginia Polytechnic Institute and State University: Blacksburg.
36. NERC, S.P.C.T.F., *Increase Line Loadability by Enabling Load Encroachment Functions of Digital Relays*. 2005.
37. Horowitz, S.H. and A.G. Phadke, *Third zone revisited.* Power Delivery, IEEE Transactions on, 2006. **21**(1): p. 23-29.
38. Arana, A., *Analysis of Electromechanical Phenomena in the Power-Angle Domain*, in *Electrical Engineering*. 2009, Virginia Tech: Blacksburg.
39. NERC, *Assessment of the Design and Effectiveness of UVLS Program*. 2005.
40. NERC, *UVLS System Maintenance and Testing*. 2005.
41. NERC, *Development and Documentation of Regional UFLS Programs*. 2005.
42. NERC, *Assuring Consistency with Regional UFLS Program Requirements*. 2005.
43. NERC, *Underfrequency Load Shedding Equipment Maintenance Programs*. 2005.
44. NERC, *UFLS Performance Following an Underfrequency Event*. 2005.
45. NERC, *Under-Voltage Load Shedding Program Data*. 2006.
46. NERC, *Under-Voltage Load Shedding Program Performance*. 2006.
47. NERC, *Review of Selected 1996 Electric System Disturbances in North America*. 2006.

48. United States. Federal Energy Regulatory Commission., *The Con Edison power failure of July 13 and 14, 1977 : final staff report*. 1978, Washington, D.C.: The Commission : for sale by the Supt. of Docs., U.S. Govt. Print. Off. v, 208 p.
49. Andersson, D., et al. *Intelligent load shedding to counteract power system instability*. in *Transmission and Distribution Conference and Exposition: Latin America, 2004 IEEE/PES*. 2004.
50. Bonian, S., X. Xiaorong, and H. Yingduo. *WAMS-based Load Shedding for Systems Suffering Power Deficit*. in *Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES*. 2005.
51. El Azab, R.M., E.H.S. Eldin, and M.M. Sallam. *Adaptive Under Frequency Load Shedding using PMU*. in *Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on*. 2009.
52. Mori, H. *State-of-the-art overview on data mining in power systems*. in *Power Engineering Society General Meeting, 2006. IEEE*. 2006.
53. Taylor, C.W., et al., *A New Out-of-Step Relay with Rate of Change of Apparent Resistance Augmentation.* Power Apparatus and Systems, IEEE Transactions on, 1983. **PAS-102**(3): p. 631-639.
54. Sheng, Y. and S.M. Rovnyak, *Decision Trees and Wavelet Analysis for Power Transformer Protection.* Power Engineering Review, IEEE, 2002. **22**(2): p. 62-62.
55. El-Arroudi, K. and G. Joos, *Data Mining Approach to Threshold Settings of Islanding Relays in Distributed Generation.* Power Systems, IEEE Transactions on, 2007. **22**(3): p. 1112-1119.
56. Bernabeu, E., *Robust State Estimation in Power Systems*. 2009.
57. *Classification and regression trees*. 1984, Belmont, Calif: Wadsworth International Group. x, 358 p.
58. Quinlan, J.R., *C4.5 : programs for machine learning*. 1993, San Mateo, Calif: Morgan Kaufmann Publishers. X-302 p.
59. Elizondo, D.C., *A Methodology to Assess and Rank the Effects of Hidden Failures in Protection Schemes based on Regions of Vulnerability and Index of Severity*, in *Electrical and Computer Engineering*. 2003, Virginia Tech: Blacksburg.
60. NERC, *Standard TPL-003-0 System Performance Following Loss of Two or More BES Elements*. 2005.
61. Clair, H.P.S., *Practical Concepts in Capability and Performance of Transmission Lines.* Power Apparatus and Systems, Part III. Transactions of the American Institute of Electrical Engineers, 1953. **72**(2): p. 1152-1157.
62. Gutman, R., P.P. Marchenko, and R.D. Dunlop, *Analytical Development of Loadability Characteristics for EHV and UHV Transmission Lines.* Power Apparatus and Systems, IEEE Transactions on, 1979. **PAS-98**(2): p. 606-617.
63. Gutman, R., *Application of line loadability concepts to operating studies.* Power Systems, IEEE Transactions on, 1988. **3**(4): p. 1426-1433.
64. Mikolinnas, T.A. and B.F. Wollenberg, *An Advanced Contingency Selection Algorithm.* Power Apparatus and Systems, IEEE Transactions on, 1981. **PAS-100**(2): p. 608-617.
65. Bernabeu, E. and A. Arana, *A Comprehensive Approach to Identifying and Ranking the Critical Locations of a Power System*. 2007, Virginia Tech: Blacksburg.

66. Altman, J.R., *A practical comprehensive approach to PMU placement for full observability*, in *Electrical and Computer Engineering*. 2008, Virginia Polytechnic Institute and State University: Blacksburg.
67. Brueni, D., *Minimal PMU placement for graph observability : a decomposition approach*, in *Electrical and Computer Engineering*. 1993, Virginia Polytechnic Institute and State University: Blacksburg.
68. Francisco, N.R., *State Estimation and Voltage Security Monitoring Using Synchronized Phasor Measurements*, in *ECE*. 2001, Virginia Tech: Blacksburg.
69. CART. *Salford Systems, salford-systems.com*.

Appendix A

A.1 Matlab implementation of CART

The algorithms presented in Chapter 3 were implemented in Matlab. As stated previously, only classification trees were considered. The code is open source for educational purposes and may not be commercialized.

A.1.1 Function: CART()

```
%-----------------------------------------------------%
% CART: classification trees                          %
% Author: Emanuel Bernabeu                            %
% Version: 1.1  Fall 2009                             %
% Only Classification Trees                           %
% Inputs:                                             %
% 1) Data: Learning Sample                            %
% 2) MinNodeData: minimun number of cases at a        %
%                 terminal node                       %
% 3) varargin: several inputs                         %
%              a) 'Priors', vector with priors        %
% Output:                                             %
% 1) Tree: sequence of trees.                         %
% Calls: None                                         %
% 1) growTmax()                                       %
% 2) CostComplexityPruning()                          %
% 3) CrossValidation                                  %
% Is called by: None                                  %
%-----------------------------------------------------%
function Tree = CART(Data,MinNodeData,V,varargin)
disp('Growing Tmax');
if isempty(varargin)
Tree = GrowTmax(Data,MinNodeData);
else
Tree = GrowTmax(Data,MinNodeData,varargin);
end
disp('Sequence of subtrees');
Tree = CostComplexityPruning(Tree);
disp('V-fold Cross-Validation');
Tree = CrossValidation(Tree,V,MinNodeData);
plotRcv(Tree);
```

