Data Mining to Characterize Signatures of Impending System Events or Performance from PMU Measurements

Final Project Report

Power Systems Engineering Research Center

Empowering Minds to Engineer
the Future Electric Energy System

Data Mining to Characterize Signatures of Impending System Events or Performance from PMU Measurements

Final Project Report

Project Team

Vijay Vittal, Project Leader
Trevor Werho, Graduate Student
Arizona State University

Mladen Kezunovic
Ce Zheng, Graduate Student
Vuk Malbasa, Post-Doctoral Research Associate
Texas A&M University

Junshan Zhang
Miao He, Graduate Student
Arizona State University

PSERC Publication 13-39

August 2013

For information about this project, contact

Vijay Vittal
PO Box 875706
Arizona State University
Tempe, AZ 85257-5706
E-Mail: vijay.vittal@asu.edu
Phone: (480)-965-1879

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
527 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-965-0745

Notice Concerning Copyright Material

Acknowledgements

This is the final report for the Power Systems Engineering Research Center (PSERC) research project titled “Data Mining to Characterize Signatures of Impending System Events or Performance from PMU Measurements” (project S-44). We express our appreciation for the support provided by PSERC’s industry members and by the National Science Foundation under the Industry / University Cooperative Research Center program.

We wish to thank:

- Naim Logic – Salt River Project
- Juan Castaneda – Southern California Edison
- Khaled-Abdul Rahman – California Independent System Operator
- James Kleitsch – American Transmission Company
- Sharma Kolluri – Entergy.

Executive Summary

This project applies data mining techniques to characterize signatures of impending system events or performance from phasor measurement units (PMU) measurements. The project will evaluate available data mining tools and analyze the ability of these tools to characterize signatures of impending systems events or detrimental system behavior. The use of PMU measurements from multiple locations will also be considered. The performance of the data mining tools will be verified by comparing the results obtained for measurements corresponding to know events on the system. The basis of the proposed approach is to use a historical data set of PMU measurements, along with information regarding actual events that occurred on the system during the historical period considered in the data set, and apply the decision tree based data mining techniques available in the commercial software Classification and Regression Trees (CART) to identify signature of impending events. A decision tree can be thought of as a flowchart representing a classification system. It consists of a sequence of simple questions regarding critical attributes (CAs).

The project consists of three parts Part 1 deals with the use of data mining in conjunction with PMU measurements to characterize signatures of impending system events. Part 2 deals with power system oscillatory stability and voltage stability based on voltage and current phasor measurements. Part 3 deals with fundamental research to improve the performance of decision trees using robust ensemble decision trees with adaptive learning and also accounting for loss of PMU measurements. Some details of each part are provided below.

Part 1: Data Mining to Characterize Signatures of an Impending Island Formation from PMU measurements

This study is aimed at using real PMU measurements to predict and detect significant system events with the help of the data-mining tool CART. The program CART (classification and regression trees) produced by Salford Systems is a data-mining tool that can be used to analyze problems that contain a large number of variables. The historical PMU data used in this study is from the Entergy power system in Louisiana when hurricane Gustav impacted the network. During the storm, 14 tie lines were lost that created an electrical island containing Baton Rouge and New Orleans. The PMU measurements captured during the storm where studied in a variety of ways to identify signatures that provide critical information regarding the status of the system.

Careful analysis was conducted to determine whether or not the island could be detected by only using the PMU measurements. It was found that the most effective approach of identifying the creation of the island was to use the PMU measurements of voltage phase angle. By comparing the phase angle measurements between PMUs, in this case, the island could have been detected in approximately 4 seconds. Also, by comparing different sets of PMUs, the location of the island could be determined by which PMUs were inside or outside of the affected area. Because this approach only considers the PMU measurements to form conclusions, the same method could be applied

to any system containing PMUs, with only slight modification, and still provide the ability to quickly and reliably detect the formation of an island within the system.

Provided with the system power flow and dynamic data corresponding to the time when hurricane Gustav entered the system, simulations were conducted to attempt to recreate and match the event to the historical PMU data. Load and generation levels across a wide range of the system were adjusted to closely match the phase angle difference see in the PMU data. Next, the conditions inside the island were adjusted using the known generator dispatch and the available SCADA data. It was found that the direction of the power flowing on the last tie line must have been opposite to the SCADA data. Also, it was found that in order to match the simulation to the PMU frequency measurements, the governor reference at one of the generators must have been reduced just following the creating of the island. Performing these actions allowed the event to closely match to the PMU measurements and provide a better understanding of what happened just after the island formed.

Lastly, the PMU data was used to try to predict the island formation and identify signatures that predicted impending events. Since there was insufficient data to search for signatures by using the single island formation in the available PMU data, 50 simulations were conducted to build a CART database. The simulations were analyzed intuitively and with CART to determine any predicting signatures. It was found that there is a strong correlation between a sudden change in voltage phase angle and the loss of a tie line. A number of simulations also showed a sudden change in voltage within the island area after the loss of a tie line. These different signatures were searched for in the real PMU data at the times when tie lines were reported to have been removed from the system. It was found that when the second to last tie line went offline, there was a 12º change in phase angle measured inside the island. This signature precedes the island formation by 38 minutes and could have alerted system operators that this area needed attention.

This study was successful in using CART, along with an in-depth knowledge of power systems, to analyze PMU data from a historic event. The data-mining tool CART helped quantify and understand the phenomenon observed in the PMU data. The method of identifying an island formation using voltage phase angle measurements is both effective and reliable, and could be used in real applications. The signatures found to predict the island formation is much less reliable. Large changes in load or generation could also create a sudden change in phase angle and the method could be prone to false alarms. This method of island formation prediction could likely be improved by pairing it with additional information, such as SCADA data. However, this study only considers the information that can be drawn from the PMUs alone. In the future as more PMUs are placed in the power system, it is a reasonable assumption that the predicting signatures found in this study will be easier to identify and provide more information.

Part 2: Data Mining to Characterize Impending Oscillatory and Voltage Stability Events

Traditionally, time-domain simulation based on system modeling is used as the primary tool to analyze power system stability. This method is straightforward and accurate as long as an adequate system model and measurements are used. However, two obstacles have prevented this method from being applied in real-time applications: 1) it is

computationally involved; 2) when a simplified model is used, concerns may be raised over approximate analysis results. As the importance of real-time stability monitoring and early detection of system events has been increasingly emphasized in literature recently, an alternate approach based on data mining methods, with a focus on the Decision Tree (DT) method, has been explored in this project.

This report first presents the use of classification trees for rapid evaluation of power system oscillatory stability and voltage stability based on voltage and current phasor measurements. An operating point is grouped into one of several stability categories based on the value of corresponding stability indicator. A new methodology for knowledge base creation has been elaborated to assure practical and sufficient training data sets. Encouraging results are shown using the generated knowledge base and the explored methodology. The impact of DT growing method and node setting on the classification accuracy has been explored in detail.

After that a regression tree-based approach to predicting the power system stability margin and detecting impending system events is proposed. The input features of the regression tree (RT) include the synchronized voltage and current phasors from measurement points across the power grid, gathered using PMUs. Modal analysis and continuation power flow are the tools used to build the knowledge base for off-line RT training. Corresponding metrics include the damping ratio of the critical oscillation mode and MW-distance to the voltage collapse point. The robustness of the proposed predictor to measurement errors and system topology variation is analyzed. The optimal placement for the PMUs based on the importance of RT variables is proposed. The differences in performance between regression tree and several other data mining tools have also been explored.

Next, by using a probabilistic learning tool in the proposed active learning scheme to interactively query a learning data set based on the importance of unlabeled data points, we show that much fewer operating conditions need to be processed via time domain simulation for accurate voltage stability and oscillatory stability estimation. The proposed methodology significantly reduces the computational burden of creating a learning data set.

A measurement-based approach to analyzing the actual PMU measurements without knowledge of detailed system model parameters is presented at the end. DT is used to estimate useful information of inter-area electromechanical oscillations, such as mode frequency and damping ratio, for online oscillatory stability assessment.

Part 3: Data Mining for Online Dynamic Security Assessment using PMU Measurements

This study focuses on online dynamic security assessment (DSA) of power systems by using DTs and real-time PMU measurements. While previous studies have proven the effectiveness of DTs for power system security assessment, two practical issues can compromise the performance of DTs when applied to *online* DSA: 1) power system operating condition (OC) variations and topology changes, which can result in different critical decision rules and inaccurate decisions of DTs; 2) missing PMU measurements of the critical attributes of DTs, which may make data-mining-based

online DSA infeasible. In this study, ensemble DT learning-based online DSA approaches are developed to handle these challenges.

Part 3 first presents a novel approach for handling OC variations and topology changes in online DSA. Different from existing approaches that rely on a single fully-grown DT, the proposed approach utilizes an ensemble of small-height DTs, each of which is assigned a voting weight for final security decision making. These small-height DTs and the corresponding voting weights are identified by using a rigorous gradient-descent algorithm in offline training. As new cases are added to the knowledge base in online DSA, the small-height DTs and the corresponding voting weights are updated, so that the classification model could smoothly track the changing situations of power systems.

Next, online DSA with missing PMU measurements is studied by using ensemble DT learning and a novel random subspace method. Specifically, each small-height DT is trained in a random attribute subspace (i.e., trained by using a randomly selected attribute subset). The random subspace method exploits the hierarchy of wide-area monitoring system (WAMS), the locational information of attributes, and the availability of PMU measurements, so as to improve the overall robustness to missing data. Particularly, in case of missing PMU measurements, the voting weights of small-height DTs are re-calculated for accuracy assurance.

The proposed approaches have been applied to the Western Electricity Coordinating Council (WECC) system, as well as IEEE test systems for illustrative purposes. The effectiveness of the proposed approaches is demonstrated via several case studies, by using a variety of realized system OCs and practical WAMS reliability indices.

Project Publications

Student Theses:

Trevor Werho – Arizona State University – "Application of Data Mining Techniques to PMU Measurements to Detect Impending Signatures of System Failures," PhD, Anticipated Date of Graduation: May 2014.

Miao He – Arizona State University – "A Data Analytics Framework for Smart Grids: Spatio-temporal Wind Power Analysis and Synchrophasor Data Mining," PhD, Date of Graduation: August 2013.

Conference Papers:

V. Malbasa, C. Zheng, and M. Kezunovic, – Texas A&M – "Power system online stability margin estimation using active learning and synchrophasor data," PowerTech 2013, Grenoble, France, June 2013.

C. Zheng, V. Malbasa, and M. Kezunovic, – Texas A&M – "A fast stability analysis scheme based on classification and regression tree," IEEE Conference on Power System Technology (POWERCON), Auckland, New Zealand, October 2012.

C. Zheng, V. Malbasa, and M. Kezunovic, – Texas A&M – "Online estimation of oscillatory stability using synchrophasors and a measurement-based approach," submitted to 17th International Conference on Intelligent System Applications to Power Systems (ISAP), Tokyo, Japan, July 2013.

M. He, V. Vittal and J. Zhang, – Arizona State University – "A Data Mining Framework for Online Dynamic Security Assessment: Decision Trees, Boosting, and Complexity Analysis," IEEE PES Conference on Innovative Smart Grid Technologies, Washington DC, United States, Jan. 2012.

Journal Papers:

C. Zheng, V. Malbasa, and M. Kezunovic, – Texas A&M – "Regression tree for stability margin prediction using synchrophasor measurements," *IEEE Transactions on Power Systems*, Vol. 28, No. 3, May 2013.

M. He, V. Vittal, and J. Zhang, – Arizona State University – "Online dynamic security assessment with missing PMU measurements: A data mining approach," *IEEE Transactions on Power Systems*, Vol. 28, No. 2, PP. 1969–1977, May 2013.

M. He, J. Zhang and V. Vittal, – Arizona State University – "Robust On-line Dynamic Security Assessment using Adaptive Ensemble Decision Tree Learning," accepted for publication, *IEEE Transactions on Power Systems.*

Table of Contents

List of Figures

List of Tables

1 Data Mining to Characterize Signatures of an Impending Island Formation from PMU Measurements

1.1 Introduction

The objective of this aspect of the project is to examine the efficacy of the commercial data-mining tool CART, in identifying signatures of impending power system events by using actual phasor measurement unit (PMU) measurements. The historical PMU data used in this study is from the Entergy power system in Louisiana. In September of 2008, hurricane Gustav made landfall in southern Louisiana. During the course of the storm an electrical island was formed around Baton Rouge and New Orleans. This study aims to use CART to analyze the PMU measurements captured during the hurricane to better understand future islanding events.

1.1.1 CART

The program CART (classification and regression trees) produced by Salford Systems is a data-mining tool that can be used to analyze problems that contain a large number of variables. CART uses a procedure called binary recursive partitioning to build a decision tree. Starting at the root node, simple questions called critical splitting rule (CSR) are asked regarding a critical attribute (CA). Each answer to the question creates two branching nodes such that each will have its own CSR. Nodes that do not branch off to other nodes are called terminal nodes that end the growth of the tree. Once all terminal nodes are reached the decision tree is complete and can be used to categorize new inputs. When given input and output data, CART will determine its inherent input-output relationship in the form of a decision tree. This process is called decision tree training.

Once training is complete, new input data can be dropped down the decision tree to generate the previously unknown output. Using this method, the historical PMU measurements will serve as the necessary information needed to train the decision tree. By training the decision tree, CART will determine any precursor signatures of the impending system event that is contained within the data. With this decision tree completed, new PMU measurements could be dropped down the tree to determine if a particular system event is likely to occur in the future [1].

1.1.2 Sample Case from Entergy

The historical PMU data used in this study is from the Entergy power system in Louisiana. On September 1, 2008 at 9:30 AM hurricane Gustav made landfall close to New Orleans. Over the course of several hours the Entergy system lost 13 tie lines that interconnected the Baton Rouge and New Orleans area to the rest of the grid. At 2:49 PM the 14^{th} and final tie line was tripped that resulted in the formation of an electrical island containing most of Baton Rouge and New Orleans in southeast Louisiana. At the time of the island there were 19 PMUs within the Entergy system that recorded the islanding event. Each PMU was capable of measuring voltage phasor, current phasor, frequency and frequency rate of change. This historical PMU data was made available by Entergy for the purpose of this study [2].

1.2 Island Detection Analysis

Any prediction signatures detectable by PMUs would be most effective if an island formation could be quickly and reliably detected. Historical PMU frequency and voltage phase angle measurements from different PMUs in the Entergy system were

studied to determine whether island formation, time and location, could be detected with a high level of confidence.

To determine if the location of the Entergy island could be predicted by only using the PMU measurements, the frequency data from four PMUs were selected; Mablevale, Sterlington, Ninemile, and Waterford. One hour of frequency data around the reported time of islanding for each of the selected PMUs can be seen in Figure 1.1, Figure 1.2, Figure 1.3, and Figure 1.4.