A.1.2 Function: growTmax()

```
%-----------------------------------------------------%
% GrowTmax: grows a maximum sized tree                %
% Author: Emanuel Bernabeu                            %
% Version: 1.1  Fall 2009                             %
% Only Classification Trees                           %
% Inputs:                                             %
% 1) Data: Learning Sample                            %
% 2) MinNodeData: minimun number of cases at a        %
%                 terminal node                       %
% 3) varargin: several inputs                         %
%              a) 'Priors', vector with priors        %
% Output:                                             %
% 1) Tree: maximum sized tree Tmax                    %
% Calls: None                                         %
% Is called by:                                       %
% 1) CART()                                           %
%-----------------------------------------------------%
function Tree = GrowTmax(Data,MinNodeData,varargin)
% Default Options
Node(1).Target = Data(:,1);
Node(1).Total = numel(Node.Target);
Tree(1).Classes = unique(Data(:,1));                % Tree Classes
Tree(1).AttrNames = NaN;                            % Tree Classes Names
Tree(1).Rcv = [];
for i=1:numel(Tree(1).Classes)
Temp(i) = numel( find(Node(1).Target == Tree(1).Classes(i)) );
end
Tree(1).C = Temp;
Node(1).C = Temp;
Tree(1).Priors = Temp/Node(1).Total;                % Priors: n(Cj)/m
Tree(1).Alpha = 0;

% Reading options                                   %
if ~isempty(varargin)
for i=1:numel(varargin)
    if strcmp(upper(varargin(i)),'PRIORS')          % Setting Priors.
    Tree(1).Priors = varargin(i+1);
    end
end
end
[m NumAttr]=size(Data(:,2:end));                    % Size of learning sample
NodeIndex = 1;
Node(1).Type = 'M';
Node(1).Draw = 'M';
Node(1).Number = 1;
Node(1).Father = NaN;
Node(1).Brother = NaN;
Node(1).Child=[];
Node(1).Attributes = Data(:,2:end);
% Estimating: p(Cj,t) = prior(Cj) * n(Cj,t)/n(Cj)
for i=1:numel( Tree(1).Priors )
Temp(i) = Tree(1).Priors(i)*Node(1).C(i)/Tree(1).C(i);
end
```

```
Node(1).pCt = Temp;                                    % Estimating: p(Cj,t)
Node(1).p = sum(Temp);                                  % Estimating: p(t) = sum
of p(Cj,t) for all j.
Node(1).pC = Temp/Node(1).p;                            % Estimating: p(Cj|t) =
p(Cj,t)/p(t)
[temp iMax] = max(Node(1).pC);
Node(1).Class = Tree(1).Classes(iMax);
Node(1).r = 1 - max(Node(1).pC);                       % Estimating: r(t) = max(
p(Cj|t) ) for all j.
Node(1).R = Node(1).r * Node(1).p;
Node(1).Impurity = 1 - sum( (Node(1).pC).^2 ) ;      % Computing: i(t) = 1 -
sum of p(Cj,t)^2 for all j.
Node(1).Split = NaN;
Node(1).SplitAttr = NaN;
Node(1).SplitWPImpurity = NaN;
SplitNodes = [1];
NodeIndex = 1;
while ~isempty(SplitNodes)
for k=1:numel(SplitNodes)
iNode = SplitNodes(k);
WPImpurity=99;
for Attr=1:NumAttr
  S = unique( sort(Node(iNode).Attributes(:,Attr)) );
  SplitV = ( S(1:end-1) + S(2:end) )/2;                % Creating vector of
midpoints.
  for i=1:numel(SplitV)
    [WPImpurityTemp    ImpurityLeft    ImpurityRight    LeftNodeTemp    CLTemp
RightNodeTemp
CRTemp]=ImpurityGini(Node(iNode).Attributes(:,Attr),Node(iNode).Target,Tree(1
).Classes,SplitV(i),Tree(1).Priors,Tree(1).C);
    if ((WPImpurityTemp < WPImpurity) && (numel(LeftNodeTemp) >= MinNodeData)
&& (numel(RightNodeTemp) >= MinNodeData))
      WPImpurity = WPImpurityTemp;
      SplitValue = SplitV(i);
      SplitAttr = Attr;
    end
  end
end
if Node(iNode).Type ~= 'M'
    Node(iNode).Type = 'S';
end
Node(iNode).Split = SplitValue;
Node(iNode).SplitAttr = SplitAttr;
Node(iNode).SplitWPImpurity = WPImpurity;
Node(iNode).Child=[NodeIndex + 1, NodeIndex + 2];
[WPImpurity ImpurityLeft ImpurityRight LeftNode CL RightNode CR pCtL pL pCL
pCtR                                                                    pR
pCR]=ImpurityGini(Node(iNode).Attributes(:,SplitAttr),Node(iNode).Target,Tree
(1).Classes,SplitValue,Tree(1).Priors,Tree(1).C);
% Data LEFT NODE
NodeIndex = NodeIndex + 1;
Node(NodeIndex).Number = NodeIndex;
Node(NodeIndex).Father = iNode;
Node(NodeIndex).Draw = 'L';
Node(NodeIndex).Brother = NodeIndex + 1;
Node(NodeIndex).Target = Node(Node(NodeIndex).Father).Target(LeftNode,:);
```

```
Node(NodeIndex).Attributes                                                    =
Node(Node(NodeIndex).Father).Attributes(LeftNode,:);
Node(NodeIndex).C = CL;
Node(NodeIndex).Total = numel(Node(NodeIndex).Target);
Node(NodeIndex).pCt = pCtL;
Node(NodeIndex).p = pL;
Node(NodeIndex).pC = pCL;
[temp iMax] = max(Node(NodeIndex).pC);
Node(NodeIndex).Class = Tree(1).Classes(iMax);
Node(NodeIndex).r = 1 - max(Node(NodeIndex).pC);                              %
Estimating: r(t) = max( p(Cj|t) ) for all j.
Node(NodeIndex).R = Node(NodeIndex).r * Node(NodeIndex).p;
Node(NodeIndex).Impurity=ImpurityLeft;
Node(NodeIndex).Type = 'X';
if       (       (Node(NodeIndex).Total       <=       2*MinNodeData)       ||
(Node(NodeIndex).Impurity==0) )
  Node(NodeIndex).Type = 'T';
end
% Data RIGHT NODE
NodeIndex = NodeIndex + 1;
Node(NodeIndex).Number = NodeIndex;
Node(NodeIndex).Father = iNode;
Node(NodeIndex).Draw = 'R';
Node(NodeIndex).Brother = NodeIndex-1;
Node(NodeIndex).Target = Node(Node(NodeIndex).Father).Target(RightNode,:);
Node(NodeIndex).Attributes                                                    =
Node(Node(NodeIndex).Father).Attributes(RightNode,:);
Node(NodeIndex).C = CR;
Node(NodeIndex).pCt = pCtR;
Node(NodeIndex).p = pR;
Node(NodeIndex).pC = pCR;
[temp iMax] = max(Node(NodeIndex).pC);
Node(NodeIndex).Class = Tree(1).Classes(iMax);
Node(NodeIndex).Total = numel(Node(NodeIndex).Target);
Node(NodeIndex).r = 1 - max(Node(NodeIndex).pC);                              %
Estimating: r(t) = max( p(Cj|t) ) for all j.
Node(NodeIndex).R = Node(NodeIndex).r * Node(NodeIndex).p;
Node(NodeIndex).Impurity=ImpurityRight;
Node(NodeIndex).Type = 'X';
if       (       (Node(NodeIndex).Total       <=       2*MinNodeData)       ||
(Node(NodeIndex).Impurity==0) )
  Node(NodeIndex).Type = 'T';
end
end % main For
% Determine the nodes that need to be splitted.
SplitNodes = [];
for p=1:NodeIndex
    if Node(p).Type == 'X'
        SplitNodes = [SplitNodes p];
    end
end
end %While
Tree(1).Node = Node;
```

A.1.3 Function: CostComplexityPruning()

```
%-----------------------------------------------------%
% CostComplexityPruning: prunes Tmax                  %
% Author: Emanuel Bernabeu                            %
% Version: 1.0  Fall 2009                             %
% Only Classification Trees                           %
% Inputs:                                             %
% 1) Tree: Maximum sized Tree                         %
% Output:                                             %
% 1)Tree: Sequence of subtrees                        %
% Calls:                                              %
% 1) prune2()                                         %
% 2) PruneDescendants()                               %
% 3) NodeDescendants()                                %
% Is called by:                                       %
% 1) CART                                             %
%-----------------------------------------------------%

% NOTE: THIS IS ALL DONE FOR PRIORS Nj/N; FIXED
function Tree = CostComplexityPruning(Tree)
%disp('Pruning Tmax');
iTree = 2;                                          % Index for subtree
tol = 1e-12;                                              % Tolerance for
R(t)>=R(tL)+R(tR)
Tree(iTree).Node = Tree.Node;                        % Saving complete tree to
be pruned
Tree(iTree).Alpha = 0;                               % Tmax and T(2) have same
alpha.
Tree(iTree).Classes = Tree(1).Classes;
Tree(iTree).AttrNames = Tree(1).AttrNames;
Tree(iTree).C = Tree(1).C;
Tree(iTree).Priors = Tree(1).Priors;

% 1) First stage of pruning. Initially we need to prune nodes with
R(t)==R(tL)+R(tR)
Done=false;
while ~Done                                            % Initiating pruning
algorithm
Terminals=find([Tree(iTree).Node.Type]=='T');          % Index with terminal
nodes.
FatherTerminals=[];
for i=1:numel(Terminals)
NBrother=Tree(iTree).Node(Terminals(i)).Brother;
if (Tree(iTree).Node(NBrother).Type == 'T')          % Testing if brother of a
terminal node is also terminal
FatherTerminals=[FatherTerminals   Tree(iTree).Node(Terminals(i)).Father];   %
Saving index for father of pair of terminal nodes.
end
end
FatherTerminals=unique(FatherTerminals);
PruneNodes=[];
for i=1:numel(FatherTerminals)
Child = Tree(iTree).Node(FatherTerminals(i)).Child;
% Checking if: R(t)== R(tL)+R(tR)
```

```
if                abs(Tree(iTree).Node(FatherTerminals(i)).R                -
sum([Tree(iTree).Node(Child).R])) < tol
% The condition below did not work properly; that's why using a tolerance.
%if                 Tree(iTree).Node(FatherTerminals(i)).R                 ==
sum([Tree(iTree).Node(Child).R])
PruneNodes=[PruneNodes Child];                          % Index of nodes to be
pruned.
end
end
PruneNodes=sort(PruneNodes,'descend');
if isempty(PruneNodes)
    Done=true;
else
    Tree(iTree).Node=prune2(Tree(iTree).Node,PruneNodes);
end
end% While

%  2)  Second  stage  of  pruning.  Tree  has  been  reduced  by  trimming
R(t)==R(tL)+R(tR).
%    Now the weakest links are found using cost-compelixity.
Done=false;
while ~Done
IndexSplit=find([Tree(iTree).Node.Type]=='S');      % Index: splitting nodes.
gNN=[];                                                 % Cost-complexity
function.
for i=1:numel(IndexSplit)
NN=IndexSplit(i);                                   % Node Number.
Descendants=NodeDescendants(Tree(iTree).Node,NN);   % Find node descendents.
NodeTypes=[Tree(iTree).Node(Descendants).Type];
DescendantsT=Descendants(find(NodeTypes=='T'));      % Find descendents which
are terminal nodes
RNN=Tree(iTree).Node(NN).R;                          % Resubstitution estimate
R(t) of node t
RDescendantsT=sum([Tree(iTree).Node(DescendantsT).R]);   %   Sum   of   R(td),
resubstitution estimate of terminal nodes
g= (RNN - RDescendantsT)/(numel(DescendantsT)-1);   % Computing function g.
gNN(i,:)=[NN g];
end
[MinValue,iMinValue] = min(gNN(:,2));                 % Finding weakest link:
Min(g).
alpha(iTree-1)=MinValue;
Tree(iTree+1).Node=PruneDescendants(Tree(iTree).Node,gNN(iMinValue,1));
Tree(iTree+1).Alpha = MinValue;
Tree(iTree+1).Classes = Tree(1).Classes;
Tree(iTree+1).AttrNames = Tree(1).AttrNames;
Tree(iTree+1).C = Tree(1).C;
Tree(iTree+1).Priors = Tree(1).Priors;

iTree=iTree+1;
if numel(Tree(iTree).Node)==3
    Done=true;
end
end
```

A.1.4 Function: prune2()

```
function NewNode=prune2(Node,PruneNodes)
if ~isempty(PruneNodes)
NewNode=Node;                                          % New set of Nodes.
NewTerminals=unique([Node(PruneNodes).Father]);       % Find fathers of nodes
to be pruned
for i=1:numel(NewTerminals)
NewNode(NewTerminals(i)).Type='T';                        % Change Type for new
terminal nodes
NewNode(NewTerminals(i)).Child=[];                     % Remove Child.
end
NewNode(PruneNodes)=[];                                % Prune Nodes.
% Tree now has different internal and external node indeces.
internal=[1:1:numel(NewNode)];
external=[NewNode(internal).Number];
for i=2:numel(internal)
NewNode(i).Number = i;
NewNode(i).Father = internal(find(NewNode(i).Father==external));
NewNode(i).Brother = internal(find(NewNode(i).Brother==external));
Child = NewNode(i).Child;
if ~isempty(Child)
ChildL = internal(find(Child(1)==external));
ChildR = internal(find(Child(2)==external));
NewNode(i).Child=[ChildL, ChildR];
end
end

end
```