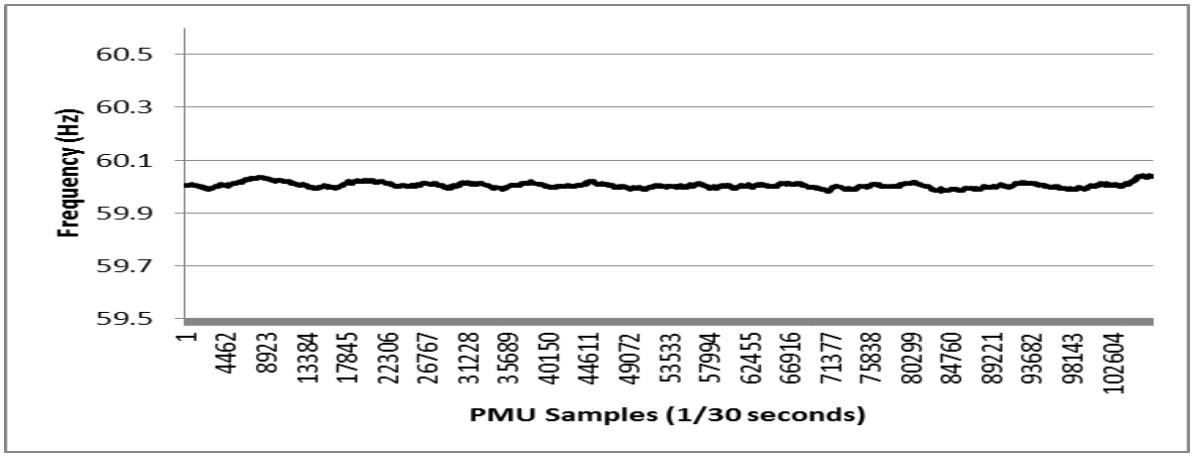


Figure 1.1 Mablevale frequency versus time

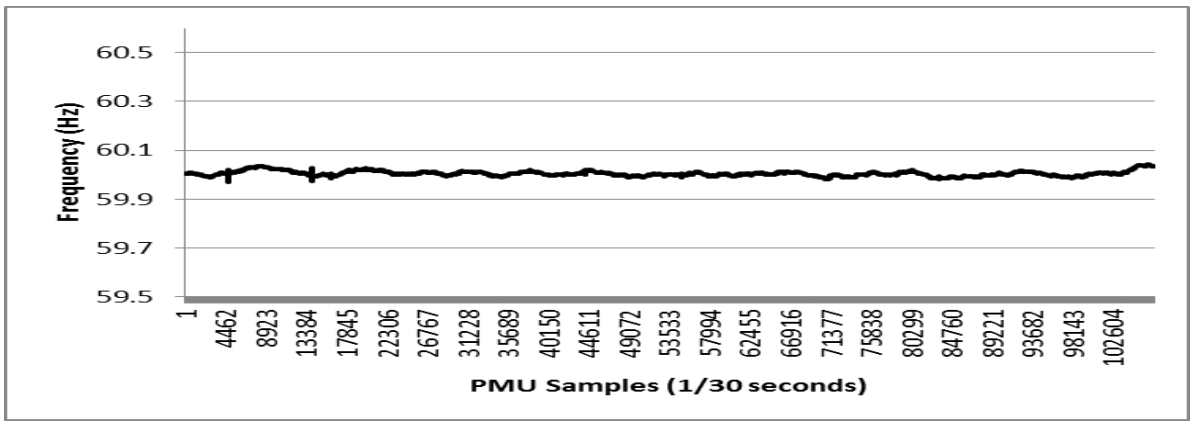


Figure 1.2 Sterlington frequency versus time

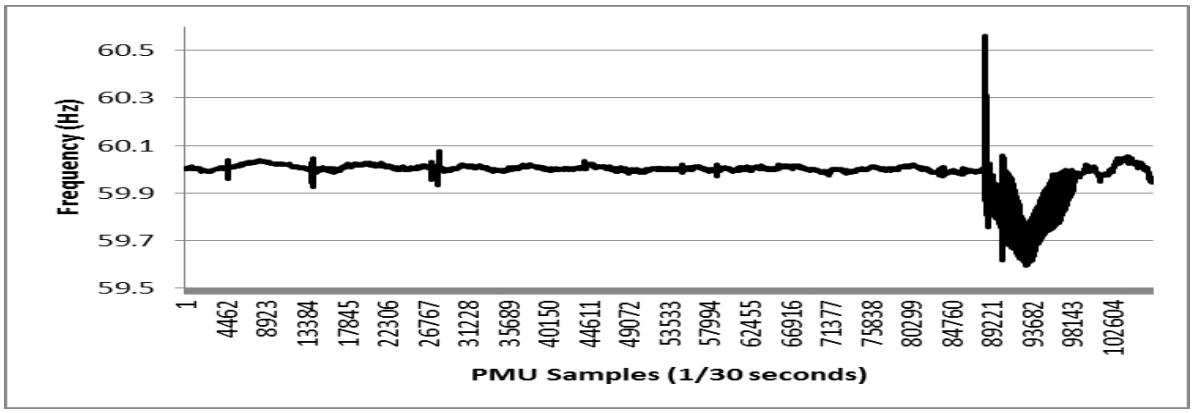


Figure 1.3 Ninemile frequency versus time

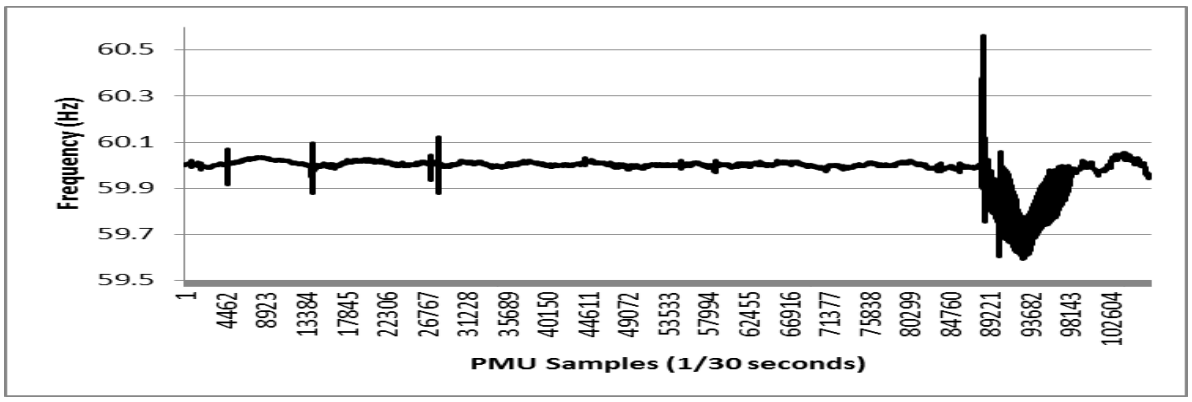


Figure 1.4 Waterford frequency versus time

It became apparent that by comparing data from different PMUs the location of the island could be observed and determined. It is clear that Mablevale and Sterlington are connected together as well as outside the island. Also, it can be seen that Ninemile and Waterford are connected together and are inside the island. Therefore, the instant at which an island formation is detected; comparing measurements from different PMUs would allow the affected area to be determined.

Two PMUs were selected, Waterford and Sterlington, to determine the amount of time needed to determine that an island had formed in the Entergy system. Normally, the PMU phase angle measurements are bounded between +240° and -180°. If the phase angle goes to 241°, the measurement will report -179°. In this way the data is "wrapped"

around the interval +240° to -180°. In order to sensibly view the phase angle measurements, the data must first be "unwrapped". This is done by taking phase angle measurements from two PMUs and taking their difference. Logic must then be used to ensure the data always lies between +240° and -180°. Once this is done the data will still be bounded to between +240° and -180°. To remove this bound, logic can be used to check the conditions just before the 360° jumps in angle and can shift accordingly so the data is a continuous curve [3]. The adjusted difference in voltage phase angle at Waterford and Sterlington was plotted around the time of island formation and can be seen in Figure 1.5.

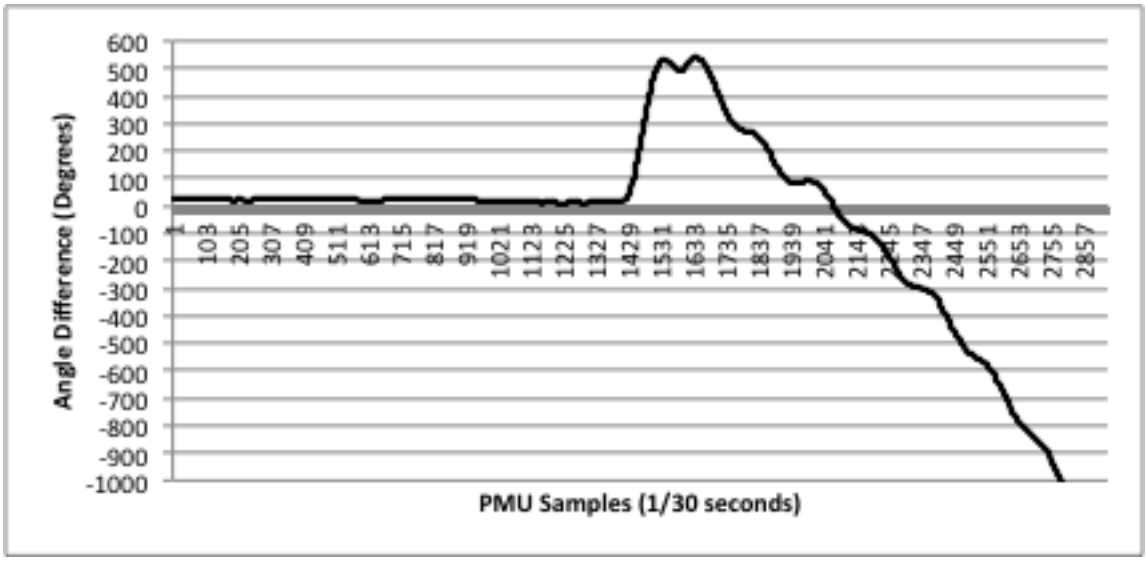


Figure 1.5 Waterford-Sterlington phase-angle difference versus time

Before island formation the voltage phase angle difference between Waterford and Sterlington was ~20º. Approximately four seconds after island formation the voltage phase angle climbs to over 500 degrees. At this point it is clear that Waterford and Sterlington are no longer connected. Therefore, in this case, the island formation could have been detected in ~4 seconds.

1.2.1 Island Detection CART Analysis

The initial analysis of island formation detection was mostly intuitive. The frequency and voltage phase angle data corresponding to the island formation and island reconnection was analyzed with CART to provide a more quantitative approach to identifying the island.

In order for CART to run an analysis it must first have a database. This database must be in a specific format in order to be used. An example CART database can be seen in Table 1.1.

Table 1.1 Example CART Database

Output Label	**Input 1**	**Input 2**
1	10	X
0	10	Y
0	10	Y
1	10	Z
1	10	X
1	10	X
0	15	Y
1	20	X
1	25	X
0	30	X

The CART database can have any number of input variables up to what the CART license will allow. Adding an additional input variable would increase the CART

database by 1 column. Additional rows may be added to the database to increase the amount of data points included in the analysis. Each CART input variable can be either continuous or categorical. The database constructed for island formation detection analysis contained 4 input variables. The variables used in the CART database are as follows:

- Frequency and voltage phase angle data measured at Waterford (inside island)
- Frequency and voltage phase angle data measured at Ninemile (inside island)
- Frequency and voltage phase angle data measured at Sterlington (outside island)
- Frequency and voltage phase angle data measured at Mablevale (outside island)

The output label used in the island detection database was **Island** or **No Island** depending upon the time the frequency and phase angle measurements were taken and whether or not the island was present in the system. The final CART island detection database contained 9 columns and approximately 120,000 rows (1 hour of PMU recordings).

The program CART uses the database to train a decision tree. A decision tree contains a specific input-output relationship. Using the decision tree requires one value of each of the input variables included in the study that correspond to the same sample. An example of this would be any of the rows of the CART database minus the output label. Starting at the top most node; apply the logic rule to the input data. This rule will then point to one of the two adjacent nodes. Continuing to apply the rules at each node will eventually point to a terminal node or a node that does not lead to any of lower nodes. Each terminal node corresponds to one of the possible output labels. The label of the terminal node that the inputs lead to is the output that corresponds to those particular inputs. In this way, a specific output category can be given to every set of input data.

The PMU data used in the CART island formation analysis was the frequency and voltage phase angle data from Ninemile, Waterford, Sterlington, and Mablevale. The PMU frequency data was given to CART unmodified while the voltage phase angle measurements required the same adjustments done in the previous island detection analysis. In the CART database all of the phase angle measurements are relative to Sterlington and the phase angle measurements at Sterlington are entered as all zeros. For example, the statement (Ninemile Phase = -10°) means the phase angle at Ninemile is 10° less than the phase angle at Sterlington. The decision tree generated by the CART island formation analysis can be seen in Figure 1.6.

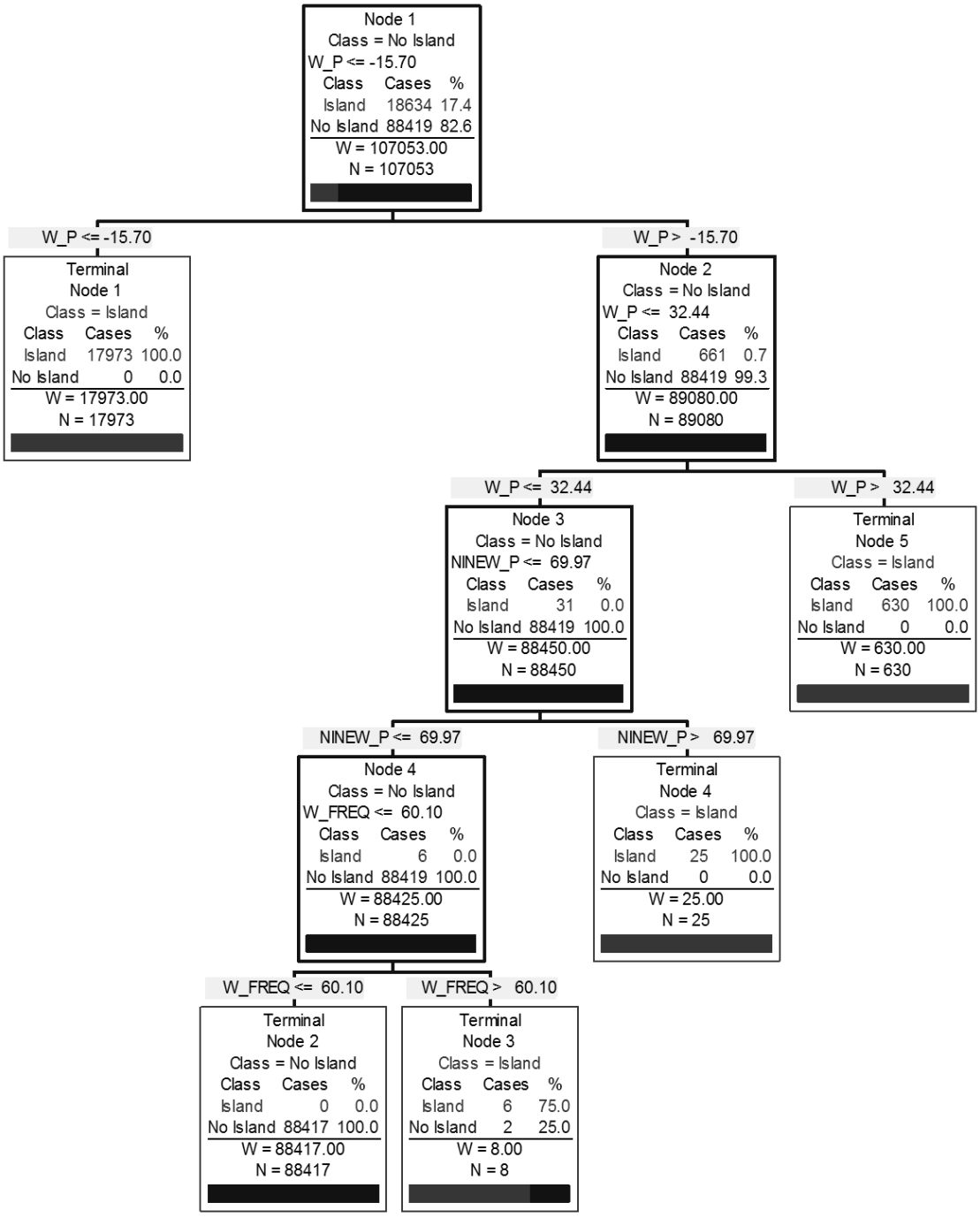


Figure 1.6 Decision tree created from island formation data

The decision tree created by CART contains 9 nodes. The properties of each node are as follows:

- Node 1:
    - If Waterford phase < -15.7° go to Terminal Node 1
    - If Waterford phase > -15.7° go to Node 2
- Terminal Node 1:
    - Terminal Node
    - Label: Island
    - Conditions to reach: Waterford phase < -15.7°
    - 17973 data points of the training data lead to this terminal node
- Node 2:
    - If Waterford phase > 32.44° go to Terminal Node 5
    - If Waterford phase < 32.44° go to Node 3
- Terminal Node 5:
    - Terminal Node
    - Label: Island
    - Conditions to reach: Waterford phase > 32.44°
    - 630 data points of the training data lead to this node
- Node 3:
    - If Ninemile phase < 69.97° go to Node 4
    - If Ninemile phase > 69.97° go to Terminal Node 4
- Terminal Node 4:
    - Terminal Node
    - Label: Island
    - Conditions to reach: Waterford phase > -15.7° and Waterford phase < 32.44°, Ninemile phase > 69.97°
    - 25 data points of the training data lead to this node
- Node 4:
    - If Waterford frequency < 60.1 Hz go to Terminal Node 2
    - If Waterford frequency > 60.1 Hz go to Terminal Node 3

- Terminal Node 3:
    - Terminal Node
    - Label: Island
    - Conditions to reach: Waterford phase > -15.7° and Waterford phase < 32.44° and Ninemile phase < 69.7° and Waterford frequency > 60.1Hz
    - 8 data points of the training data lead to this node
- Terminal Node 2:
    - Terminal Node
    - Label: No Island
    - Conditions to reach: Waterford phase > -15.7° and Waterford phase < 32.44° and Ninemile phase < 69.97° and Waterford frequency < 60.1 Hz
    - 88417 data points of the training data lead to this node

Scoring the decision tree using the training data shows the decision tree is correct 99.999% of the time. However, it would still not be judicious to apply this decision tree to a future island formation, even to an islanding event in the same location. This is because the CART database only contains the data from a single islanding event. Only by training a decision tree with many different island formations would the decision tree become reliable enough to be implemented for island detection. However, this decision tree is still useful. Notice that about 99.97% of all training data points lead to terminal nodes 1, 2, and 5. These are the dominant terminal nodes. Combining the rules of all three dominant nodes leads to the statement; if the phase angle at Waterford is between +32.4396° and -15.6969° from Sterlington then there is no island, otherwise, an island must exist. The exact values of the thresholds found by CART are unique to this particular event, but the rule suggests that when the phase angles recorded at PMUs within one area differ greatly from PMUs outside that area, then there is a high likelihood

an island has formed. This is very similar to what was seen in the intuitive analysis of island detection. It can be seen that CART only uses frequency data to classify 8 data points of the ~120,000 points of training data. This supports the finding that voltage phase angle measurements are much more sensitive to island formations than frequency.