A.1.5 Function: PruneDescendants()

```
function NewNode=PruneDescendants(Node,PruneNode)
Descendants=NodeDescendants(Node,PruneNode);
if ~isempty(Descendants)
NewNode=Node;                                   % New set of Nodes.
NewNode(PruneNode).Type='T';                    % New Terminal Node.
NewNode(PruneNode).Child=[];                    % New Terminal Node.
NewNode(PruneNode).Split = [];
NewNode(PruneNode).SplitAttr = [];
NewNode(Descendants)=[];                        % Prune Descendants.
% Tree now has different internal and external node indeces.
internal=[1:1:numel(NewNode)];
external=[NewNode(internal).Number];
for i=2:numel(internal)
NewNode(i).Number = i;
NewNode(i).Father = internal(find(NewNode(i).Father==external));
NewNode(i).Brother = internal(find(NewNode(i).Brother==external));
Child = NewNode(i).Child;
if ~isempty(Child)
ChildL = internal(find(Child(1)==external));
ChildR = internal(find(Child(2)==external));
NewNode(i).Child=[ChildL, ChildR];
end
end

end
```

A.1.6 Function: NodeDescendants()

```
function Descendants=NodeDescendants(Node,NodeNumber)
Descendants=[];
if Node(NodeNumber).Type~='T'
Child=[NodeNumber];
Done=false;
while ~Done
NewDescendants=[];
for i=1:numel(Child)
ChildTemp = Node(Child(i)).Child;
NewDescendants=[NewDescendants ChildTemp];
end
Descendants=[Descendants NewDescendants];
%disp(NewDescendants)
IndexTerminals=[];
for i=1:numel(NewDescendants)
%disp(NewDescendants(i))
if Node(NewDescendants(i)).Type=='T'
    IndexTerminals=[IndexTerminals i];
end
end
NewDescendants(IndexTerminals)=[];
Child=NewDescendants;
if isempty(Child)
    Done=true;
end
end%While
end%If
```

A.1.7 Function: CrossValidation()

```
function Tree = CrossValidation(Tree,V,MinNodeData)
%-----------------------------------------------------%
% V-Fold Cross Validation:                            %
% Author: Emanuel Bernabeu                            %
% Version: 1.0  Fall 2009                             %
% Only Classification Trees                           %
% Inputs:                                             %
% 1) Tree: Tmax and subsequence of trees.             %
% 2) V: total number of folds.                        %
% 3) MinNodeData: minimum number of cases in a        %
%                 terminal node.                      %
% Output:                                             %
% 1)Tree with misclassification Rcv estimates         %
% Calls:                                              %
% 1) ScoreCV: returns misclassification rates of a    %
%             Tree                                    %
% Is called by:                                       %
% 1) CART                                             %
%-----------------------------------------------------%
m=Tree(1).Node(1).Total;
for i=3:numel(Tree)-1
Alpha(i-2)  =  (Tree(i).Alpha*Tree(i+1).Alpha)^(1/2);  %  Geometric  distance
between alphas.
end
Groups = ceil(V * randperm(m)' / m);                  % Dividing data into V
groups
Data = [Tree(1).Node(1).Target Tree(1).Node(1).Attributes]; % Learning Sample
L
for v=1:V
iSample = find(Groups == v);
% Growing and pruning V trees.
NewL = Data;
NewL(iSample,:)=[];                                    % New Learning sample
L(v)
CV(v).Tree = GrowTmax(NewL,MinNodeData);             % Growing Tmax with L(v)
CV(v).Tree = CostComplexityPruning(CV(v).Tree);      % Pruning with L(v)
% Estimating error rates.
TestSample = Data(iSample,:);                          % Test sample for
learning sample L(v)
CV(v).Tree  =  ScoreCV(CV(v).Tree,TestSample);          % Misclassification
errors.
end
Tree(end).Rcv = Tree(end).Node(1).R;                 % T(root) has Rcv = R
for k=2:2                                              % Misclassification rate
for Tmax.
 nCiCj=[0 0];
 for v=1:V
 nCiCj = nCiCj + CV(v).Tree(k).Rcv;
 end
 Tree(k).Rcv = sum( (nCiCj./Tree(k).C) .* Tree(k).Priors );
end
for k=3:numel(Tree)-1                                % Loop through subtrees
 nCiCj=[0 0];
```

```
 for v=1:V
 iCVTree = max( find([CV(v).Tree.Alpha] <= Alpha(k-2)) );% Finding equivalent
tree
 nCiCj = nCiCj + CV(v).Tree(iCVTree).Rcv;                % Adding equivalent
subtrees error rates.
 end
 Tree(k).Rcv    =    sum(    (nCiCj./Tree(k).C)    .*    Tree(k).Priors    );    %
Misclassification rate for subtree Tk
end
```

A.1.8 Function: ScoreCV()

```
function Tree = ScoreCV(Tree,TestSample)
[m NumAttr]=size(TestSample);
for k=1:numel(Tree)
Tree(k).Rcv = zeros(1,numel(Tree(1).Classes));
for i=1:m
iNode = 1;
Done=false;
% Finding a path to a terminal node.
while ~Done
iAttr = Tree(k).Node(iNode).SplitAttr + 1;          % Extra column of target.
if TestSample(i,iAttr) <= Tree(k).Node(iNode).Split
iNode = Tree(k).Node(iNode).Child(1);               % Left Child.
else
iNode = Tree(k).Node(iNode).Child(2);               % Right Child.
end
if Tree(k).Node(iNode).Type == 'T'
Done = true;
end
end % While, path leads to terminal node iNode.
% Check if xi is misclassified.
if ( Tree(k).Node(iNode).Class ~= TestSample(i,1) )
index = find(Tree(k).Classes == TestSample(i,1));
Tree(k).Rcv(index) = Tree(k).Rcv(index) + 1;
end
end % for: sample

end
```

A.1.9 Function: ImpurityGini()

```
function [WPImpurity ImpurityLeft ImpurityRight LeftNode CL RightNode CR pCtL
pL pCL pCtR pR pCR] = ImpurityGini(Data,Target,Classes,Split,Priors,nCj)
%-----------------------------------------------------%
% ImpurityGini: returns Gini impurity function      %
% Author: Emanuel Bernabeu                          %
% Version: 1.0  Fall 2009                           %
% Only Classification Trees                         %
% Inputs:                                           %
% 1) Data: an attribute, i.e, column of L           %
% 2) Target: target.                                %
% 3) Classes: vector with unique classes            %
% 4) Split: splitting value                         %
% 5) Priors                                         %
% 6) nCj: Total number of class Cj                  %
% Output:                                           %
% 1)WPImpurity: Weighted Impurity function          %
% 2)ImpurityLeft: Gini impurity of left child       %
% Calls: NONE                                       %
% Is called by:                                     %
% 1) GrowTmax                                       %
%-----------------------------------------------------%
N=numel(Data);
LeftNode = find( Data <= Split );
NL = numel(LeftNode);
RightNode = find( Data > Split);
NR = numel(RightNode);
for i=1:numel(Classes)
CL(i) = numel( find(Target(LeftNode) == Classes(i)) );   %n(Cj)
CR(i) = numel( find(Target(RightNode) == Classes(i)) );
end
pCtL=(Priors.*CL)./nCj;                                          % p(Cj,t) =
pi(Cj)*n(Cj,t)/n(Cj)
pCtR=(Priors.*CR)./nCj;
pL = sum(pCtL);                                           % Estimating: p(t) =
sum of p(Cj,t) for all j.
pR = sum(pCtR);
pCL = pCtL/pL;                                            % Estimating: r(t) =
max( p(Cj|t) ) for all j.
pCR = pCtR/pR;
ImpurityLeft = 1 - sum( (pCL).^2 ) ;                     % Computing: i(t) = 1
- sum of p(Cj,t)^2 for all j.
ImpurityRight = 1 - sum( (pCR).^2 ) ;                    % Computing: i(t) = 1
- sum of p(Cj,t)^2 for all j.
WPImpurity   =   (NL/N)*ImpurityLeft+(NR/N)*ImpurityRight;       %   Weighted
Probability Impurity
```