A CART analysis was also done using the PMU data corresponding to the island resynchronization. The PMU data used in the CART island resynchronization analysis was the frequency and voltage phase angle data from Ninemile, Waterford, Sterlington, and Mablevale. The decision tree created by CART island resynchronization analysis can be seen in Figure 1.7.

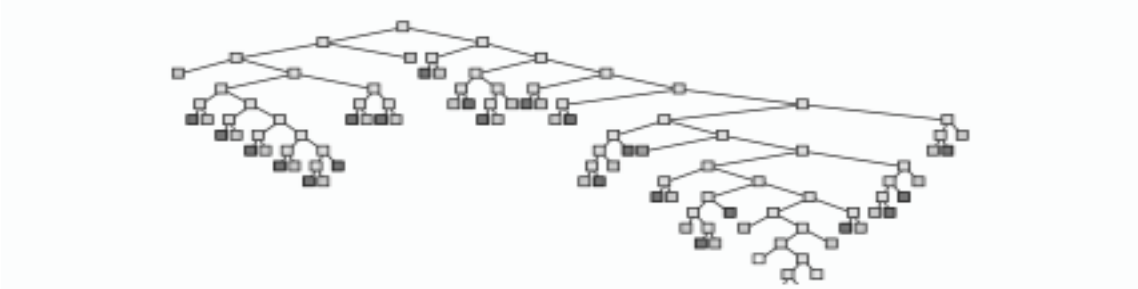


Figure 1.7 Decision tree created from island resynchronization data

The decision tree created from the resynchronization data contains 55 nodes. The bounds on the phase angle data used in the resynchronization cannot be removed like in the island formation analysis. In the island formation event it can be assumed that before island formation the phase angles are within 360° of one another. Once the island forms and the phase angles begin to wrap, it can be assumed that the phase angle has exceeded the bounds of +240° and -180°. However, in the island resynchronization event, before resynchronization the phase angles could be any multiple of 360° away from one another. Once the areas reconnect the phase angles will only approach the closest multiple of 360°. Because of this reason, it is unknown how far to shift the phase angles before the

resynchronization occurs and the phase angle data cannot be modified like the previous analysis. Because the phase angle data does not contain the same information as the previous analysis CART must use the frequency data when detecting the island reconnection. This causes the decision tree to become much more complex. The decision tree's dominant nodes state that if Ninemile phase angle is within +43.11° and -73.0714° of Sterlington *and* Ninemile frequency is within .04 Hz of Mablevale then there is no longer an island present. If these conditions are not true then an island is present. These rules show some resemblance to the rules found in the island formation decision tree. Here the bound in phase angle is much wider but still suggests that a large difference in phase angle is a strong indicator of an island being present.

1.3 Simulation of Entergy Power System

The power flow and dynamic data, that represents the Entergy Power system as it was at 2:49 PM on September 1, 2008 when the southeast area disconnected from the rest of the system, was made available by Entergy for the purpose of this study. Previous attempts by other investigators to recreate the islanding event in simulation did not match the historical PMU data [4]. As part of this study, the system power flow and dynamic data were used to accurately represent the system at the time of islanding. Several modifications were made to the system data in order to generate these results.

1.3.1 Network Data Modification

The first modification from the original data was done using the historical PMU measurements of phase angle. The PMU measurements showed that the phase angle between the islanding area and the center of the Entergy system at 8:00 am was around

11º (with the inside of island leading the outside). However, the power flow data showed that the phase angle between these two areas was around 40º (outside of island leading the inside). The load and generation over a large area were reduced. Additionally, the generation at bus #303007 was increased. These changes resulted in a phase angle difference of 9.24º (inside of island leading the outside). The adjusted areas can be seen in Table 1.2. The adjusted generator can be seen in Table 1.3.

Table 1.2 Area Load and Generation Modifications

Area Numbers	**Area Names**	**Old Values**	**New Values**
332	LAGN	P Load = 15632	P Load = 13248
351	EES	Q Load = 5288	Q Load = 4476
502	CELE	P Gen = 14750	P Gen = 12753
503	LAFA		

Table 1.3 Generation Modification

Bus	**Old Value**	**New Value**
303007	5.49 MW	575 MW

The next modification from the original data was done using the known generator dispatch and tie line SCADA data. The original data was modified such that there were only 3 generators online within the islanded area. The generator Ninemile Unit 5 was set to 220 MW, Waterford Unit 1 was set to 49 MW, and Gypsy Unit 2 was set to 77 MW. All other generators in the islanded area were turned off. The load within the island was determined using the known generator dispatch and the SCADA data from the tie line Gypsy-Fairview 230kV, which was last to go offline. The SCADA data available from the last tie line shows that power was flowing into the island. It was later determined through simulation that power must have been flowing out of the island instead but with

similar magnitude. The generators modified within the island can be seen in Table 1.4 and the final island generator and load settings can be seen in Table 1.5.

Table 1.4 Generators Modified Within Island

Bus Number	Bus Name	P_{old}	P_{new}
336002	GA Gulf	8	offline
336151	WAT U1	41	49
336179	UCARBST2	39	offline
336222	GYP U2	36	77
336252	NMIL U5	220	220

Table 1.5 Final Island Generator and Load Settings

Bus Numbers	P Gen	Q Gen	P Load	Q Load
335568-335572 335601 335613-335620 335665 336001-336464	346	-358	246	56

1.3.2 Dynamic Data

Within the island there are three generators online: Waterford Unit 1, Gypsy Unit 2, and Ninemile Unit 5. The dynamic data used for these three generators can be seen in Tables 1.6, 1.7, and 1.8. The generator dynamic models and the exciter dynamic models were not altered from the dynamic data received from Entergy. However, the governor dynamic models did receive modifications. First, the governor model for Ninemile was originally set as an IEESGO model. When initial simulations did not match the PMU data, the governor model was switched to a TGOV1 model to match the other governor models. This was done to remove any variations a difference in governor models might

cause while investigating the simulated event. Also, the parameter values for all governor models were reverted back to the default model values. Finally, the values of R were increased on all governor models. These changes helped modify the initial simulated frequency response after island formation.

Table 1.6 Generator Dynamic Models

Generator	**Waterford U1**	**Gypsy U2**	**Ninemile U5**
Model	GENROU	GENROU	GENROU
T'do	5.6	4.6	4.33
T"do	.05	.05	.041
T'qo	1	.52	.481
T'qo	.06	.072	.059
H	2.539	2.944	2.62
D	0	0	0
Xd	1.9701	1.5795	1.783
Xq	1.9305	1.512	1.764
X'd	.3247	.1849	.291
X'q	.5692	.3901	.411
X"d=X"q	.2148	.1251	.249
X1	.1611	.1201	.199
S(1.0)	.074	.1	.11
S(1.2)	.381	.464	.119

Table 1.7 Exciter Dynamic Models

Exciter	Waterford U1	Gypsy U2	Ninemile U5
Model	IEEX2A	IEEX1	IEEX1
TR	0	0	0
KA	500	50	400
TA	.04	.06	.02
TB	0	0	0
TC	0	0	0
VRMAX	1.5	1.5	8.15
VRMIN	-1.5	-1.5	-7.33
KE	.05	-.045	1
TE	.2	.5	1.21
KF	.08	.08	.03
TF1	1	1	1
E1	3.1875	3.3784	2.71
SE(E1)	.17	.074	.94
E2	4.25	4.5045	3.62
SE(E2)	.24	.267	1.25

Table 1.8 Governor Dynamic Models

Governor	Waterford U1	Gypsy U2	Ninemile U5
Model	TGOV1	TGOV1	TGOV1
R	.07	.07	.07
T1	.5	.5	.5
Vmax	1	1	1
Vmin	0	0	0
T2	3	3	3
T3	10	10	10
Dt	0	0	0

1.3.3 Dynamic Simulation

The islanding event recorded by the PMU measurements was recreated in dynamic simulation. The island at the time of the hurricane had 14 tie lines that interconnected the islanded area to the rest of the system. The network data used for the simulation contains 13 of the 14 tie lines. When the line connecting bus #303153 to bus #335507 was added to the network the solution would not converge. It was decided to

neglect this tie line because it is a lower voltage tie line (138kV) and other simulations have shown this line has little impact during the event.

The simulation was conducted by removing the tie lines in the order that they went offline during the hurricane. However, line Coly – Willow Glen 500kV was not disconnected in the same way as the other tie lines. Information available indicated that this line was unable to serve the island because the three transformers at Willow Glen went offline. To simulate this scenario the Willow Glen bus #335618 was disconnected from the system rather than disconnecting the tie line.

Many simulations were performed in order to understand the governor response to the island formation. It was concluded that actions must be taken during the dynamic simulation in order to match the simulation frequency to the historical PMU data. Four seconds after the island is formed in the simulation the load within the island is increased 40 MW. After eight seconds, the load in the island is increased again another 80 MW. The exact actions taken during simulation are shown in Table 1.9.

Table 1.9 Actions Taken During Dynamic Simulation

Time of Action	Bus Number(s)	Action Taken
0-5sec	--	Flat Line Run
5sec	336462-500360	Disconnect Line
10sec	336015-336016	Disconnect Line
15sec	336032-303202	Disconnect Line
20sec	336141-303204	Disconnect Line
25sec	336006-336007	Disconnect Line
30sec	335568-335660	Disconnect Line
35sec	335536-335665	Disconnect Line
40sec	335771-303200	Disconnect Line
45sec	335500-335618	Disconnect Line
50sec	335568-335659	Disconnect Line
55sec	335657-335658	Disconnect Line
60sec	335618	Disconnect Bus
70sec	336190-336138	Disconnect Line
74sec		Increase island load 40 MW
82sec		Increase island load 80 MW
100sec		End Simulation

The island formation captured by PMU measurements can be seen in Figure 1.8. The frequency inside the island was measured at Ninemile. The frequency outside the island was measured at El Dorado. The plot of frequency of the simulated event can be seen in Figure 1.9.

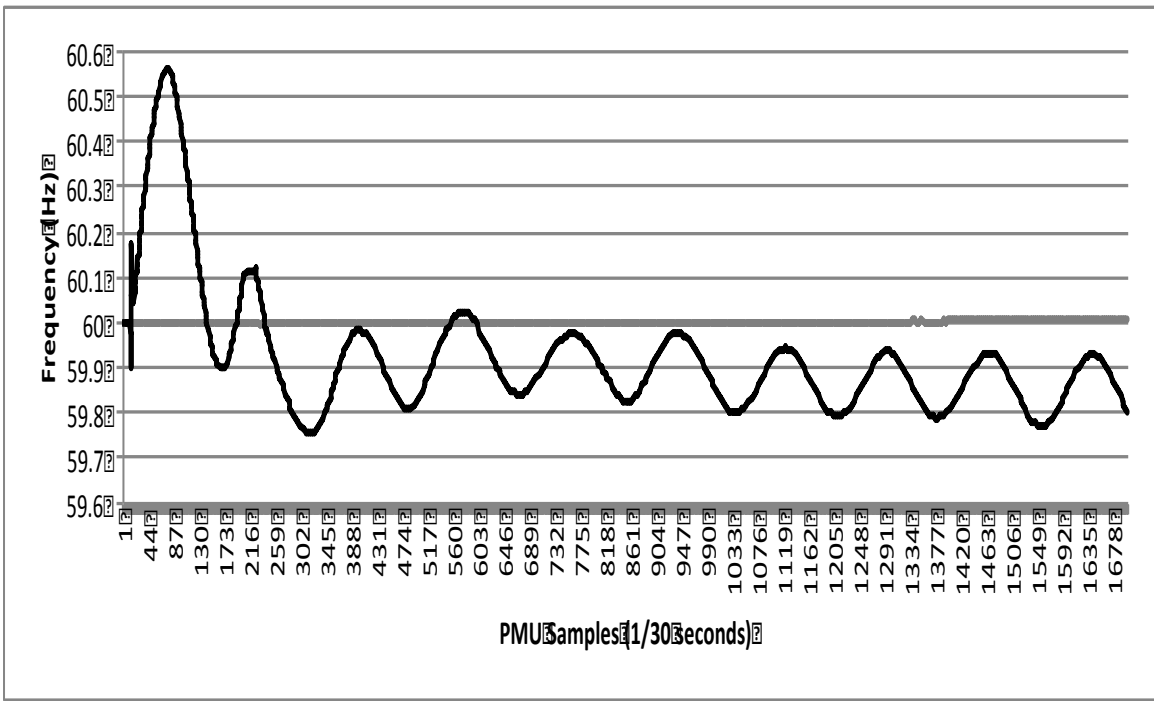


Figure 1.8 PMU frequency measurements of island formation

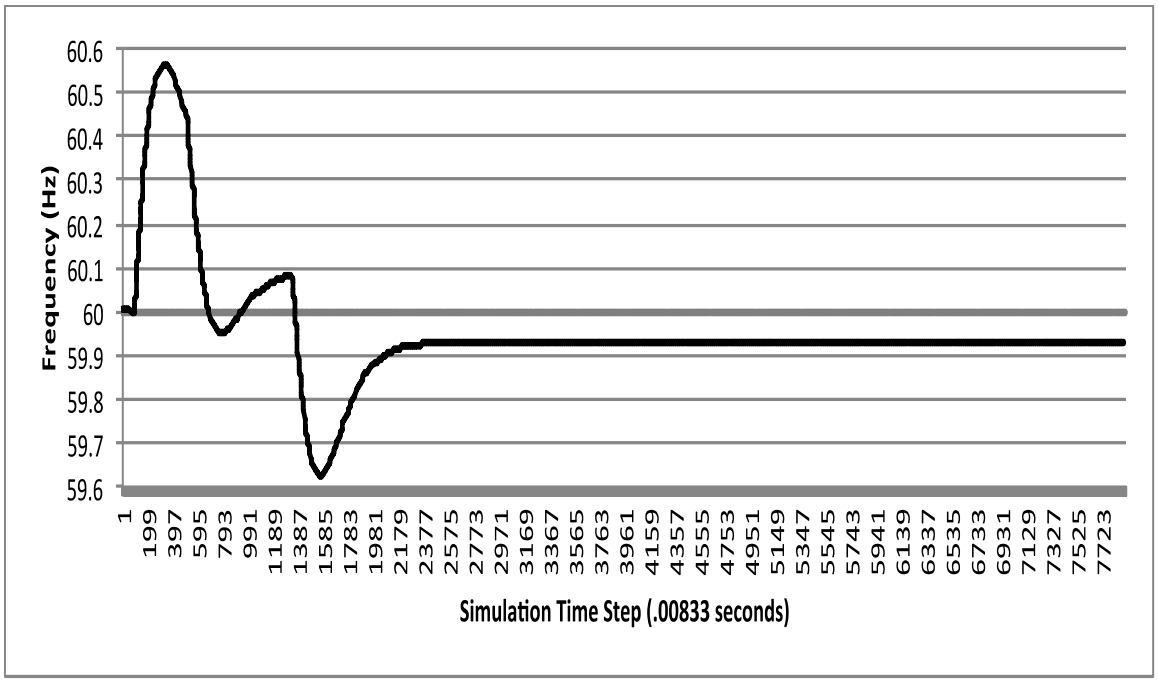


Figure 1.9 Frequency plot of simulated island formation

The simulated event has a maximum frequency of 60.542 Hz, compared to the PMU measurements of 60.54 Hz. Also, the duration of the first peak in simulation is

around 5.6 seconds compared to around 5 seconds in the PMU data. It can be seen that the frequency in the island under simulation recovers to 59.934 Hz. The frequency in the PMU data (ignoring present oscillations) recovers to about 59.9 Hz. The oscillations seen in the PMU data were being driven by some unknown source within the island. Those oscillations would not be able to be captured by the dynamic simulation.

It was decided that generation reduction just after island formation was much more feasible than the load increasing. A second simulation was conducted using the same conditions as the previous island formation simulation. Instead of scaling load after island formation, the governor reference at Ninemile Unit 5 was reduced while leaving the other generators, Waterford Unit 1 and Gypsy Unit 2, unmodified. Four seconds after island formation the governor reference at Ninemile Unit 5 was reduced by 6 MW. Twelve seconds after island formation the governor reference at Ninemile Unit 5 was reduced by an additional 2.5 MW. The exact actions taken during simulation are shown in Table 1.10.

Table 1.10 Actions Taken During Second Dynamic Simulation

Time of Action	Bus Number(s)	Action Taken
0-5sec	--	Flat Line Run
5sec	336462-500360	Disconnect Line
10sec	336015-336016	Disconnect Line
15sec	336032-303202	Disconnect Line
20sec	336141-303204	Disconnect Line
25sec	336006-336007	Disconnect Line
30sec	335568-335660	Disconnect Line
35sec	335536-335665	Disconnect Line
40sec	335771-303200	Disconnect Line
45sec	335500-335618	Disconnect Line
50sec	335568-335659	Disconnect Line
55sec	335657-335658	Disconnect Line
60sec	335618	Disconnect Bus
70sec	336190-336138	Disconnect Line
74sec		Ninemile Gref reduced 6MW
82sec		Ninemile G ref reduced 2.5 MW
100sec		End Simulation

The island formation captured by PMU measurements can be seen in Figure 1.10. The frequency inside the island was measured at Ninemile. The frequency outside the island was measured at El Dorado. The plot of frequency of the simulated event can be seen in Figure 1.11.

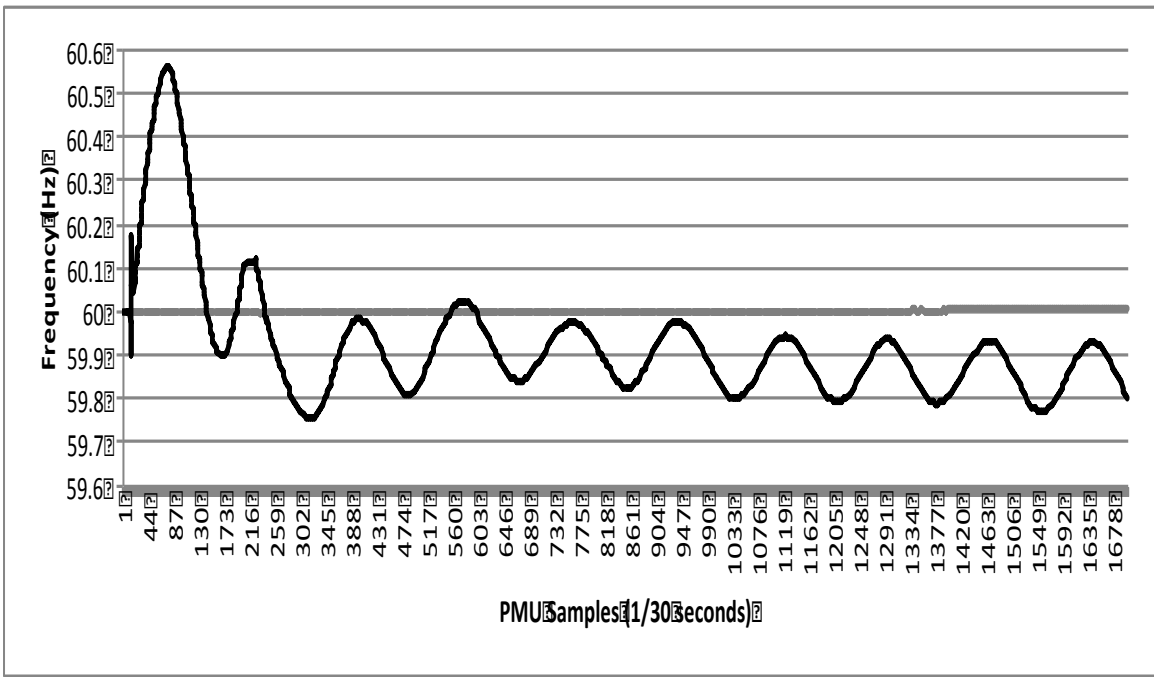


Figure 1.10 PMU frequency measurements of island formation

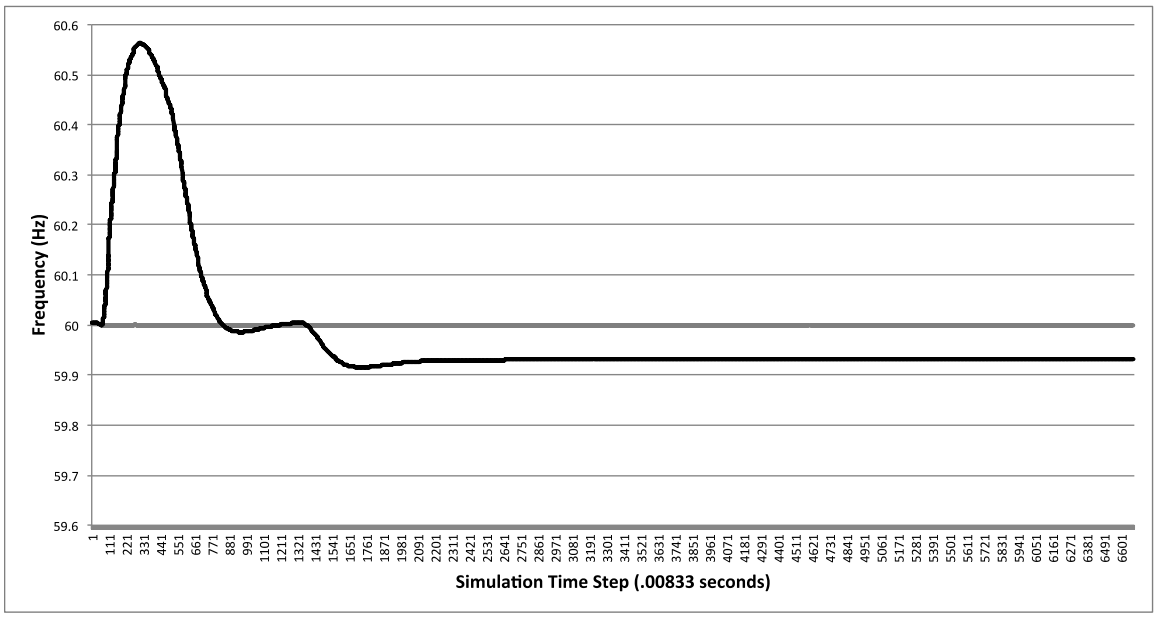


Figure 1.11 Frequency plot of simulated island formation

The simulated event has a maximum frequency of 60.542 Hz, compared to the PMU measurements of 60.54 Hz. Also, the duration of the first peak in simulation is around 7.7 seconds compared to around 5 seconds in the PMU data. It can be seen that

the frequency in the island under simulation recovers to 59.934 Hz. The frequency in the PMU data (ignoring present oscillations) recovers to about 59.9 Hz. Just as in the first simulation, the oscillations seen in the PMU data are not captured in this dynamic simulation.

The plot of Gypsy-Fairview 230kV historic line flows can be seen in Figure 1.12. The plot of simulated line flow of Gypsy-Fairview 230kV can be seen in Figure 1.13.

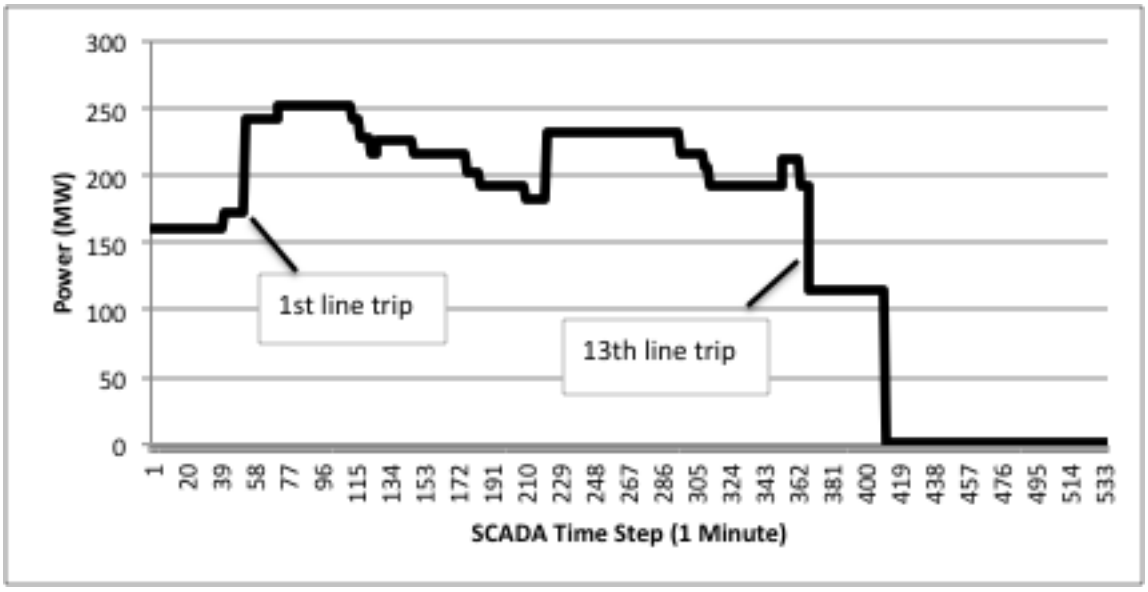


Figure 1.12 Historical line flows of Gypsy – Fairview 230 kV

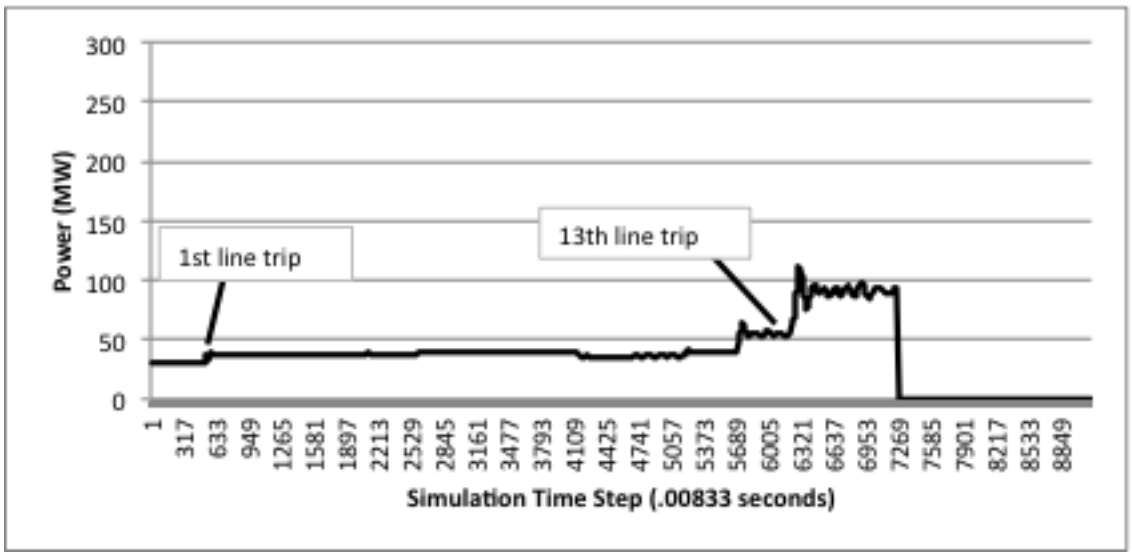


Figure 1.13 Simulated MW flow of Gypsy – Fairview 230 kV

The plot in Figure 1.12 is measured over the course of 10 hours and the plot in Figure 1.13 is measured over a simulation of 150 seconds. The relative distance between tie line tripping is different in the plots. However, the locations of the important events are labeled in both plots. The first important point to compare the two plots is when the 1st tie line trips. It can be seen that in both plots the loading on Gypsy-Fairview 230kV increases. The second important point to compare the two plots is when the 13th tie line is lost. At this point the flows in both plots are most similar. The power flow in the PMU data is approximately 115 MW and the power flow in the simulation is about 91 MW. Before the 13th tie line trips the flow levels in simulation do not match that of the recorded data. This is mostly likely because the simulation is designed to best recreate the conditions in the system just before island formation, whereas the SCADA data records the tie line operation up to 10 hours before the island formed. It is reasonable to assume that the simulation would not match conditions present in the system hours prior to island formation.

The original power flow and dynamic data received from Entergy was modified to more closely match the conditions present when hurricane Gustav hit the system. The simulations done using the modified power flow and dynamic data showed a reasonable match to the available historical data, and should represent the actual conditions of the system more accurately at the time of island formation.

1.4 Island Prediction Analysis

Ideally, the best way to determine and characterize signatures of an impending island formation would be to build a CART database of only real PMU measurements of island formations. However, this would require many different island formations captured by PMUs, and the island formed would need to always be in the same location. The real PMU data available for this study only contains one island formation, making this approach impossible. It was decided that the best alternative would be to build a CART database using simulated island formations. The island formed in the simulations would always be the same island that was formed during hurricane Gustav. In this way, any predicting signatures found by this study could be searched for in the real PMU data to determine whether or not the Gustav island formation could have had any advance warning.

1.4.1 Hurricane Isaac Cases and Simulations

Building a CART database of simulated island formations requires having several different operating conditions of the Entergy power system. To ensure any signatures found from this database can be applied to the Gustav event, it is important that the power flow cases used to build the database come from times around when a hurricane

has hit the system. Five power flow cases were provided by Entergy, that correspond to hurricane Isaac. Hurricane Isaac made landfall at 7:00 PM on August 28, 2012 near the mouth of the Mississippi River [5]. The five power flow cases correspond to the times; 10:30 AM August 28th, 12:00 PM August 28th, 12:00 PM August 29th, 6:00 PM August 29th, and 12:00 PM August 31st. Along with using the five different operating conditions, ten different orders of tie-line outages were used to create a total of 50 simulations to complete the CART database. The order of line outages that actually occurred during the Gustav event was included as one of the ten orders. The remaining nine orders were mostly random. However, the last two lines in each of the nine orders were intelligently selected to allow every line, at least once, to serve as either the last line or second to last line in the outage order. The simulations were conducted using PSS®E v33.3. As stated previously, the same island in each simulation was created using one of the five power flow cases and one of the ten tie-line outage orders, providing a total of 50 simulations. During each simulation, the bus values of frequency, voltage magnitude, and voltage phase angle, were recorded at each of the PMU sites at El Dorado, Mablevale, Waterford, and Ninemile. In each simulation, the 14 tie lines were removed at a rate of one every five seconds until the island formed. As an example, the exact actions taken during the simulation using the original Gustav line outage order is shown in Table 1.11.

Table 1.11 Actions Taken During Island Simulation

Time of Action	Bus Number(s)	Action Taken
0-5sec	--	Flat Line Run
5sec	336462-500360	Disconnect Line
10sec	336015-336016	Disconnect Line
15sec	336032-303202	Disconnect Line
20sec	336141-303204	Disconnect Line
25sec	336006-336007	Disconnect Line
30sec	335568-335660	Disconnect Line
35sec	335536-335665	Disconnect Line
40sec	335771-303200	Disconnect Line
45sec	335500-335618	Disconnect Line
50sec	335568-335659	Disconnect Line
55sec	335657-335658	Disconnect Line
60sec	303153-335456	Disconnect Line
65sec	335618-335837	Disconnect Line
70sec	336190-336138	Disconnect Line
90sec	--	End Simulation

1.4.2 Simulated Island Formation Results

Once all 50 simulations were complete, the simulation output file was converted to an Excel document to plot and review the results. After carefully studying the simulation results, several features were observed. The frequency data in each simulation did not seem to show any useful information. Both the voltage magnitude and voltage phase angle showed interesting characteristics, and the simulations appeared to create three categories. In the first category, a sudden change in difference in phase angle of at least 5º was observed during the removal of at least one tie line. In some simulations, a sudden

change in phase angle could be observed for the loss of several lines. A clear example of this can be seen in figure 1.14.

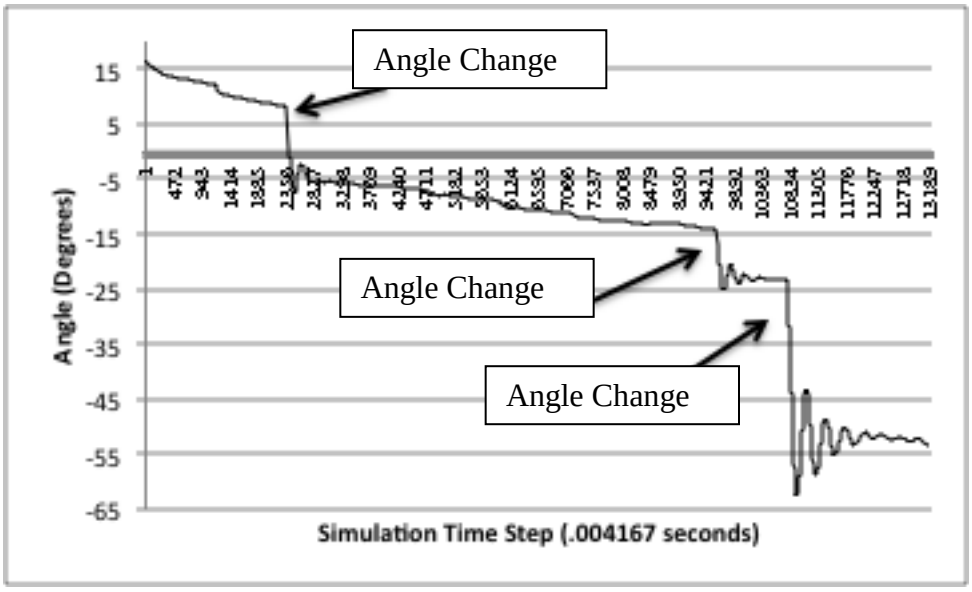


Figure 1.14 Voltage phase angle at Waterford versus time

In the second category, a sudden change in voltage magnitude in the islanded area of at least 5% was observed during the removal of at least one tie line. An example of this can be seen in figure 1.15.