A.1.10 Function: plotSimpleTree()

```
function plotSimpleTree(Tree)
defaultcolors={'b','r','y','m','c','k','w'};
fig = figure;
AxesHandler = axes('Position',[0 0 1 1]);
pan on
plot(0,0,'Marker','s','MarkerFaceColor','green')            % Root Node
dcm_obj = datacursormode(fig);
set(dcm_obj,'UpdateFcn',@CursorNodeInfo);
hold on;
xoffsetL = -1;
xoffsetR = 1;
yoffset = -1;
TerminalNodes=find('T'==[Tree.Node.Type]);
SplitNodes = find('S'==[Tree.Node.Type]);
SplitNodes = [1 SplitNodes];                                  % Root Node type
is actually M
TreeWidth = numel(TerminalNodes);
TreeHeight = 0;
for i=1:numel(TerminalNodes)
HeightTemp=numel(FindPath(Tree.Node,TerminalNodes(i)));
if HeightTemp > TreeHeight
TreeHeight=HeightTemp;
end
end
xGap=[TreeHeight:-1:1];
xGap=2.^xGap;

NodeTemp = Tree.Node;
x=0; y=0;
NodeTemp(1).x=x;
NodeTemp(1).y=y;

for i=2:numel(Tree.Node)
PathSeq=FindPath(Tree.Node,i);
NY = numel(PathSeq);
NL = numel(find('L'==PathSeq));
NR = numel(find('R'==PathSeq));
Father=NodeTemp(i).Father;
x = NodeTemp(Father).x;
%xOffSet=x+xoffsetL*NL*xGap(NY)+xoffsetR*NR*xGap(NY);
iNL=1;
iNR=0;
if Tree.Node(i).Draw=='R'
iNL=0;
iNR=1;
end
xOffSet=x+iNL*xoffsetL*xGap(NY)+xoffsetR*iNR*xGap(NY);
yOffSet = y+yoffset*NY;
%text(xOffSet,yOffSet,[num2str(Node(i).Number)],'Color','k');
%plot(xOffSet,yOffSet,'Marker','s','MarkerFaceColor','green')
NodeTemp(i).x=xOffSet;
NodeTemp(i).y=yOffSet;
end
```

```
for i=1:numel(SplitNodes)
Child=NodeTemp(SplitNodes(i)).Child;
line([NodeTemp(SplitNodes(i)).x
NodeTemp(Child(1)).x],[NodeTemp(SplitNodes(i)).y NodeTemp(Child(1)).y]);
line([NodeTemp(SplitNodes(i)).x
NodeTemp(Child(2)).x],[NodeTemp(SplitNodes(i)).y NodeTemp(Child(2)).y]);
end

for i=1:numel(NodeTemp)
xOffSet=NodeTemp(i).x;
yOffSet=NodeTemp(i).y;
if NodeTemp(i).Type == 'T'
iColor = find(NodeTemp(i).Class == Tree.Classes);
plot(xOffSet,yOffSet,'Marker','s','MarkerFaceColor',defaultcolors{iColor})
else
plot(xOffSet,yOffSet,'Marker','s','MarkerFaceColor','green')
end
end
axis off
Tree.Node = NodeTemp;
assignin('base','TreePlotted',Tree);
yborderoffset = TreeHeight*0.05;
ylim([-TreeHeight-yborderoffset, yborderoffset]);
xvalues =[NodeTemp.x];
xborderoffset = (max(xvalues)-min(xvalues))*0.05;
xlim([min(xvalues)-xborderoffset, max(xvalues)+xborderoffset]);
```

A.1.11 Function: CursorNodeInfo()

```
function txt = CursorNodeInfo(empt,event_obj)
pos = get(event_obj,'Position');
Tree = evalin('base','TreePlotted');
iNode = find([Tree.Node.x]==pos(1));
NN = Tree.Node(iNode).Number;
Split = Tree.Node(iNode).Split;
SplitAttr = Tree.Node(iNode).SplitAttr;
Class = Tree.Node(iNode).Class;
Classes = Tree.Classes;
C = Tree.Node(iNode).C;
Total = Tree.Node(iNode).Total;
%pepe=Node(1).R;
txt = {['Node: ',num2str(NN)],...
       ['Total: ',num2str(Total)]};
if Tree.Node(iNode).Type ~= 'T'
txt = [txt, {['Split: a',num2str(SplitAttr),' <= ',num2str(Split)]}];
else
txt = [txt, {['Classification: ',num2str(Class)]}];
end
%txt = [txt, {['Class    Cases       %']}];
txt = [txt, {['Class    Cases']}];
for i=1:numel(Classes)
pct = 100*C(i)/Total;
pct = round(pct*10)/10;
%txt  =  [txt,  {[num2str(Classes(i)),'                        ',num2str(C(i)),'
',num2str(pct)]}];
txt = [txt, {[num2str(Classes(i)),'            ',num2str(C(i))]}];
end
```

A.1.12 Function: plotRcv()

```
function plotRcv(Tree)
for k=2:numel(Tree)
NumTerminal(k-1) = numel(find([Tree(k).Node.Type]=='T'));
end
fig = figure;
plot(NumTerminal,[Tree.Rcv],'Marker','s','MarkerFaceColor','green');
dcm_obj = datacursormode(fig);
set(dcm_obj,'UpdateFcn',@CursorRcvInfo);
xlabel('Number of Terminal Nodes');
ylabel('Misclassification Rate');
xlim([0 max(NumTerminal)+2]);
```

A.1.13 Function: CursorRcvInfo()

```
function txt = CursorRcvInfo(empt,event_obj)
pos = get(event_obj,'Position');
Tree = evalin('base','Tree');
for v=2:numel(Tree)
NTerminal = numel(find([Tree(v).Node.Type]=='T'));
if NTerminal == pos(1)
iTree = v;
end
end
txt = {['Sub-tree: ',num2str(iTree)],...
       ['Terminal Nodes: ',num2str(pos(1))]...
       ['Misclassification rate: ',num2str(pos(2))]};
```

A.1.14 Function: FindPath()

```
function PathSeq = FindPath(Node,NodeNumber)
Done=false;
List=[];
ListTemp=[NodeNumber];
NN=NodeNumber;
PathSeq=[];
while ~Done
PathSeq=[PathSeq Node(NN).Draw];
NN=Node(NN).Father;
if ( NN==1 )
   Done=true;
end
end%while
```

A.2 Experiment 1

To allow the reader to replicate the results obtained in Experiment 1, the learning sample is attached.

```
% Target a1         a2
0      0.72552     0.1047
0      0.42127     0.045074
0      0.54034     0.5864
0      0.015772    0.82822
0      0.35837     0.71879
0      0.66612     0.025194
0      0.08499     0.20533
0      0.10487     0.48262
0      0.15601     0.25504
0      0.52212     0.61266
0      0.94108     0.52874
0      0.34455     0.020777
0      0.15855     0.62804
0      0.15078     0.83022
0      0.94828     0.18136
0      0.081292    0.40914
0      0.21431     0.28595
0      0.25121     0.31939
0      0.77703     0.61343
0      0.86048     0.63202
0      0.95476     0.4211
0      0.36802     0.23115
0      0.32463     0.19867
0      0.38194     0.37315
0      0.10465     0.060955
0      0.085324    0.60506
0      0.99334     0.82133
0      0.29709     0.19021
0      0.95928     0.47502
0      0.37442     0.82286
0      0.57705     0.82412
0      0.23723     0.82958
0      0.97511     0.2583
0      0.39666     0.027045
0      0.13994     0.8961
0      0.37622     0.050644
0      0.44293     0.1387
0      0.32386     0.1284
0      0.30026     0.927
0      0.19607     0.48873
0      0.6147      0.74124
0      0.62947     0.33354
0      0.61105     0.34006
0      0.92698     0.79357
0      0.81011     0.19546
0      0.36198     0.33108
0      0.66719     0.49505
0      0.86122     0.76805
```

0	0.56271	0.0012935
0	0.20554	0.54306
0	1.3633	1.4951
0	1.1384	1.8542
0	1.8936	1.7145
0	1.2736	1.0596
0	1.0154	1.7085
0	1.3566	0.84687
0	1.497	0.93957
0	1.8333	1.7465
0	1.3083	1.1029
0	0.89861	1.6875
0	1.4945	1.8058
0	1.0087	1.588
0	1.2071	0.9072
0	1.0967	0.85844
0	1.574	1.9529
0	1.7023	1.5882
0	1.9311	1.3435
0	1.974	1.4627
0	0.91009	0.99213
0	1.3461	1.932
0	1.9176	1.722
0	1.4335	1.8149
0	1.2084	1.3436
0	1.8256	1.6332
0	1.6892	0.89427
0	1.0927	1.0214
0	1.7428	1.8041
0	0.94469	1.9645
0	1.1118	1.1997
0	1.6189	0.81116
0	1.4632	1.6643
0	0.91974	1.6895
0	1.3209	1.8212
0	1.1545	1.4564
0	1.5235	1.4563
0	1.7767	1.0518
0	1.5937	1.5811
0	0.95194	1.8781
0	1.665	1.9858
0	1.6539	1.5984
0	0.93745	1.1258
0	1.977	1.1817
0	0.86379	1.1923
0	1.1277	1.3972
0	1.7126	0.80584
0	1.0849	1.8974
0	1.1784	0.83872
0	1.7866	0.99823
0	1.9731	1.4506
0	1.9892	1.2092
1	0.4144	1.4805
1	0.83912	1.8512
1	0.48616	1.81
1	0.35269	1.2312
1	0.78417	0.87307

1	0.77188	1.8074
1	0.591	1.5343
1	0.6937	1.2892
1	0.38978	1.2679
1	0.44738	0.90417
1	0.24724	1.2873
1	0.74866	0.95765
1	0.27086	1.1117
1	0.10228	1.6544
1	0.55018	1.8116
1	0.38151	0.83437
1	0.90097	1.3427
1	0.42215	1.6044
1	0.83556	1.9769
1	0.17948	1.2706
1	0.056478	1.2166
1	0.30116	1.3541
1	0.80014	1.5212
1	0.62653	1.1673
1	0.10733	0.83848
1	0.41744	1.0868
1	0.22211	1.3639
1	0.71996	1.1954
1	0.038795	1.9739
1	0.14272	0.99259
1	0.3582	1.4722
1	0.84763	1.0146
1	0.2108	1.379
1	0.31096	1.8582
1	0.2433	1.1954
1	0.39078	1.6725
1	0.83839	1.5036
1	0.059772	1.7787
1	0.28113	0.91612
1	0.68866	0.82518
1	0.55742	0.89975
1	0.93793	1.7548
1	0.94258	1.6763
1	0.50572	1.4438
1	0.92399	1.9231
1	0.79728	1.8541
1	0.90897	0.9082
1	0.45187	1.5005
1	0.58182	1.4255
1	0.37293	1.2625
1	1.8545	0.56885
1	1.8803	0.26586
1	1.7588	0.30396
1	1.0767	0.27998
1	0.95946	0.010891
1	1.7036	0.41838
1	1.3724	0.44689
1	1.3308	0.67428
1	1.6566	0.011313
1	1.0152	0.69396
1	1.603	0.21077
1	1.4297	0.34469

1	1.7397	0.10051
1	0.90927	0.14537
1	1.7162	0.62557
1	1.0771	0.9331
1	0.93321	0.488
1	1.4767	0.42682
1	1.6432	0.13843
1	1.5877	0.88145
1	0.86852	0.26964
1	1.5379	0.55229
1	1.7343	0.96169
1	1.9447	0.70106
1	0.95761	0.14326
1	1.8914	0.56035
1	1.6067	0.66773
1	1.3604	0.9178
1	0.94194	0.17991
1	1.5819	0.65491
1	1.8489	0.97398
1	1.5357	0.58364
1	0.91418	0.72762
1	1.2946	0.14786
1	1.8153	0.53798
1	1.7998	0.73257
1	0.90667	0.33031
1	1.5432	0.52991
1	1.9912	0.36333
1	1.5399	0.76553
1	0.8637	0.99938
1	0.87412	0.48129
1	1.2038	0.24393
1	1.539	0.10856
1	1.3011	0.92428
1	1.7223	0.54626
1	1.556	0.03457
1	1.9166	0.26992
1	1.2201	0.88867
1	1.4001	0.8361

Appendix B

B.1 Static Index

The list assembled with cases that had a significant impact in the system is shown in Table B-1.

Table B-1. Set of cases studied with the Dynamic Index

Case	Type	Name	Name	ck	Sec	Line	Dist	ISGA
107	F	ROUND MT	TABLE MT	1	2	1084	0.99	7.7347
107	HF	TABLE MT	VACA-DIX	1	0	1090		
107	HF	TABLE MT	TESLA	1	0	1093		
113	F	ROUND MT	TABLE MT	2	2	1087	0.99	7.7348
113	HF	TABLE MT	VACA-DIX	1	0	1090		
113	HF	TABLE MT	TESLA	1	0	1093		
115	F	TABLE MT	VACA-DIX	1	2	1090	0.01	7.7235
115	HF	ROUND MT	TABLE MT	1	0	1084		
115	HF	TABLE MT	TESLA	1	0	1093		
116	F	TABLE MT	VACA-DIX	1	2	1090	0.01	7.7237
116	HF	ROUND MT	TABLE MT	2	0	1087		
116	HF	TABLE MT	TESLA	1	0	1093		
118	F	TABLE MT	TESLA	1	2	1093	0.01	7.7246
118	HF	ROUND MT	TABLE MT	1	0	1084		
118	HF	TABLE MT	VACA-DIX	1	0	1090		
119	F	TABLE MT	TESLA	1	2	1093	0.01	7.7248
119	HF	ROUND MT	TABLE MT	2	0	1087		
119	HF	TABLE MT	VACA-DIX	1	0	1090		
120	F	TABLE MT	TESLA	1	2	1093	0.99	8.0942
120	HF	VACA-DIX	TESLA	1	0	1101		
120	HF	TRACY	TESLA	1	0	1103		
121	F	TABLE MT	TESLA	1	2	1093	0.99	8.1836
121	HF	VACA-DIX	TESLA	1	0	1101		
121	HF	TESLA	METCALF	1	0	1105		
122	F	TABLE MT	TESLA	1	2	1093	0.99	8.2523
122	HF	VACA-DIX	TESLA	1	0	1101		
122	HF	TESLA	LOSBANOS	1	0	1106		
123	F	TABLE MT	TESLA	1	2	1093	0.99	8.8318
123	HF	TRACY	TESLA	1	0	1103		

123	HF	TESLA	METCALF	1	0	1105		
124	F	TABLE MT	TESLA	1	2	1093	0.99	8.9646
124	HF	TRACY	TESLA	1	0	1103		
124	HF	TESLA	LOSBANOS	1	0	1106		
127	F	VACA-DIX	TESLA	1	2	1101	0.99	8.1008
127	HF	TABLE MT	TESLA	1	0	1093		
127	HF	TRACY	TESLA	1	0	1103		
128	F	VACA-DIX	TESLA	1	2	1101	0.99	8.1892
128	HF	TABLE MT	TESLA	1	0	1093		
128	HF	TESLA	METCALF	1	0	1105		
129	F	VACA-DIX	TESLA	1	2	1101	0.99	8.2585
129	HF	TABLE MT	TESLA	1	0	1093		
129	HF	TESLA	LOSBANOS	1	0	1106		
134	F	TRACY	TESLA	1	1	1103	0.99	8.1073
134	HF	TABLE MT	TESLA	1	0	1093		
134	HF	VACA-DIX	TESLA	1	0	1101		
135	F	TRACY	TESLA	1	1	1103	0.99	8.8394
135	HF	TABLE MT	TESLA	1	0	1093		
135	HF	TESLA	METCALF	1	0	1105		
136	F	TRACY	TESLA	1	1	1103	0.99	8.9719
136	HF	TABLE MT	TESLA	1	0	1093		
136	HF	TESLA	LOSBANOS	1	0	1106		
151	F	TESLA	METCALF	1	1	1105	0.01	8.194
151	HF	TABLE MT	TESLA	1	0	1093		
151	HF	VACA-DIX	TESLA	1	0	1101		
152	F	TESLA	METCALF	1	1	1105	0.01	8.8384
152	HF	TABLE MT	TESLA	1	0	1093		
152	HF	TRACY	TESLA	1	0	1103		
157	F	TESLA	LOSBANOS	1	1	1106	0.01	8.2605
157	HF	TABLE MT	TESLA	1	0	1093		
157	HF	VACA-DIX	TESLA	1	0	1101		
158	F	TESLA	LOSBANOS	1	1	1106	0.01	8.9703
158	HF	TABLE MT	TESLA	1	0	1093		
158	HF	TRACY	TESLA	1	0	1103		
237	F	GATES	DIABLO	1	1	1115	0.99	4316.469
237	HF	DIABLO	MIDWAY	2	0	1118		
237	HF	DIABLO	MIDWAY	3	0	1119		
256	F	DIABLO	MIDWAY	2	1	1118	0.01	4315.375
256	HF	GATES	DIABLO	1	0	1115		
256	HF	DIABLO	MIDWAY	3	0	1119		
269	F	DIABLO	MIDWAY	2	1	1118	0.99	9.7647
269	HF	MIDWAY	VINCENT	1	0	3857		

269	HF	MIDWAY	VINCENT	2	0	3860		
272	F	DIABLO	MIDWAY	3	1	1119	0.01	4315.688
272	HF	GATES	DIABLO	1	0	1115		
272	HF	DIABLO	MIDWAY	2	0	1118		
285	F	DIABLO	MIDWAY	3	1	1119	0.99	9.7257
285	HF	MIDWAY	VINCENT	1	0	3857		
285	HF	MIDWAY	VINCENT	2	0	3860		
298	F	MIDWAY	VINCENT	1	2	3857	0.01	9.7549
298	HF	DIABLO	MIDWAY	2	0	1118		
298	HF	MIDWAY	VINCENT	2	0	3860		
300	F	MIDWAY	VINCENT	1	2	3857	0.01	9.7142
300	HF	DIABLO	MIDWAY	3	0	1119		
300	HF	MIDWAY	VINCENT	2	0	3860		
302	F	MIDWAY	VINCENT	1	2	3857	0.01	2858.255
302	HF	MIDWAY	VINCENT	2	0	3860		
302	HF	MIDWAY	VINCENT	3	0	3863		
304	F	MIDWAY	VINCENT	1	2	3857	0.99	9.6782
304	HF	LUGO	VINCENT	1	0	3442		
304	HF	MIDWAY	VINCENT	2	0	3860		
306	F	MIDWAY	VINCENT	1	2	3857	0.99	9.6782
306	HF	LUGO	VINCENT	2	0	3443		
306	HF	MIDWAY	VINCENT	2	0	3860		
308	F	MIDWAY	VINCENT	1	2	3857	0.99	6389.492
308	HF	MIDWAY	VINCENT	2	0	3860		
308	HF	MIDWAY	VINCENT	3	0	3863		
319	F	MIDWAY	VINCENT	2	2	3860	0.01	9.7545
319	HF	DIABLO	MIDWAY	2	0	1118		
319	HF	MIDWAY	VINCENT	1	0	3857		
321	F	MIDWAY	VINCENT	2	2	3860	0.01	9.7139
321	HF	DIABLO	MIDWAY	3	0	1119		
321	HF	MIDWAY	VINCENT	1	0	3857		
323	F	MIDWAY	VINCENT	2	2	3860	0.01	2773.469
323	HF	MIDWAY	VINCENT	1	0	3857		
323	HF	MIDWAY	VINCENT	3	0	3863		
325	F	MIDWAY	VINCENT	2	2	3860	0.99	9.678
325	HF	LUGO	VINCENT	1	0	3442		
325	HF	MIDWAY	VINCENT	1	0	3857		
327	F	MIDWAY	VINCENT	2	2	3860	0.99	9.6781
327	HF	LUGO	VINCENT	2	0	3443		
327	HF	MIDWAY	VINCENT	1	0	3857		
329	F	MIDWAY	VINCENT	2	2	3860	0.99	6407.404
329	HF	MIDWAY	VINCENT	1	0	3857		

329	HF	MIDWAY	VINCENT	3	0	3863		
344	F	MIDWAY	VINCENT	3	3	3863	0.01	734.7981
344	HF	MIDWAY	VINCENT	1	0	3857		
344	HF	MIDWAY	VINCENT	2	0	3860		
350	F	MIDWAY	VINCENT	3	3	3863	0.99	6721.188
350	HF	MIDWAY	VINCENT	1	0	3857		
350	HF	MIDWAY	VINCENT	2	0	3860		
456	F	LUGO	VINCENT	1	1	3442	0.99	9.6902
456	HF	MIDWAY	VINCENT	1	0	3857		
456	HF	MIDWAY	VINCENT	2	0	3860		
483	F	LUGO	VINCENT	2	1	3443	0.99	9.6902
483	HF	MIDWAY	VINCENT	1	0	3857		
483	HF	MIDWAY	VINCENT	2	0	3860		