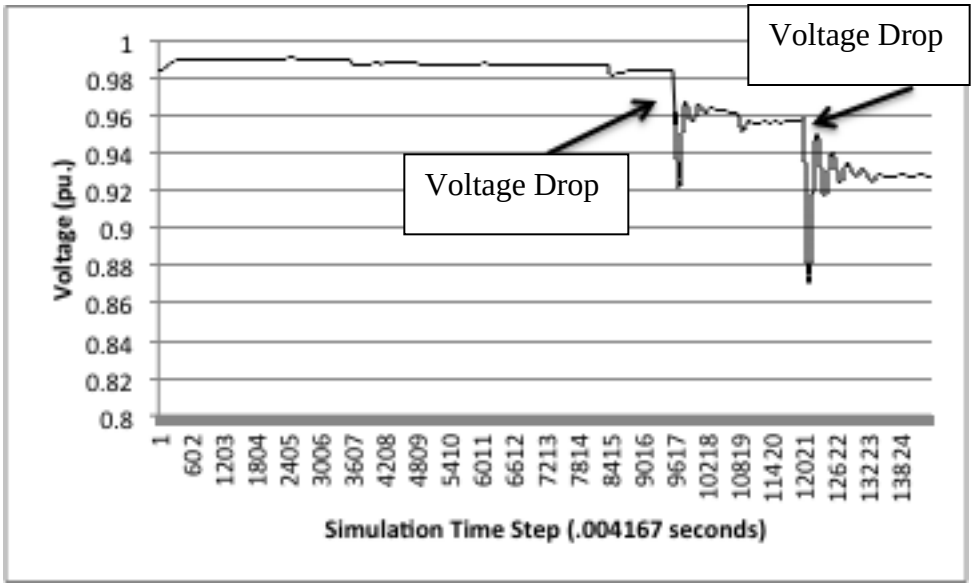


Figure 1.15 Voltage magnitude at Waterford versus time

In the third category, any sudden changes in voltage phase angle or voltage magnitude were either deemed too small to be seen by a PMU among normal disturbances in the system or no sudden changes were observed at all. An example of this can be seen in Figure 1.16 and Figure 1.17. The gentle slope in phase angle seen in Figure 1.17 is seen at all four buses and does not result in any change in phase angle difference between areas.

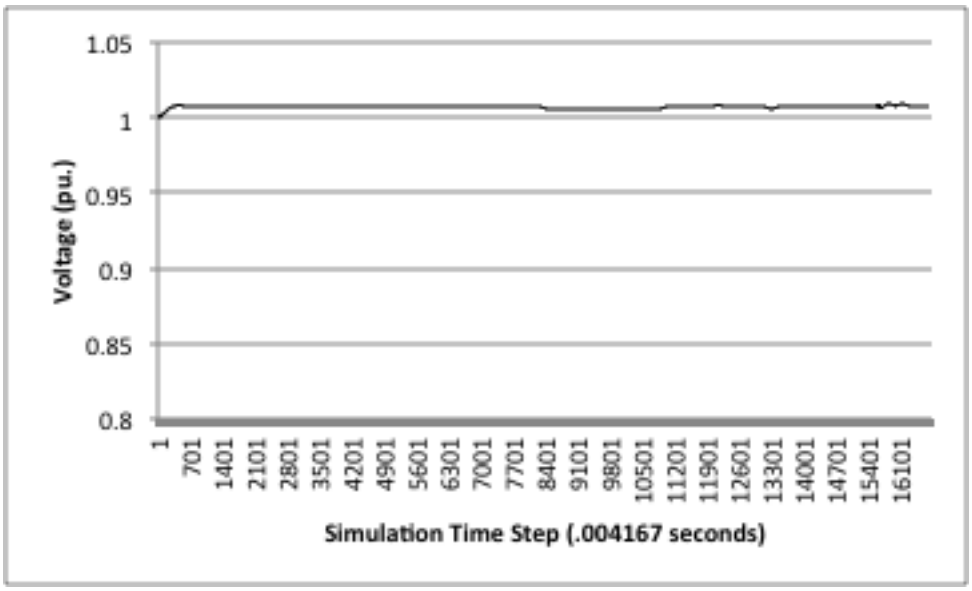


Figure 1.16 Voltage magnitude at Waterford versus time

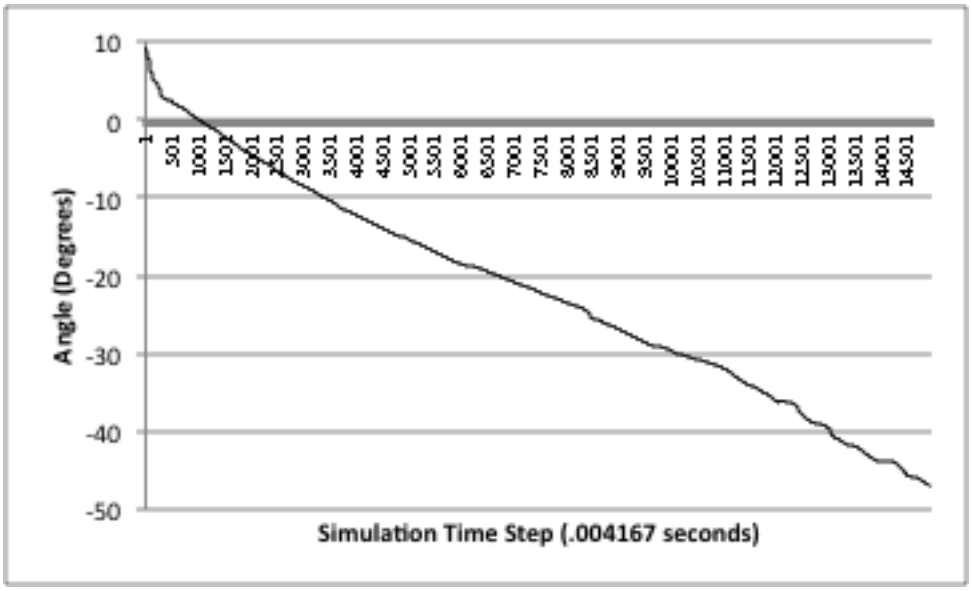


Figure 1.17 Voltage phase angle at Waterford versus time

After looking through all the simulations it was found that 43 of the 50 simulations fell into category one, 31 of the 50 simulations fell into category two, and five of the 50 simulations fell into category three. Some simulations qualified for both category one and category two simultaneously.

This information suggests that there is a correlation between changes in the PMU data and the loss of a tie line. This could be very important in predicting island formations. By using PMUs to monitor changes in tie line status, if the number of remaining tie lines is very low, a system operator could be notified that an island formation in a particular area is a reasonable threat.

1.4.3 Island Prediction CART Analysis

Before CART can train a decision tree, a CART database must be created using the island formation simulations. It is very important to assemble the CART database correctly. Setting up a CART database poorly would make CART look at the data in a way that has nothing to do with the problem being studied, and provide useless results. An example of the island prediction CART database structure can be seen in Table 1.12.

Table 1.12 Island Prediction CART Database Structure

Lines	**El V**	**M V**	**W V**	**N V**	**M A**	**W A**	**N A**	**El F**	**M F**	**W F**	**N F**
4	data	data	data	data	data	data	data	data	data	data	data
4	data	data	data	data	data	data	data	data	data	data	data
4	data	data	data	data	data	data	data	data	data	data	data
3	data	data	data	data	data	data	data	data	data	data	data
3	data	data	data	data	data	data	data	data	data	data	data
2	data	data	data	data	data	data	data	data	data	data	data
2	data	data	data	data	data	data	data	data	data	data	data
1	data	data	data	data	data	data	data	data	data	data	data
1	data	data	data	data	data	data	data	data	data	data	data

As a reminder, the quantities recorded during simulation are voltage (V), phase angle (A), and frequency (F). These values were recorded during simulation at the PMU sites at El Dorado (El), Mablevale (M), Waterford (W), and Ninemile (N). The reason El Dorado phase angle is not seen in the CART database is that the other phase angles were taken relative to the phase angle at El Dorado. Doing this causes the angle at El Dorado to always be zero and is not needed in the analysis. For each simulation, only the duration

of the simulation where only four tie lines (or less) are remaining is used in the database. It was decided to only use the data from the last four tie lines for a number of reasons. Firstly, the CART license being used for this study has a database size limit of 8 megabytes. Only using the data from the last 4 tie lines prevents the database from reaching this limit. Also, it was observed that most of the significant changes in recorded data occurred during the loss of the last few tie lines. Lastly, it is more important to identify when the system goes from having four tie lines to having only one or two lines remaining than the loss of earlier tie lines.

The column labeled "Lines" is the target variable. For each time-step, the data recorded is labeled corresponding to the number of tie lines still operating in the system. Each row of the database is independent of the other rows. Therefore, the order that the simulations are entered into the database is irrelevant. Setting up the database in this way will force CART to try to determine the number of tie lines remaining by only looking at the data that would be available from the PMU locations in the system. If there is a correlation between changes in measurements and the loss of tie lines, CART will be able to find and characterize it.

1.4.4 Island Prediction Decision Tree Testing

For the first CART analysis, two CART databases were constructed. Both databases used the same structure as that described previously. The first database contained the data from the first 25 simulations. This database serves as the decision tree training database. The second database contained the data from the second 25 simulations. This database serves as the decision tree testing database. By building a decision tree using the first 25 simulations and then testing the accuracy of the tree using

the second 25 simulations, CART will show whether or not a correlation between the simulation data and changes in tie line status exists.

The decision tree created by CART using the first 25 simulations contained 876 terminal nodes. The result of testing this tree using the remaining 25 simulations is shown in Table 1.13.

Table 1.13 Decision Tree Test Results

Actual Class	**Total Class**	**Percent Correct**	**1 Line**	**2 Lines**	**3 Lines**	**4 Lines**
1 Line	28846	47.5%	13703	6601	4756	3786
2 Lines	28824	18.62%	7255	5367	7403	8749
3 Lines	28824	26.35%	1994	7374	7594	11862
4 Lines	28824	62.96%	1649	3276	5751	18148

The most important results of this test are in the column labeled “Percent Correct”. Since there are four possible classes, if it were truly impossible to determine the number of remaining tie lines by looking at the available data, one would expect all the numbers in this column to be 25%. This would correspond to the decision tree just randomly guessing. For the classes 2 Lines and 3 Lines this seems to be the case, at 18.62% and 26.35% respectively. However, the classes of 1 Line, at 47.5%, and 4 Lines, at 62.96%, show significant improvement. It was not expected that CART be able to look at the available simulation measurements and determine exactly how many tie lines remain in the system at every time step. However, this does confirm the suspicion that there is a correlation between PMU measurements and changes in tie line status. It is important to point out that these results only used 25 simulations of training and there are only four PMU locations being measured. In the future, as more PMUs are placed in the system it is a reasonable assumption that the ability to determine tie line status from PMU measurements would vastly improve.

1.4.5 CART Signature Characterization

For the second CART analysis, a single decision tree was trained using all 50 simulations. It was already shown that there is a correlation between the simulation data and changes in tie line status. The purpose of this second decision tree is to give some idea of the rules CART uses to classify the data. Because the optimal decision tree using all 50 simulations has over 800 terminal nodes, the decision tree was pruned down to 5 terminal nodes. Pruning a decision tree reduces the size of the tree by removing the less significant splitting rules. This causes terminal nodes to combine together and reduce the size of the tree. Pruning the decision tree will reduce the overall accuracy of the tree, but also highlights the most significant attributes used for classification. The pruned decision tree using all 50 simulations can be seen in Figure 1.18.

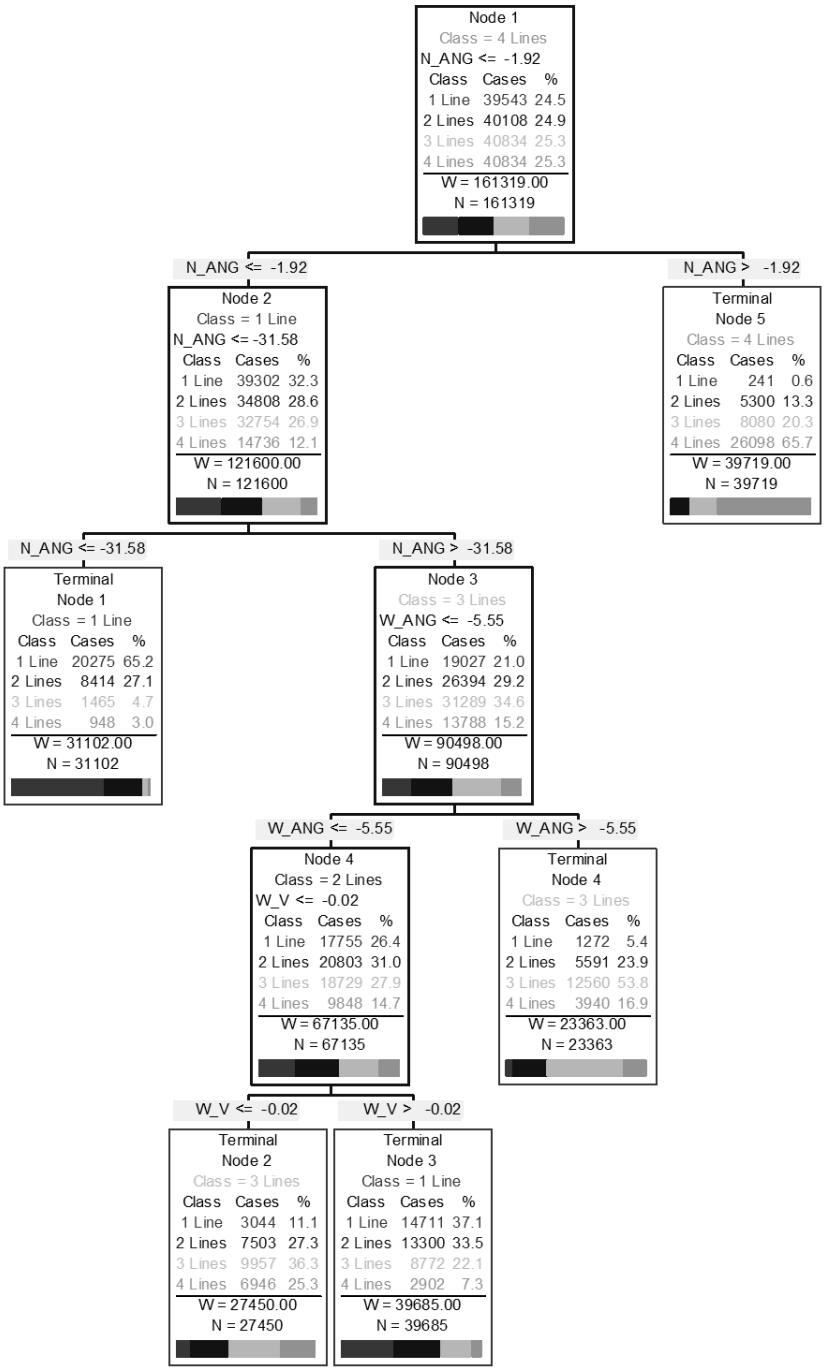


Figure 1.18 Pruned CART decision tree

The decision tree created by CART contains 5 terminal nodes. As a reminder, all phase angles in the CART database are relative to the phase angle at El Dorado. The rules required for data to reach each terminal node are as follows:

- Terminal Node 5:
    - Label: 4 Lines
    - Conditions to reach: Ninemile angle > -1.91º
- Terminal Node 4:
    - Label: 3 Lines
    - Conditions to reach: Ninemile angle >-31.58º and Ninemile angle <= -1.91º and Waterford Angle > -5.55º
- Terminal Node 3:
    - Label: 1 Line
    - Conditions to reach: Ninemile angle >-31.58º and Ninemile angle <= -1.91º and Waterford Angle <= -5.55º, and Waterford voltage has reduced no more than -.0192 pu
- Terminal Node 2:
    - Label: 3 Lines
    - Conditions to reach: Ninemile angle >-31.58º and Ninemile angle <= -1.91º and Waterford Angle <= -5.55º, and Waterford voltage has reduced more than -.0192 pu
- Terminal Node 1:
    - Label: 1 Line
    - Conditions to reach: Ninemile angle <= -31.58º

This decision tree contains several interesting aspects. First, frequency is not used to classify data in the pruned decision tree. This agrees with what was observed after plotting the simulation results. Also, when the phase angle at Ninemile is close to the phase angle at El Dorado, the decision tree determines there are 4 tie lines operating. When the phase angle at Ninemile is more than 31.58º behind the phase angle at El

Dorado, the decision tree determines that there is only 1 tie line remaining. This suggests that large changes in voltage phase angle between areas are important in identifying the loss of a tie line.

1.4.6 Signatures in Gustav Island Event

It was shown previously that a sudden change in difference in phase angle between two areas is correlated to a loss of a tie line that connects said areas. Also, in some island simulations the loss of a tie line caused a significant change in voltage in the islanding area. Because the islands created in simulation were always the same island that was formed during hurricane Gustav, these signatures can be searched for in the real PMU data. The island formed during hurricane Gustav had 14 tie lines that connected it to the rest of the system. The time that each of these lines went offline was recorded during the event. Any signatures of interest contained in the PMU data must be at these locations.

The real PMU voltage magnitude and voltage phase angle data were studied at the times were tie lines were removed from the system. The last two lines to go offline were Coly-Willow Glen 500 kV (2^{nd} to last) and Gypsy-Fairview 230 kV (last). The first period of interest was the location where the 2^{nd} to last tie line went offline. The PMU data at this point showed a signature in the phase angle measurements, but nothing was found in the voltage measurements. A plot of the voltage phase angle difference between Ninemile (inside island) and Sterlington (outside island) can be seen in Figure 1.19 and a plot of the voltage phase angle difference between Waterford (inside island) and Sterlington (outside island) can be seen in Figure 1.20.