B.2 Decision Tree: Heavy Winter Model

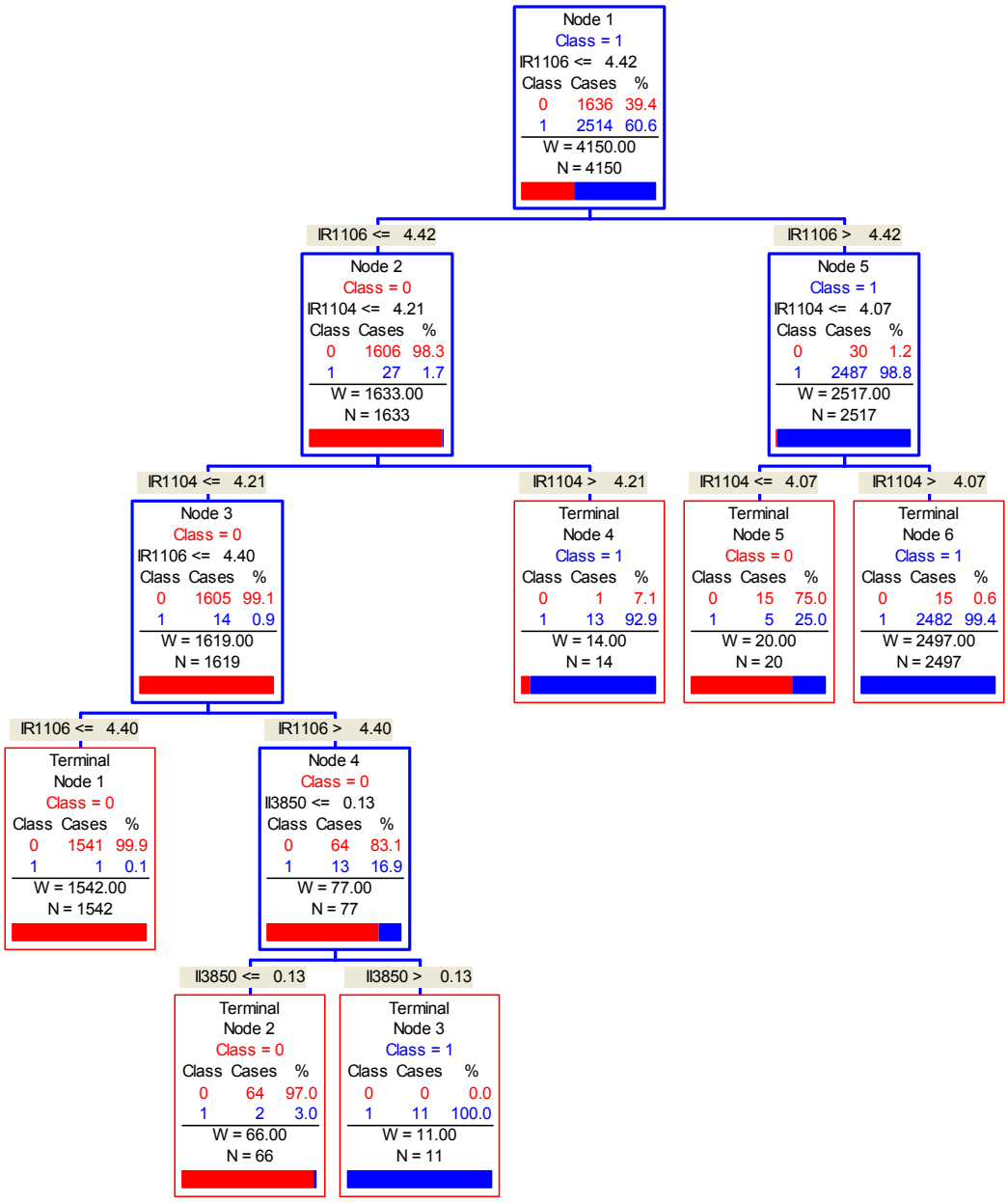

Figure B-1. Selected Decision Tree. Misclassification rate = 0.99%

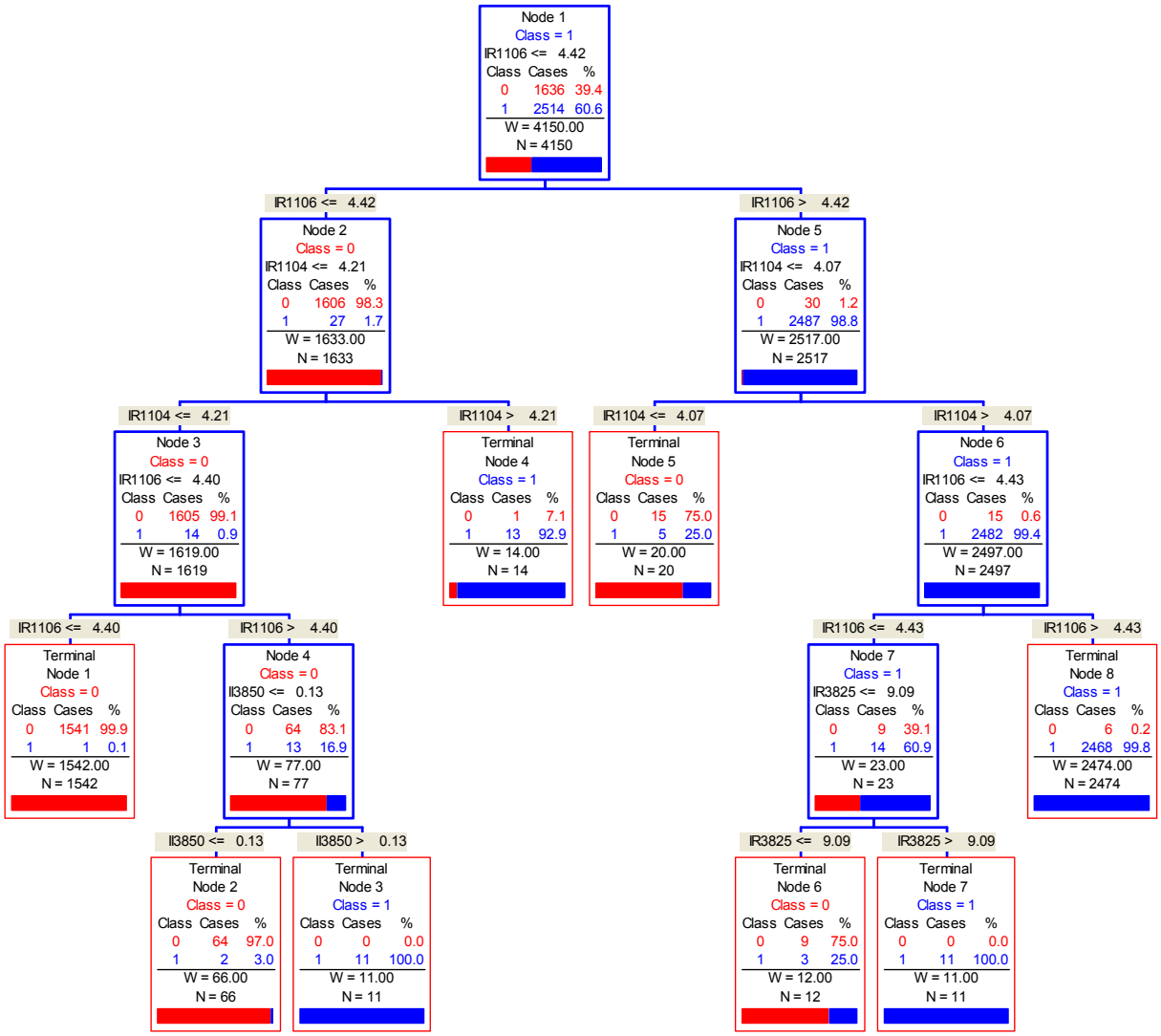

Figure B-2. Tree with minimum misclassification rate: 0.89%.

B.2.1 Partition Sequence

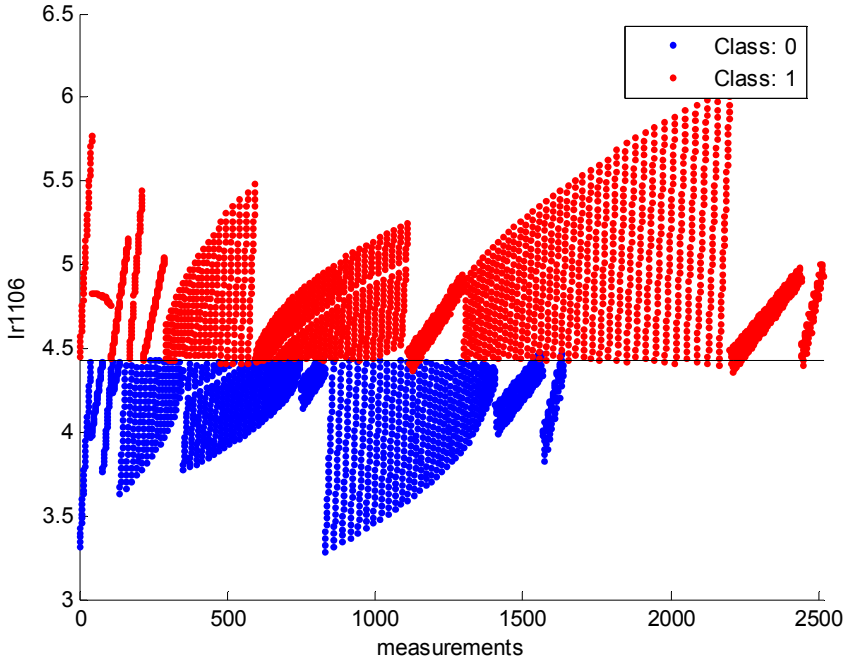

Figure B-3. Split at node 1.

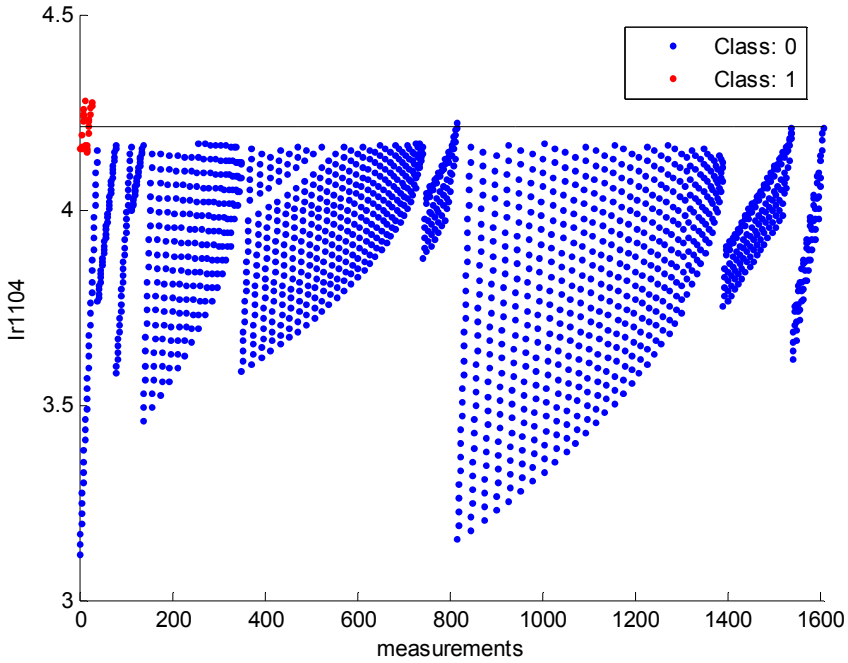

Figure B-4. Split at node 2.

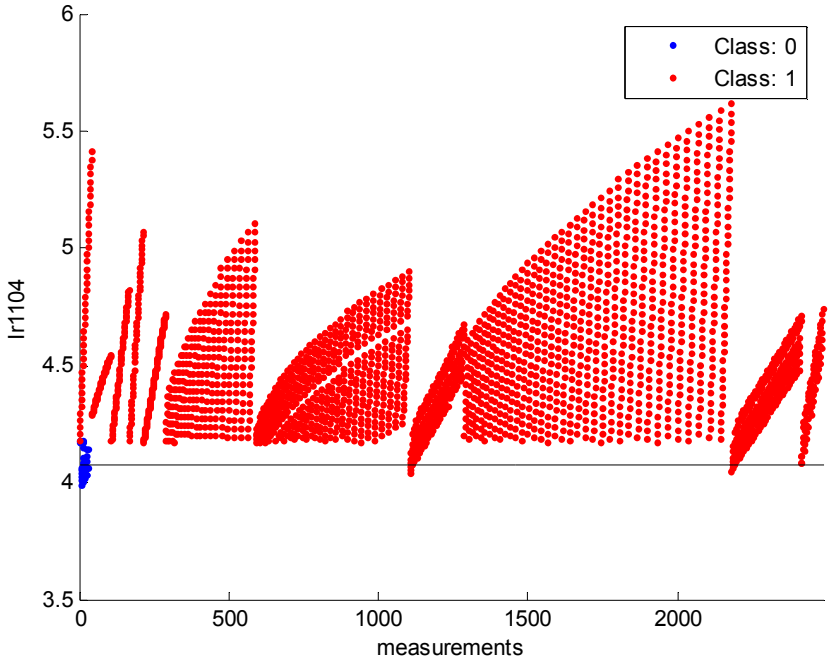

Figure B-5. Split at node 3.

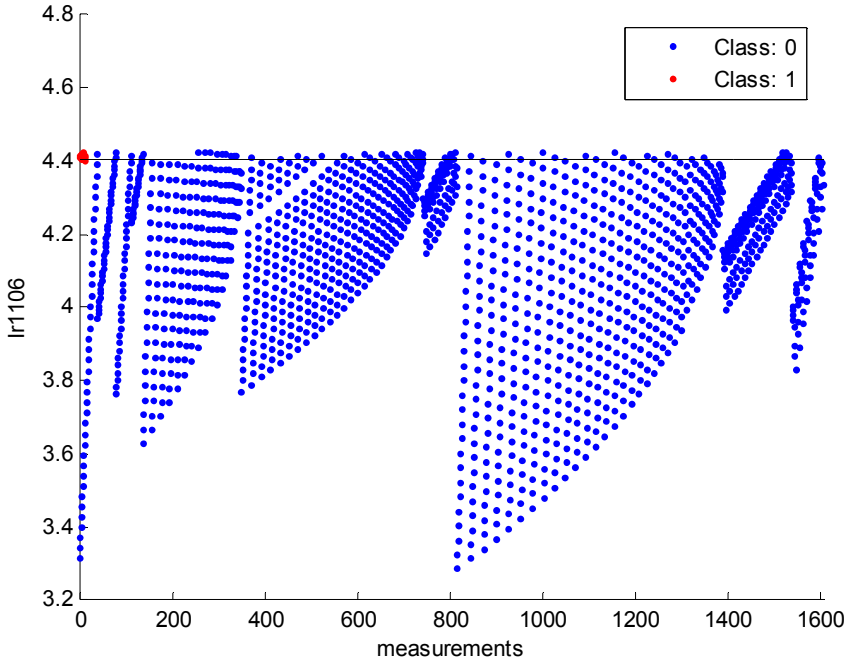

Figure B-6. Split at node 4.

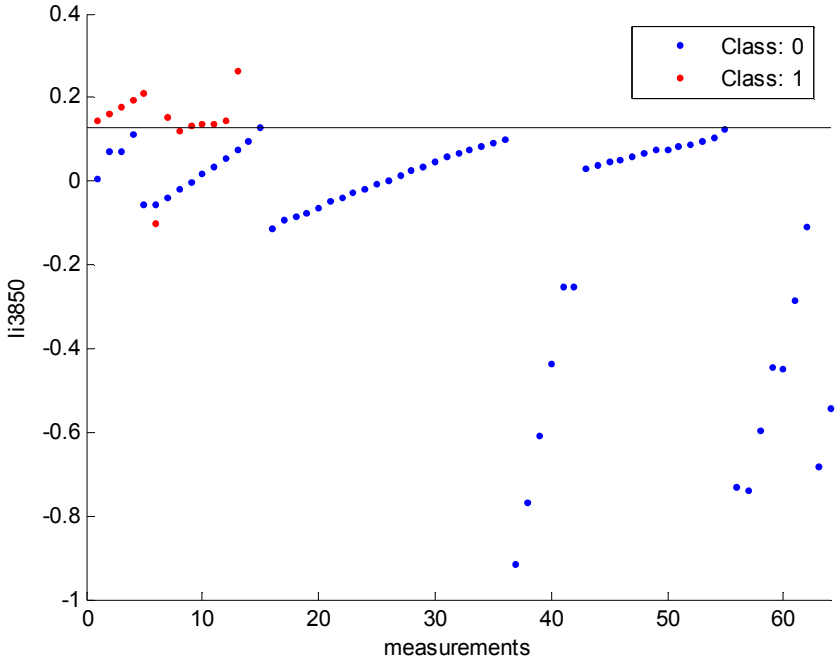

Figure B-7. Split at node 9.

B.3 Decision Tree: Heavy Summer Model

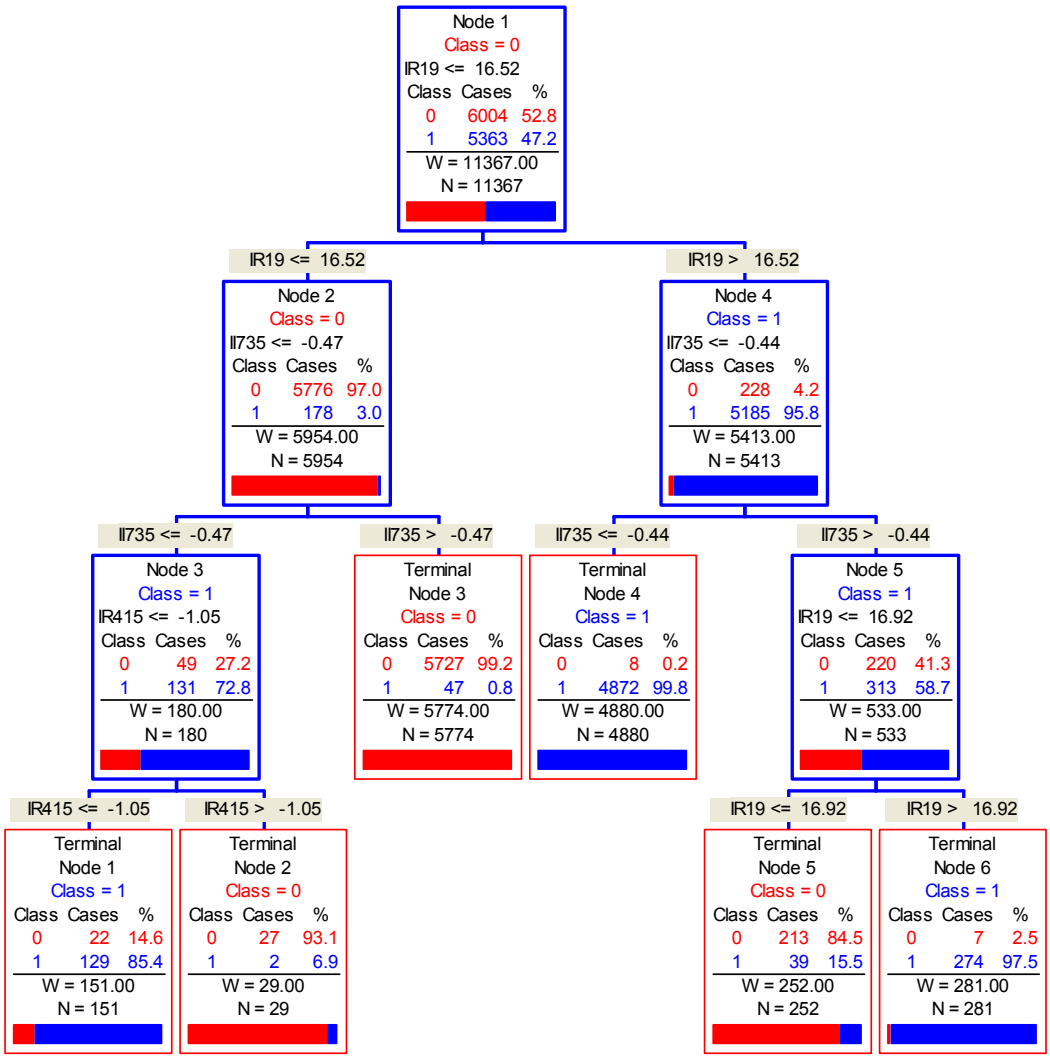

Figure B-8. Decision Tree: Heavy Summer Model.