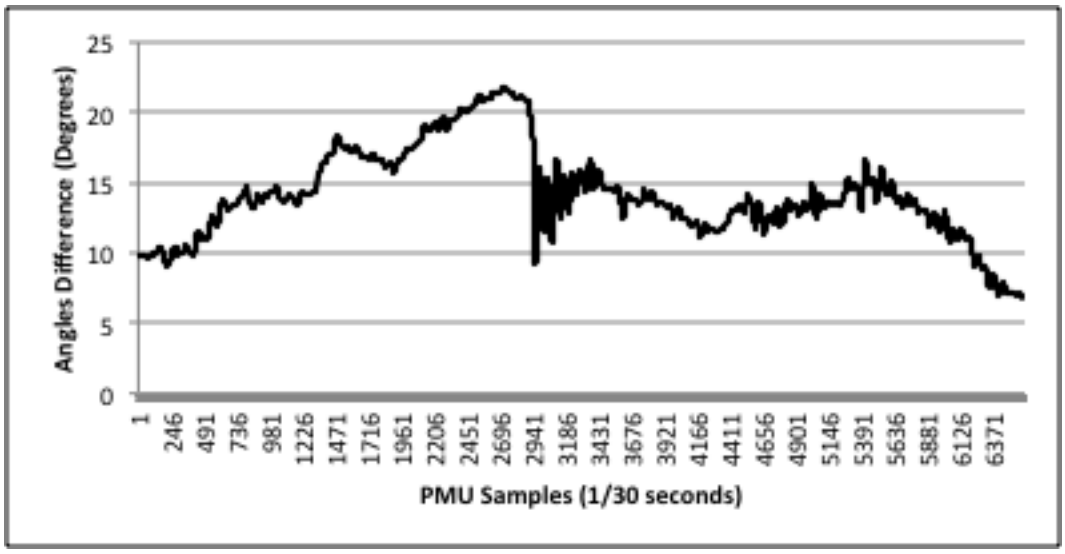


Figure 1.19 Ninemile-Sterlington phase angle difference versus time

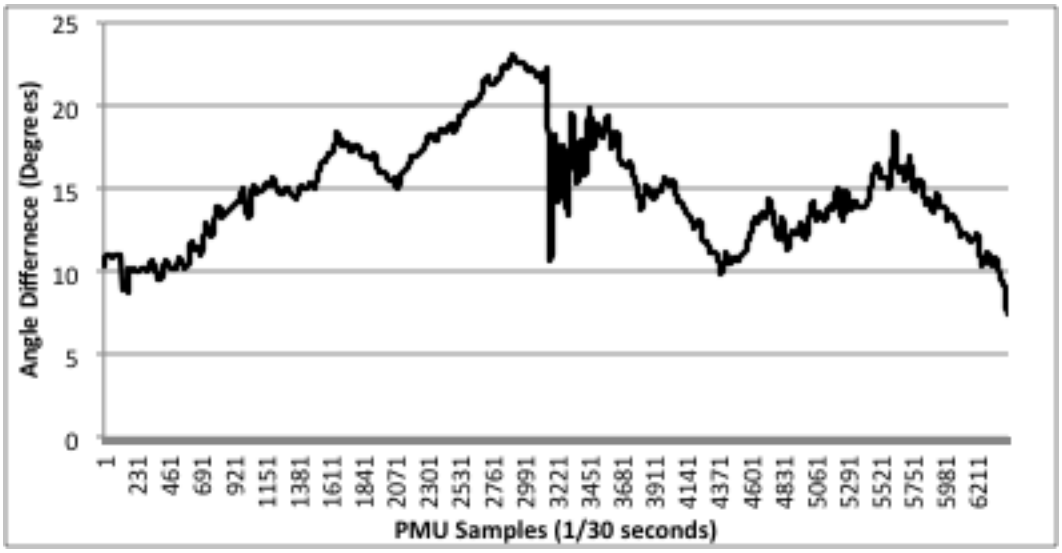


Figure 1.20 Waterford-Sterlington phase angle difference versus time

At the moment the 500 kV line was lost, the PMU data observed a ~12º change in the difference between phase angles that stands out from the rest of the data. This is consistent with what was seen in the simulated island formations. Observing this signature at this instant is very important because this line was lost 38 minutes before the

island was formed. This signature was a warning sign that an island formation in the nearby area was a reasonable threat and that the area should be closely monitored.

Other points where lines were lost in the PMU data were reviewed, but only one other disturbance was found. The Webre-Willow Glen 500 kV line was the 6th to last line to be lost before the island formed. A plot of the voltage phase angle difference between Waterford (inside island) and Sterlington (outside island) can be seen in Figure 1.21.

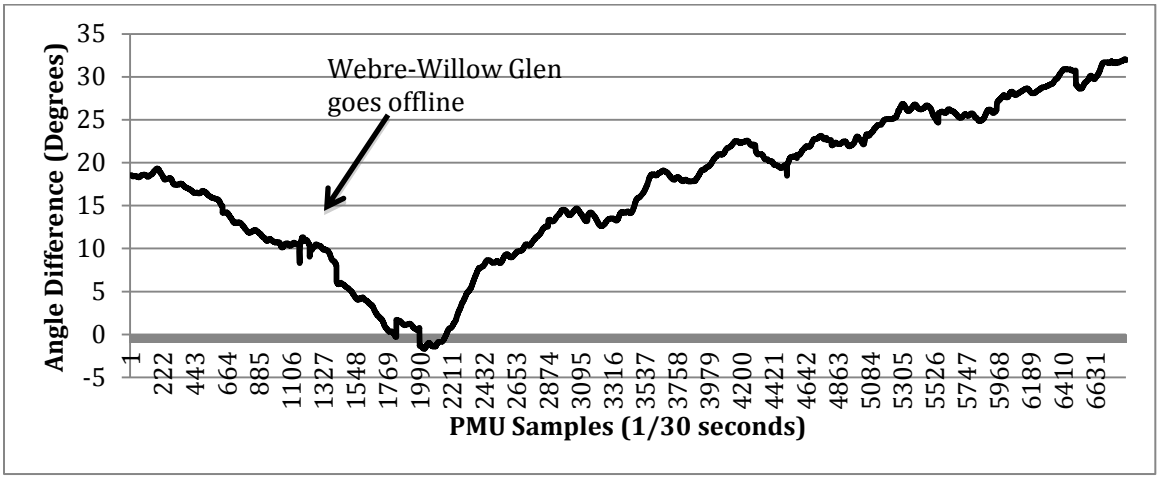


Figure 1.21 Waterford-Sterlington phase angle difference versus time

The change in phase angle when this line was lost was only about 3º. It is very unlikely that this could have alerted operators to what was going on. However, it is important to show that it is visible, and if the operating conditions in the system had been different at the time this may have been much more significant.

1.5 Conclusions

This study aimed at using real PMU measurements to predict and detect significant system events especially islanding with the help of the data-mining tool CART. The PMU data offered by Entergy, containing the island formation event when hurricane Gustav impacted the system, provided an excellent case for this study. During the storm,

14 tie lines were lost that created an island containing Baton Rouge and New Orleans. Careful analysis was conducted to determine whether or not the island could be detected by only using the PMU measurements. It was found that the most effective approach of identifying the creation of the island was to use the PMU measurements of voltage phase angle. By comparing the phase angle measurements between PMUs, in this case, the island could have been detected in approximately 4 seconds. Also, by comparing different sets of PMUs, the location of the island could be determined by which PMUs were inside or outside of the affected area. Because this approach only considers the PMU measurements to form conclusions, the same method could be applied to any system containing PMUs, with only slight modification, and still provide the ability to quickly and reliably detect the formation of an island within the system.

Provided with the system power flows and dynamic data corresponding to the time when hurricane Gustav entered the system, simulations were conducted to attempt to recreate and match the event to the historical PMU data. Load and generation levels across a wide range of the system were adjusted to closely match the phase angle difference see in the PMU data. Next, the conditions inside the island were adjusted using the known generator dispatch and the available SCADA data. It was found that the direction of the power flowing on the last tie line must have been opposite to the SCADA data. Also, it was found that in order to match the simulation to the PMU frequency measurements, the governor reference at one of the generators must have been reduced just following the creating of the island. Doing these things allowed the event to closely match to the PMU measurements and provide a better understanding of what happened just after the island formed.

Lastly, the PMU data was used to try to predict the island formation. With not enough data to search for signatures by using the one island formation in the PMU data, 50 simulations were conducted to build a CART database. The simulations were analyzed intuitively and with CART to determine any predicting signatures. It was found that there is a strong correlation between a sudden change in voltage phase angle and the loss of a tie line. A number of simulations also showed a sudden change in voltage within the island area after the loss of a tie line. These different signatures were searched for in the real PMU data at the times when tie lines were reported to have been removed from the system. It was found that when the second to last tie line went offline, there was a 12º change in phase angle measured inside the island. This signature precedes the island formation by 38 minutes and could have alerted system operators that this area needed attention.

This study was successful at using CART, along with a strong knowledge of power systems, to analyze PMU data from a historic event. The data-mining tool CART helped quantify and understand the phenomenon observed in the PMU data. The method of identifying an island formation using voltage phase angle measurements is both effective and reliable, and could be used in real applications. The signatures found to predict the island formation is much less reliable. Large changes in load or generation could also create a sudden change in phase angle and the method could be prone to false alarms. This method of island formation prediction could likely be improved by pairing it with additional information, such as SCADA data. However, this study only considers the information that can be drawn from the PMUs alone. In the future as more PMUs are

placed in the power system, it is a reasonable assumption that the predicting signatures found in this study will be easier to identify and provide more information.

2 Data Mining to Characterize Impeding Oscillatory and Voltage Stability events

2.1 Introduction

2.1.1 Problem Statement

Several electrical utility companies are installing large numbers of phasor measurement units (PMUs) to monitor system conditions. In addition, several utilities also have collected a significant amount of historical PMU data. These sets of stored data also include measurements obtained during known events occurring on the system. At the PSERC summer workshop in Maine, and as noted in the 2010 research solicitation, there is a definite need to identify signatures of impending events detrimental to system performance from PMU measurements.

From the control center operators' point of view, the fast assessment of power system oscillatory stability and voltage stability is of great importance for real-time operation. It is desirable that the impending system events can be immediately detected and that operators are provided with updated information on whether or not a power system can maintain synchronism and acceptable voltage levels when subject to disturbances.

Traditionally, the method of time-domain simulation is used to analyze system stability status [6]. However two obstacles prevent the traditional method's application in real-time monitoring and control. Firstly, full system model computation makes the simulation method time-consuming. Considering the fast onset of an instability event, the traditional methods may not be able to provide immediate event detection. On the other

hand, using a simplified system model could accelerate the simulations, but this brings concern over approximate analysis results leading to inaccurate decisions. Secondly, the data used for the stability analysis in electrical utilities are obtained from the supervisory control and data acquisition (SCADA) system or state estimation functions, which are refreshed on a time scale from several seconds to several minutes. Figure 2.1 shows the state-of-the-art data acquisition structure and its possible implementation in analyzing two types of power system stability status, i.e. oscillatory stability and voltage stability. In addition, the SCADA measured data does not have the characteristics needed to implement the new analysis and control tools due to the lack of time-synchronized sampled waveform data [7]-[8]. Compared to a traditional SCADA system, synchrophasor IEDs such as PMUs enable a much higher data sampling rate and provide the synchronized phasor measurements across the network.

In some cases the forecasted load pattern and unit commitment dispatch are used instead of actual data to predict system performance. When a disturbance occurs and immediate controls need to be initiated, traditional stability analysis using slowly updated or forecasted data can only provide very limited decision making support.

To make the situation worse, in power system planning and on-line applications, a complete model may not be readily available. This model is necessary for obtaining the linearized system description required by traditional oscillatory stability analysis [9]. Similar problems exist in the voltage stability assessment process [10]. Under such circumstances, the data mining techniques, benefiting from accurate generalization ability without detailed knowledge of all system parameters, becomes an attractive alternative.

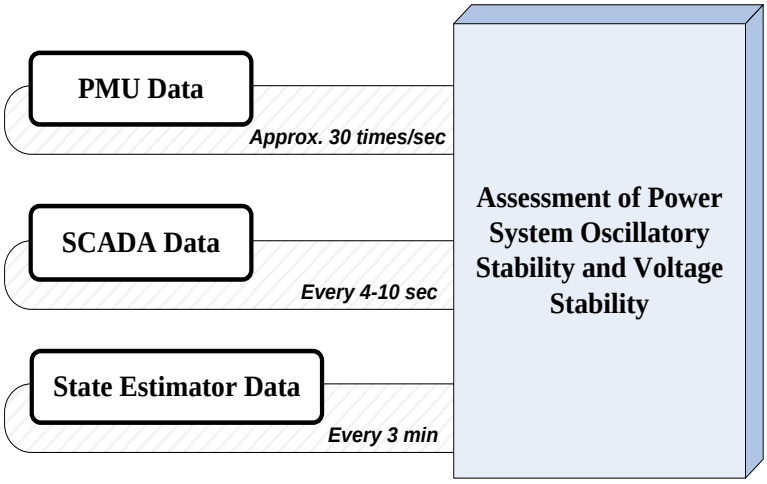


Figure 2.1 Power system stability analysis using data from various sources

2.1.2 Project Objectives

The objectives of this project are summarized as follows:

- Investigate the use of data mining tools to examine historical PMU measurements and develop decision trees (DTs) to characterize signatures for identifying and preventing future events or failures;

- Evaluate the performance of CART (Classification and Regression Trees), [11], algorithm for processing synchrophasor measurements;

- Evaluate other available data mining tools and analyze the ability of these tools to characterize signatures of impending systems events or detrimental system behavior;

- Consider the use of PMU measurements from multiple locations;

- Verify the performance of data mining tools by comparing the results obtained for measurements corresponding to known events on the system.

The commercial data mining software CART developed by Salford Systems, [12], is employed in this project. It allows users to analyze data from many different dimensions or angles, categorize them, and summarize the relationships identified.

2.1.3 Literature Review

In the field of power systems, Wehenkel et al. first introduced the DT method to solve the transient stability assessment problems using SCADA data [13]-[14]. In [15]-[19], DTs were successively applied to assess system operational security by applying a pre-defined set of credible contingencies and enforcing an acceptable threshold criterion on system variables based on standard operating practices. Later, in [20], the system post-disturbance stability has been analyzed by DT using its fast evaluation capability. In [21], a genetic algorithm was applied in feature selection to search for the best inputs to DT for oscillatory stability region prediction. In [22] and [23], Kamwa et al. showed that there is a trade-off between a data mining model's accuracy and its transparency. A review of literature reveals that the problem of using DT for stability margin monitoring from substation field measurements has not yet been fully explored.

The concept of decision tree comprises the Classification Tree and Regression Tree, [11]. While in previous works classification trees have been extensively studied to group an operating point (OP) into one of several pre-defined stability categories, the use of regression trees (RT) to predict the stability margin, i.e. how far the system is away

from a possible instability event, has not yet been fully studied. With respect to its online use, the areas that remain unexplored include how fast the RT can process PMU measurements, how well the RT can deal with measurement errors, and how robust the RT is to the system topology changes. It is also imperative to develop a systematic approach to generating a sufficient and realistic knowledge base for off-line training of DT.

Several other data mining tools such as Neural Networks, [24], and Support Vector Machines, [25], have been used to evaluate the system stability status. Compared with some "black-box" tools, the DT's piece-wise structure provides system operators with a clearer cause-effect relationship of how the system variables lead to the onset of an instability event. Using DTs it is possible to identify the critical variables and thresholds that need to be analyzed to gain insight into the stability margin of a system.

2.1.4 Proposed Research

As shown in Figure 2.2, both the model-based and measurement-based (depending on whether the system model data is used) methods will be explored in this project. For the model-based approach, a knowledge base will be created through exhaustive simulations on known system model parameters and then utilized to train the decision trees. In situations where detailed model parameters are missing, a measurement-based approach using data mining and signal processing techniques will be explored to estimate system stability status directly from the synchrophasor measurements collected at the

PMU-equipped substations. The efficacy of the measurement-based approach is going to be tested using field PMU measurements.

In addition, the following two issues are involved in the proposed research:

- Performance comparison between the DT and other data mining tools;
- Robustness test of the DT predictive model.

In particular, the relationship and difference between the conventional time-domain simulation approach and the proposed data mining technique, the DT method, is shown in Figure 2.3. Compared with the traditional method, the advantage of DT method lies in its capability of fast analysis facilitated by fewer required inputs and straightforward model structure. By learning the system behavior from a known set of operating points (OPs), the DT model can predict system responses without detailed model computations. In addition, the DT method is appealing because it uses a white-box model, which makes the results easy to interpret. Based on the combination of splitting rules along a path of the tree, preventive and corrective control strategies could be formulated.

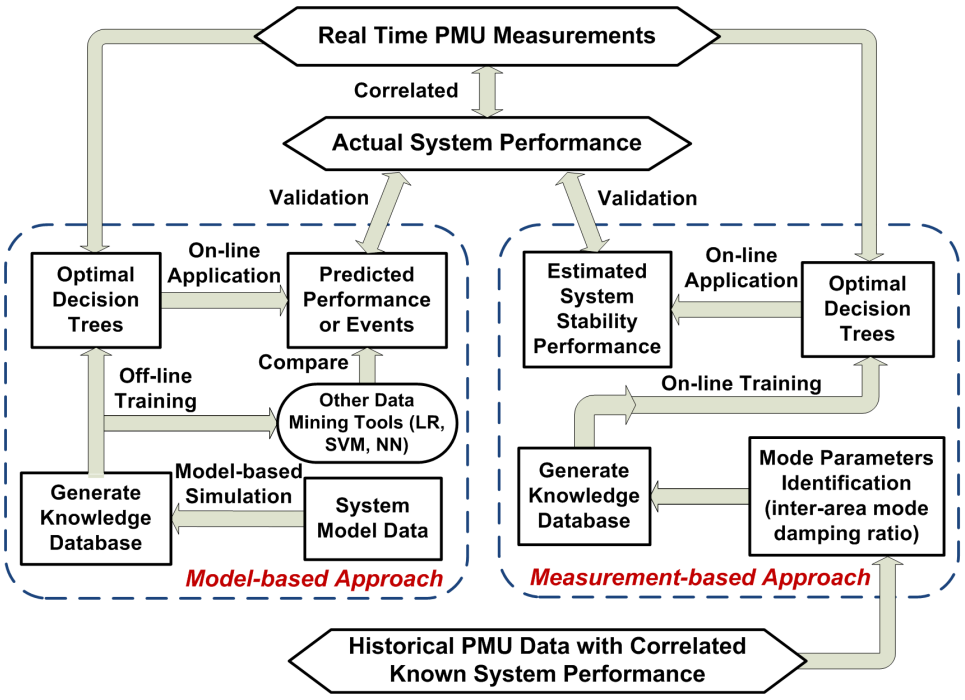


Figure 2.2 Proposed research framework

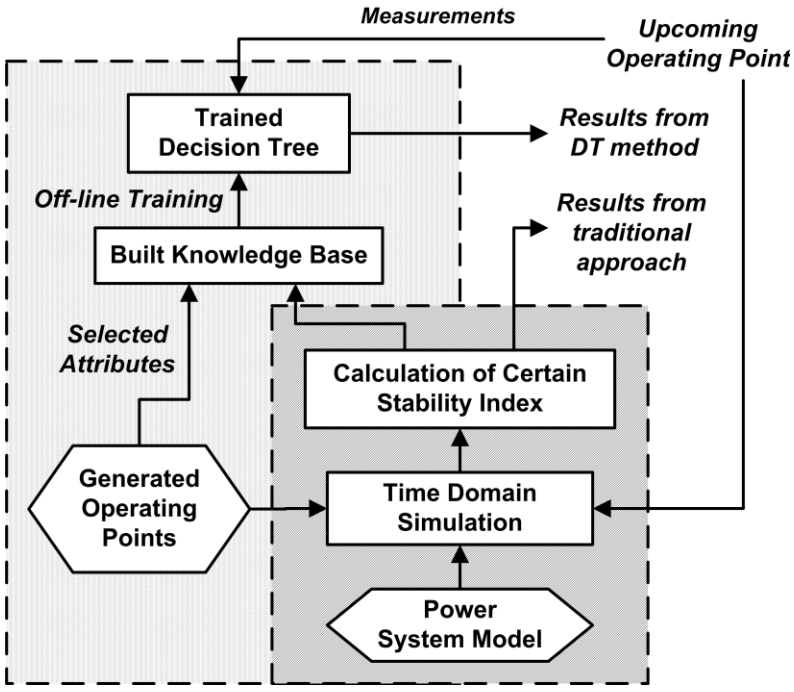


Figure 2.3 Difference between conventional approach and the DT method

A specification of the proposed research is as follows:

- Develop a methodology that makes use of the PMU collected synchrophasor measurements for online stability estimation and early detection of impending system instability events;

- Examine the prediction accuracy and robustness of DT for online assessment of system oscillatory stability and voltage stability status. The comparison in accuracy and efficiency between the DT and other data mining tools such as Support Vector Machine (SVM) and Neural Networks (NN) will be compared;

- The important issue of DT robustness with respect to PMU measurement errors and changes in system topology will be explored;

- Develop a methodology for optimal PMU placement. Check the performance of DT using synchrophasor measurements from a limited number of PMUs;

- Develop an approach that takes use of active learning technique to reduce the computational burden of both simulation and training by selecting the most effective training dataset;

- Explore a measurement-based approach that directly applies data mining and signal processing techniques to field PMU measurements.

2.2 Technical Background

2.2.1 Introduction

The proposed assessment scheme (using DTs on PMU measurements) is shown in Figure 2.4 and compared with existing analytical method (using model simulation and SCADA data). As mentioned before, classification and regression trees are trained to emulate system behavior and predict system stability status. An abnormal operating point with an insufficient stability margin can be immediately identified. Compared with the traditional time-domain simulation approach that requires full model computation each time a new OP emerges, the DT method is faster since repetitive model computations are avoided.

2.2.2 Theoretical Formulation

Two important aspects of system operational performance, namely oscillatory stability and voltage stability, are targeted for monitoring. First the definition of an instability event is revisited:

- Oscillatory stability is related to Hopf bifurcation. An instability event occurs whenever, following a small disturbance, the damping torques are insufficient to bring the system to a steady-state operating condition which is identical or close to the pre-disturbance condition [9].

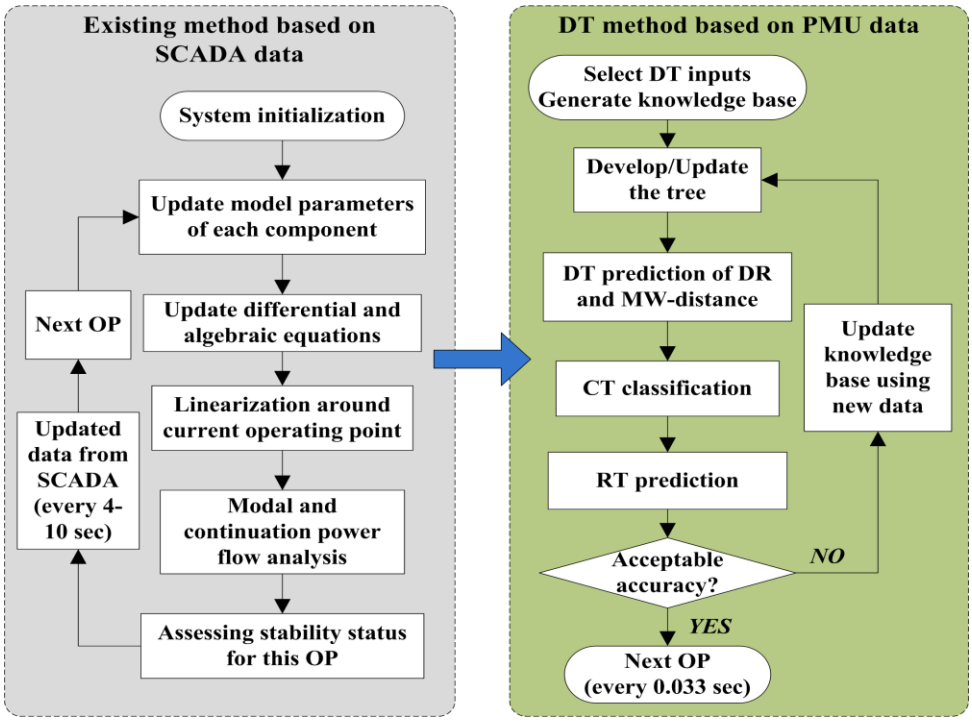


Figure 2.4 From time-domain simulation to the proposed scheme

- Voltage stability is related to saddle-node bifurcation. Voltage instability occurs when the load attempts to step beyond the capability of the combined transmission and generation system [10] [26] [27].

2.2.2.1 Oscillatory Stability Assessment (OSA)

Modern power systems have evolved into systems of increasingly large size. Initially separate systems have been interconnected. Different areas with larger generation capacity and inertia are added. Due to the deregulation and the difficulty of transmission expansion today, system operators are often forced to operate the system close to its stability limits, which leads to the recurrence of small-signal oscillation problem. As a consequence, in large interconnected power systems small signal stability, especially inter-area oscillatory stability, become increasingly important.

The oscillatory stability may be analyzed by modal analysis. A power system can be described as a set of non-linear differential algebraic equations (DAE):

$$\begin{cases} \dot{x} = f(x, y, u) \\ 0 = g(x, y, u) \end{cases} \tag{2.1}$$

where x is the state vector, y is the algebraic vector, and u is the input vector. The DAEs are formulated by detailed modeling of each network component. By linearizing the non-linear equations in Eq. 2.1 at a particular system operating point, the following equations are derived:

$$\begin{cases} \Delta\dot{x} = A\Delta x + B\Delta u \\ \Delta y = C\Delta x + D\Delta u \end{cases} \tag{2.2}$$

The matrices A, B, C, and D in Eq. 2.2 provide a linearization around the system equilibrium point. Each pair of complex conjugate eigenvalues of matrix A corresponds to an oscillation mode of the system. The A matrix can be further decomposed as:

$$A = \Phi \Lambda \Psi \tag{2.3}$$

In Eq. 2.3, Λ represents the diagonal eigenvalue matrix, and Φ and Ψ are left and right eigenvector matrices respectively. For the i^{th} oscillation mode with the following conjugate eigenvalues:

$$\lambda_i = \sigma_i \pm j\omega_i \tag{2.4}$$

The oscillation frequency is given by:

$$f_i = \omega_i / 2\pi \tag{2.5}$$

The mode damping ratio (DR) can be calculated by:

$$\xi_i = \frac{\sigma_i}{\sqrt{\sigma_i^2 + \omega_i^2}} \tag{2.6}$$

The oscillation modes that carry a significant amount of energy, but with insufficient DR, are critical among all modes and need to be closely monitored. Occurrence of an instability event is possible when a poorly damped mode is excited by a small or large disturbance.

In this work the DR of the critical oscillation mode is used as the oscillatory stability margin (OSM) indicator. Assuming DR_{crit} is the damping ratio of the critical mode, the scheme shown in Figure 2.2 is proposed for OSA. As shown in the figure, the OSM becomes progressively more stringent as the value of critical mode DR decreases.

The damping ratio is not an index from the parameter space, so strictly speaking it may not be proper to term it as 'margin'. In this work DR is selected as the OSM indicator in the sense that it provides smooth movement trajectory, a clear partition between stable/unstable states, and an explicit distance from unstable point.

As shown in Figure 2.5, three oscillatory stability states, namely "Stable" (including "Good" and "Fair"), "Alert" and "Unstable', are defined according to the value of DR_{crit}. A classification tree (CT) is used to assign a system operating point (OP) into one of the above stability states.

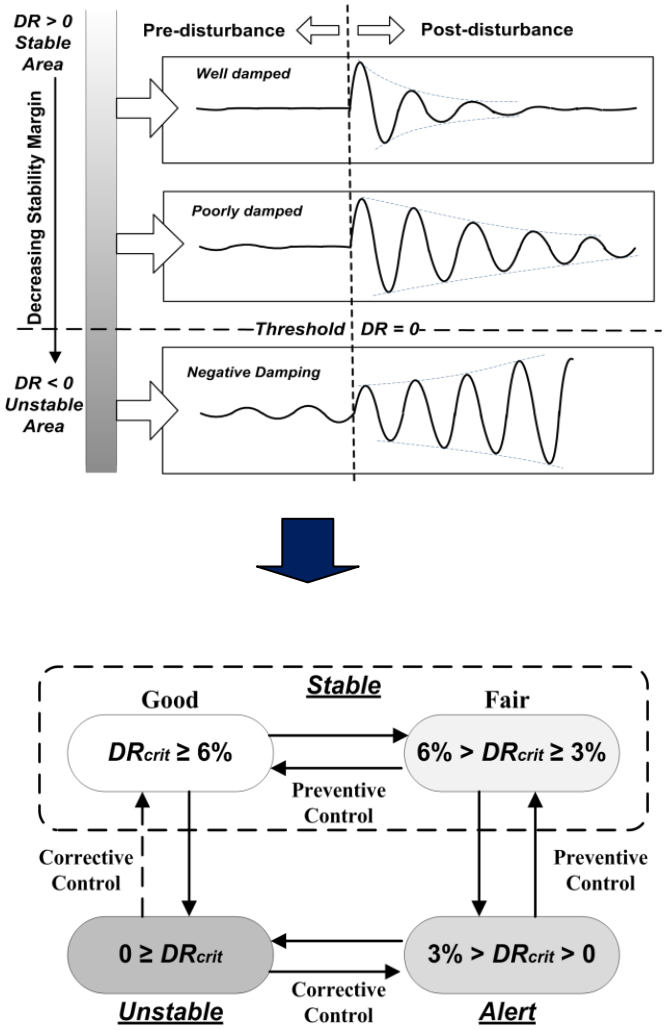


Figure 2.5 Proposed oscillatory stability assessment scheme

2.2.2.2 Voltage Stability Assessment (VSA)

The variation of load bus voltage magnitude with different load demand is plotted as the P-V curve shown in Figure 2.6. The MW-distance from the current operating point to the voltage collapse point ('Knee' point), where the load demand equals the maximum deliverable power, provides a reasonable measure of system voltage stability margin. The

VSM referred to here corresponds to system long-term voltage stability [6], which cannot be used to capture the short-term voltage stability.

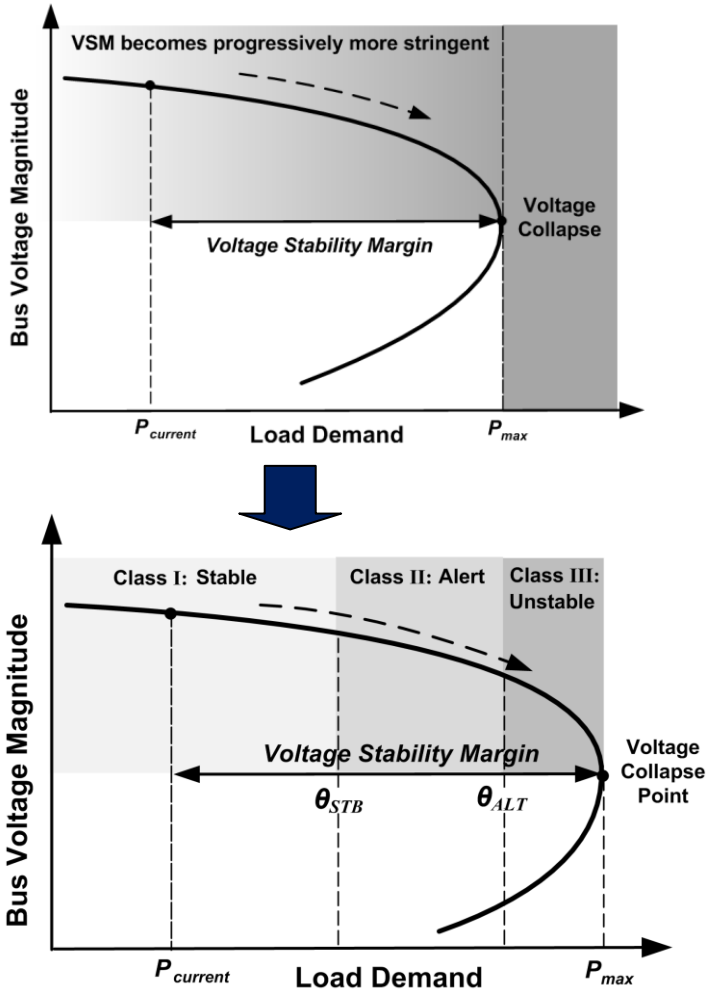


Figure 2.6 Proposed voltage stability assessment scheme

The focus is to find the voltage collapse point. In this work the idea of Continuation Power Flow (CPF) proposed in [28] is applied. Assuming a constant load power factor, slowly increasing load demand will push the operating point from the base case towards the collapse point along the P-V curve. The voltage collapse point is achieved when the load flow Jacobian becomes singular. System voltage stability margin is hereby expressed as:

$$MW_{\text{distance}} = P_{\max} - P_{current} \tag{2.7}$$

where P_{max} is the maximum deliverable power, and $P_{current}$ is the active load demand of current OP. The proposed procedure for voltage stability margin prediction is as follows:

(a) Generate *n* different OPs

(b) For each OP, determine the maximum deliverable power by means of the CPF technique

(c) Calculate the voltage stability margin for the i^{th} OP using the following index:

$$VS^{i}_{margin} = \frac{MW^{i}_{distance}}{P^{i}_{\max}} \times 100\% \tag{2.8}$$

(d) Train the RT off-line using selected features from the *n* OPs and their corresponding VS^{i}_{margin}

(e) Use the trained RT to predict VSM in real time

As shown in Figure 2.6, for the given voltage stability thresholds $\boldsymbol{\theta_{STB}}$ and $\boldsymbol{\theta_{ALT}}$ ($\boldsymbol{\theta_{STB}} > \boldsymbol{\theta_{ALT}}$), OPs will be labeled as "Stable" as long as they satisfy $VS^{i}_{margin} \geq \boldsymbol{\theta_{STB}}$; and 'Unstable' when $\boldsymbol{\theta_{ALT}} \geq VS^{i}_{margin}$. The remaining OPs are labeled as "Alert".