B.3.1 Partition Sequence

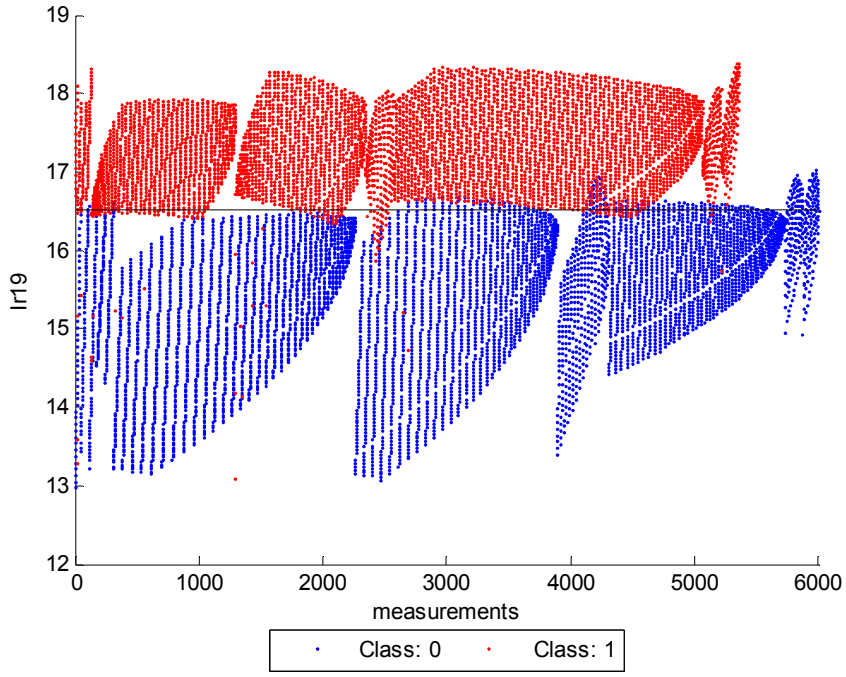

Figure B-9. Split at node 1.

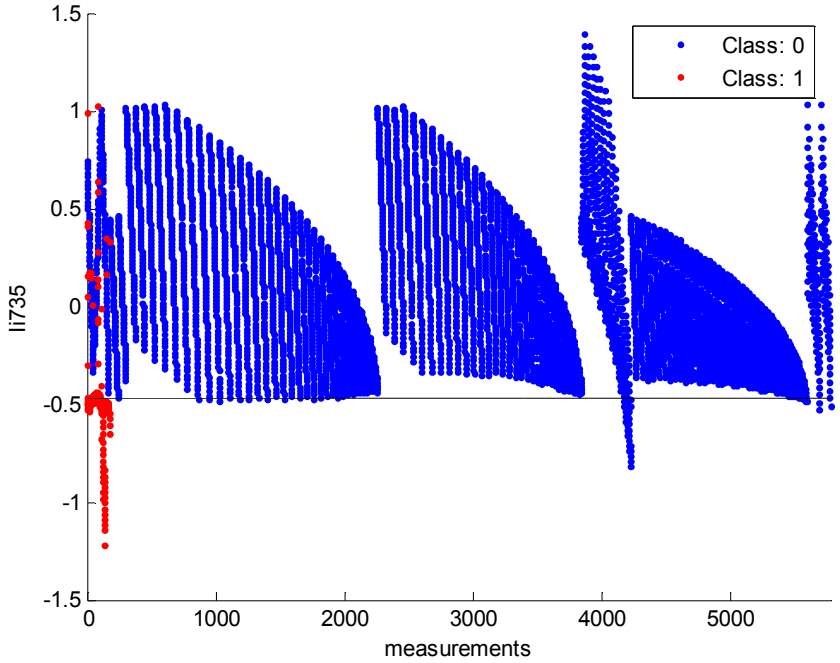

Figure B-10. Split at node 2.

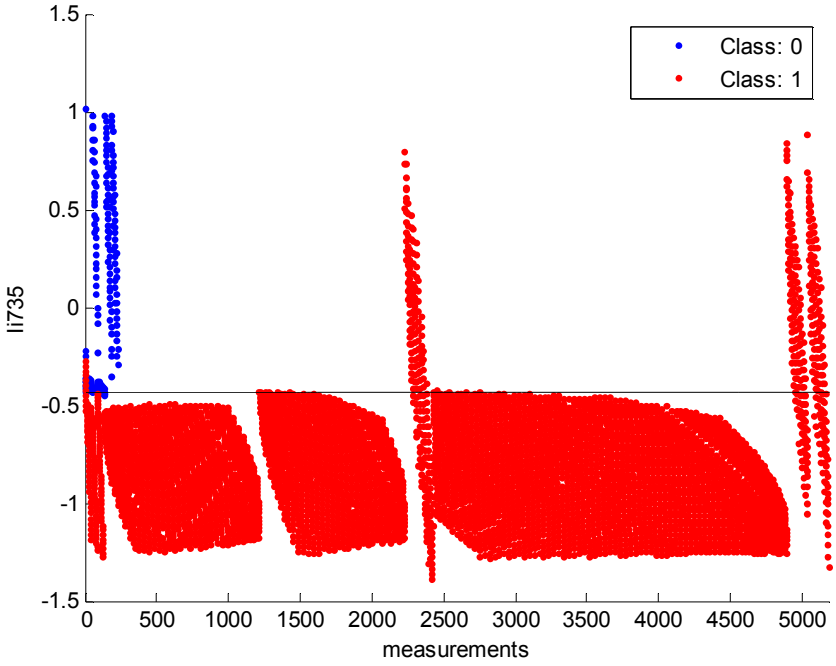

Figure B-11. Split at node 3.

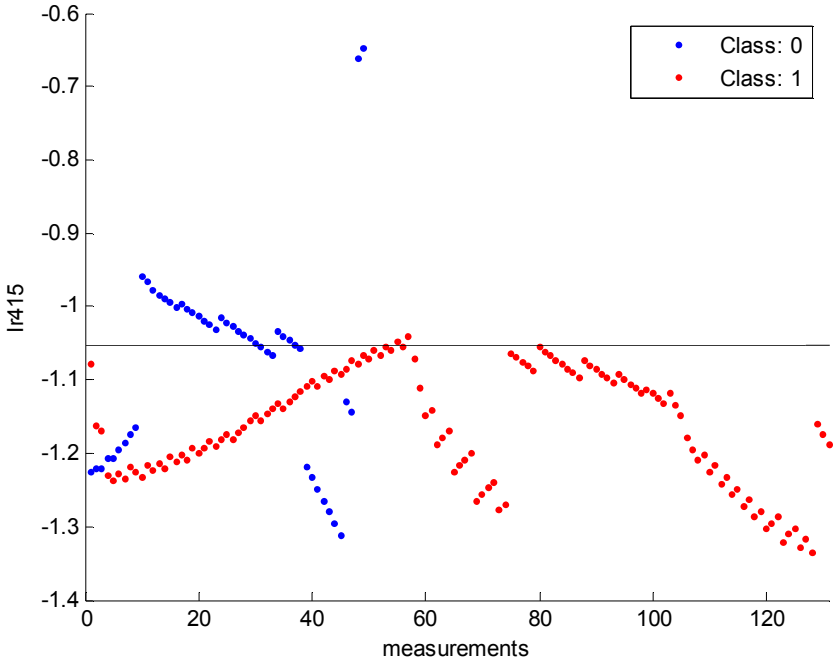

Figure B-12. Split at node 4.

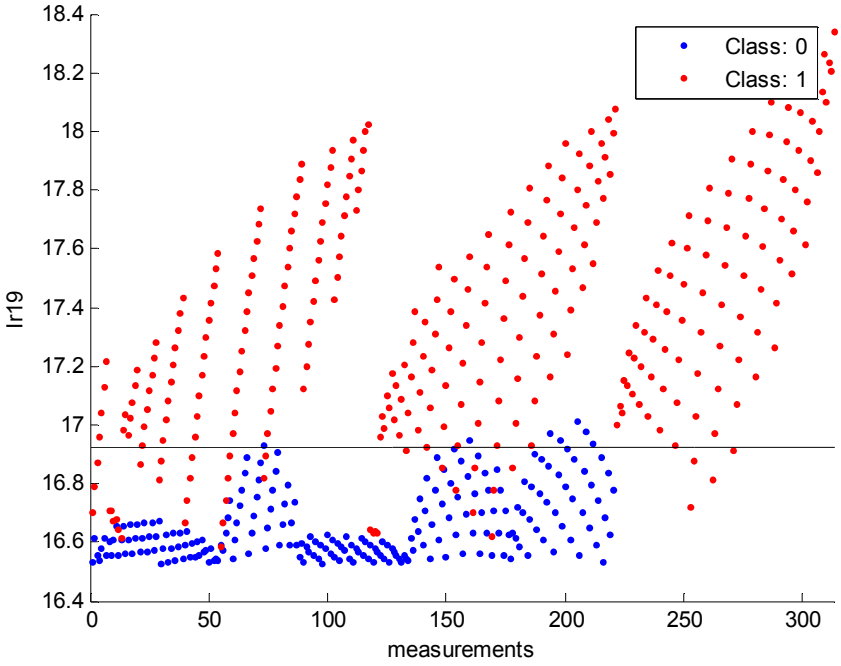

Figure B-13. Split at node 7.