2.3 Model-based Approach for Real-Time Stability Assessment Using Classification Tools

2.3.1 Categorization of Stability States

In this work the data mining classification tools, in particular the Classification Tree (CT) method, have been adopted to estimate the system operating stability in real time.

As is shown in Figure 2.5 and Figure 2.6, several stability categories have been defined for oscillatory and voltage stability respectively. These states are specified according to the value of corresponding stability indicator.

2.3.2 Approach to Generating Training Database

The knowledge base is a database used for off-line training of the CT-based predictive model. It is composed of a number of instances, and each instance represents a system operating point, labeled with corresponding stability states, [29]. Our preliminary research revealed that the larger the system is, the more attributes, and more instances, are needed to characterize the OP using CT. These attributes comprise voltage and current phasors, active/reactive power flow, and some composite attributes.

Typically, the DT-based predictive model will gain more generalization power if a larger number of instances are included in the knowledge base. However, the database generation process should be correctly designed; otherwise it may not capture sufficient information from the entire problem space.

Both the voltage stability and oscillatory stability are closely related to the load/generation composition of a power system, and their increase/decrease trend at a

certain system snapshot [30]. If the load/generation composition varies, different OPs are formed. The change in the load demand and generation output can be described as:

$$P_G = P_G^0 + \Delta P_G \qquad Q_G = Q_G^0 + \Delta Q_G$$

$$P_L = P_L^0 + \Delta P_L \qquad Q_L = Q_L^0 + \Delta P_L \times Q_L^0 / P_L^0 \tag{2.9}$$

where P_G and Q_G are active/reactive power outputs of all the generators except the slack bus generator, and P_L and Q_L are vectors of active/reactive power delivered to the loads. Superscript 0 represents the base case OP. The vectors ΔP_G, ΔQ_G, ΔP_L and ΔQ_L stand for the variations in power.

In this work, the commercial software PSS/E [31] is used for iteratively solving load flows, and deriving the characteristic matrix A at different OPs through numerical perturbation. Python and MATLAB [32] programs are developed to automate the PSS/E simulations, perform modal analysis, conduct the CPF-based voltage stability analysis, compute stability margins, and establish the knowledge base. The pseudo-code for knowledge base creation is illustrated below.

2.3.3 Features Available to CT for Prediction

With respect to the input attributes of a decision tree, it is reported that different attribute combinations may result in different data mining accuracies [33]. In order to accelerate the prediction process, it is desirable to use the least number of attributes as CT inputs while keeping an acceptable level of overall prediction accuracy. Typically the input attributes are selected using engineering insight and empirical evidence.

In this work we consider the basic measurements from a PMU. The involved CT

input attributes are as follows:

- VM_i and VA_i: positive sequence voltage magnitude and phase angle at Bus i
- IM_i_j and IA_i_j: positive sequence current magnitude and phase angle from Bus i to Bus j

The commercial software CART [12] is used to develop CTs for evaluation of oscillatory stability and voltage stability.

2.3.4 Performance Examination of Classification Tree

2.3.4.1 Description of Test Systems

Two test systems, namely the IEEE 3-machine 9-bus system [34] and the IEEE 10-machine 39-bus system (New England system) [35], are used to implement the proposed scheme. The one-line diagrams of these two test systems are shown in Figure 2.7.

2.3.4.2 Knowledge Base Preparation

Using the previously described approach, the knowledge bases for the two test systems are generated and summarized in Table 2.1.

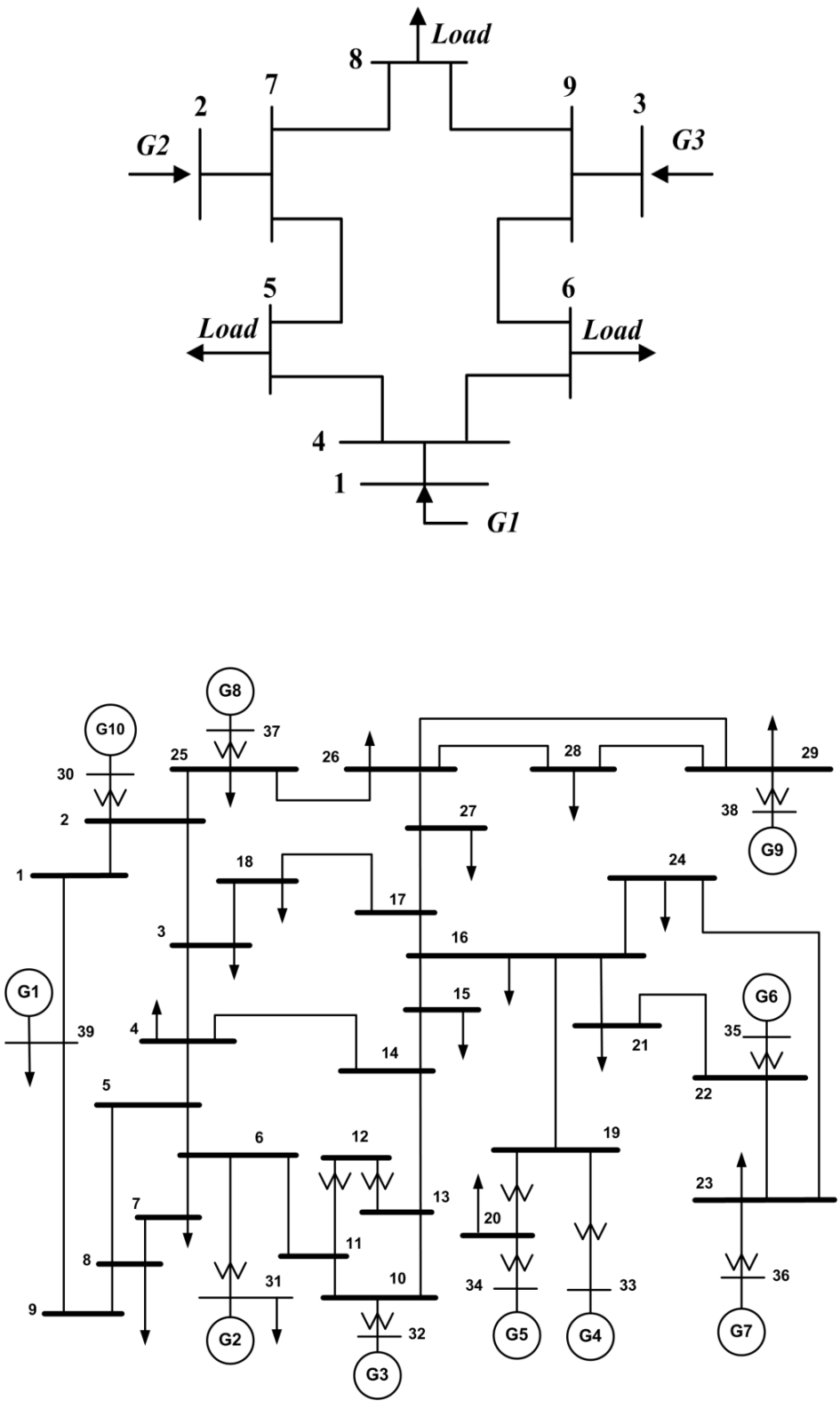


Figure 2.7 One-line diagrams of the IEEE 9-bus and 39-bus test systems

Table 2.1 Knowledge Base Generated for Classification Analysis

System	Instances included in OSA Knowledge Base			Total
	Stable	Alert	Unstable	
9-Bus	663 (61.90%)	358 (33.43%)	50 (4.67%)	1071
39-Bus	2549 (71.30%)	962 (26.91%)	64 (1.79%)	3575
	Instances included in VSA Knowledge Base			
9-Bus	707 (51.68%)	495 (36.18%)	166 (12.13%)	1368
39-Bus	2206 (60.21%)	1175 (32.07%)	283 (7.72%)	3664

2.3.4.3 Adjustment of Priors and Selection of Attributes

It can be observed from Table 2.1 that the number of instances included in each class is highly unbalanced. Compared with some other data mining tools that do not perform well when dealing with unbalanced data sets, the classification tree integrated in CART has the ability to assure that every class will be treated equally regardless of its size. This is achieved by specifying the *Priors* for each class. In this work the *Prior* for the "Unstable" class has been adjusted to be slightly higher than that of other classes. The objective is to put more emphasis on the detection of unstable instances.

2.3.4.4 Performance Comparison Using Different Tree Growing Methods

The theoretical background of developing a CT in CART can be found in [11]. Each of the above generated knowledge bases has been randomly split into two data sets: 80% of the instances are used as training set; the remaining 20% serve the purpose of independent testing. Due to the stochastic nature of the splitting process, slight differences may occur between the derived CTs which affect their performance.

Therefore, in this work the process of knowledge base splitting, tree training and testing has been replicated at least 10 times until the mean value and standard deviation of independent case testing accuracy become stable.

The *Entropy* method is adopted to grow the CTs in CART. The performance of CTs in independent case testing is summarized in Table 2.2.

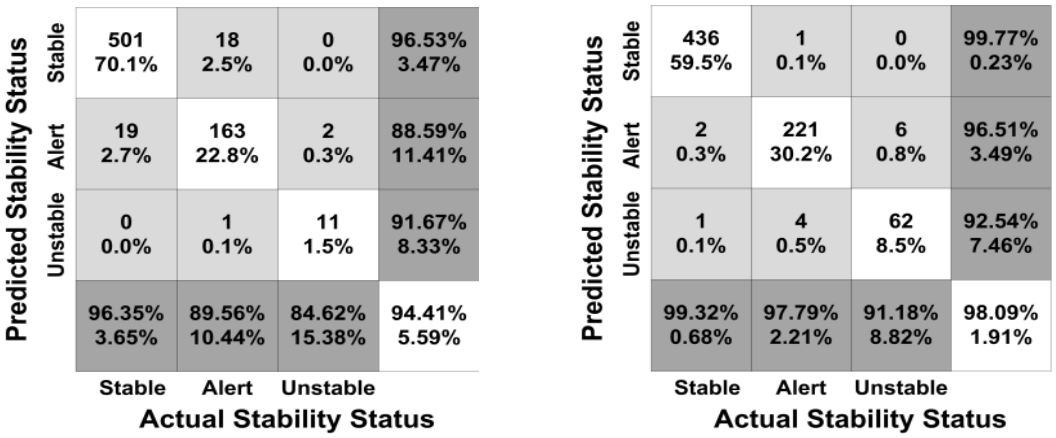
Table 2.2 Performance of the Classification Tree

System	Method	Accuracy of New Case Testing	
		OSA	VSA
9-Bus	*Entropy*	98.63%	99.56%
39-Bus	*Entropy*	94.38%	97.95%

The independent case testing results of CT for the IEEE 39-bus system are shown in Figure 2.8. An interesting observation from Table 2.2 and Figure 2.8 is that the CT performance for OSA is less than that of VSA. This is because the system oscillatory stability behavior is highly non-linear. In order to reach certain prediction accuracy, a larger training dataset is needed by OSA-CT compared with VSA-CT. In this work, more instances could be generated if we set the *Stopping Criterion 2)* in Section 4.2 with a higher accuracy requirement.

The classification tree can be developed using different methodologies, e.g. *Gini*, *Twoing*, and *Entropy*, [12]. Another important setting is the minimum cases a parent node should have, which may impact the size of resulted CT. In this work the tree settings are varied to explore their impact on assessment accuracy.

The results are shown in Figure 2.9.



a) Testing results of 39-bus OSA b) Testing results for 39-bus VSA

Figure 2.8 CT stability assessment for the 39-bus system in one replication

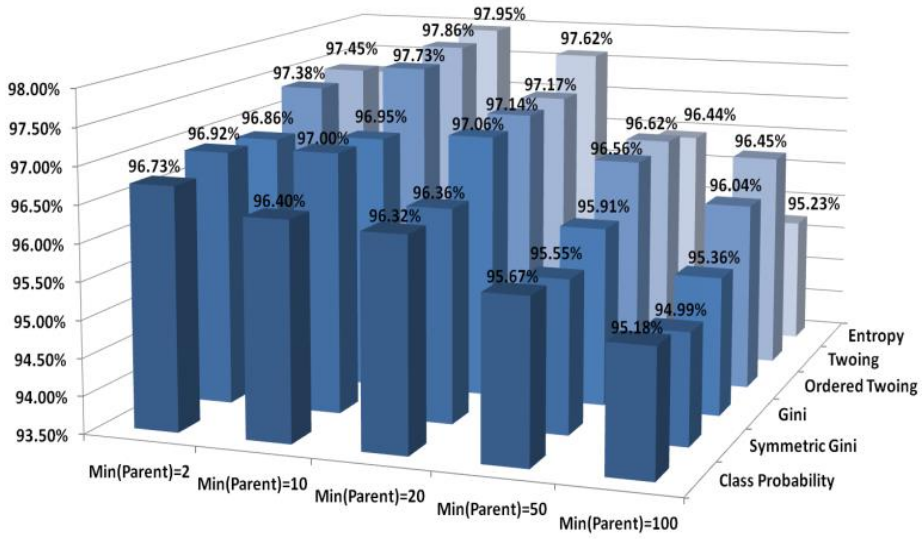


Figure 2.9 Classification tree performance using different tree growing methods

Two conclusions could be made from Figure 2.9: 1) the CT performance for the stability assessment problem is related to how a tree is trained. In this case the *Entropy* method achieved the best classification accuracy; 2) the setting for minimum parent node cases can alter the shape of the resulted tree as well as its performance. In general, the more cases a parent node is required to have, the fewer terminal nodes the derived CT

may possess. This experiment demonstrated that there is a trade-off between tree complexity and accuracy. A large-sized tree may encounter the over-fitting problem, whereas a small-sized tree that is not adequately developed may produce less accurate classification results. A trial and comparison process is needed to find the best CT size, and this can typically be accomplished by nested cross-validation.

2.3.5 Summary

This section explores the use of classification trees for fast evaluation of oscillatory stability and voltage stability. The following is a summary of the research:

- The two previously proposed stability metrics have been deployed to define corresponding stability states. The classification trees are trained to estimate system operating stability status in real time;

- A systematic methodology for knowledge base generation has been proposed. Stopping criteria were elaborated to assure a sufficient dataset for CT training. Encouraging results were obtained through performance examination using the generated knowledge base;

- The CT classification accuracy is related to how the tree is developed, and the setting for minimum parent node cases can alter the shape of the resulting tree impacting its accuracy.

PSEUDO-CODE FOR KNOWLEDGE BASE GENERATION

1. Initialize PSS/E in Python. Import system model parameters:
Number of Generation Buses = i, Number of Load Buses = j
Number of buses with shunt capacitor = k

2. Let u ($u \in N$) be the iteration index with a step change of $C_{G/L/S}\%$
Suppose G_1 is slack bus. Repeat:
for $A_2=0 \rightarrow u_2$ do
Scale the output of G_2 to: $P_{G2} = P_{G2}^0 (1 + A_2 \times C_{G2}\%)$
…
for $A_i=0 \rightarrow u_i$ do
Scale the output of G_i to: $P_{Gi} = P_{Gi}^0 (1 + A_i \times C_{Gi}\%)$
for $A_{i+1}=0 \rightarrow u_{i+1}$ do
Scale load 1 to: $P_{L1} = P_{L1}^0 (1 + A_{(i+1)} \times C_{L1}\%)$
…
for $A_{i+j}=0 \rightarrow u_{i+j}$ do
Scale load j to: $P_{Lj} = P_{Lj}^0 (1 + A_{(i+j)} \times C_{Lj}\%)$
for $A_{i+j+1}=0 \rightarrow u_{i+j+1}$ do
Scale shunt 1 to: $Q_{S1} = Q_{S1}^0 (1 + A_{(i+j+1)} \times C_{S1}\%)$
…
for $A_{i+j+k}=0 \rightarrow u_{i+j+k}$ do
Scale shunt k to: $Q_{Sk} = Q_{Sk}^0 (1 + A_{(i+j+k)} \times C_{Sk}\%)$
Solve the load flow at: $\{P_{G2},...,P_{Gi},P_{L1},...,P_{Lj},Q_{S1},...,Q_{Sk}\}$
If this OP is unsolvable: eliminate

Oscillatory Stability Analysis:
Import system model dynamic data. Derive the *A* matrix.

Voltage Stability Analysis:
Derive the voltage collapse point via continuation-based method
Export computed features of current OP
End Loops

3. Repeat: for i=0→number of OPs do
Modal analysis of *A* matrix using (3)-(5): *DR* (ζ_i)
Compute voltage stability index using (6)-(7): VS^i_{margin}
Export computed stability margins
End Loop

2.4 Model-based Approach for Real Time Stability Margin Prediction Using Regression Tools

2.4.1 Proposed Research

2.4.1.1 Regression Tree Method

Compared with the traditional time domain simulation approach that requires full model computation each time a new OP has emerged, the advantage of RT method lies in its simplified model structure and fast OP analysis facilitated by fewer required inputs. Figure 2.10 provides a simple example of RT structure. The unfolding OP is related to its stability margin through a unique top-down path. The splitting rule at each node that belongs to a given path represents an operational threshold. Based on the combination of splitting rules along the path, preventive and corrective control strategies could be formulated and initiated.

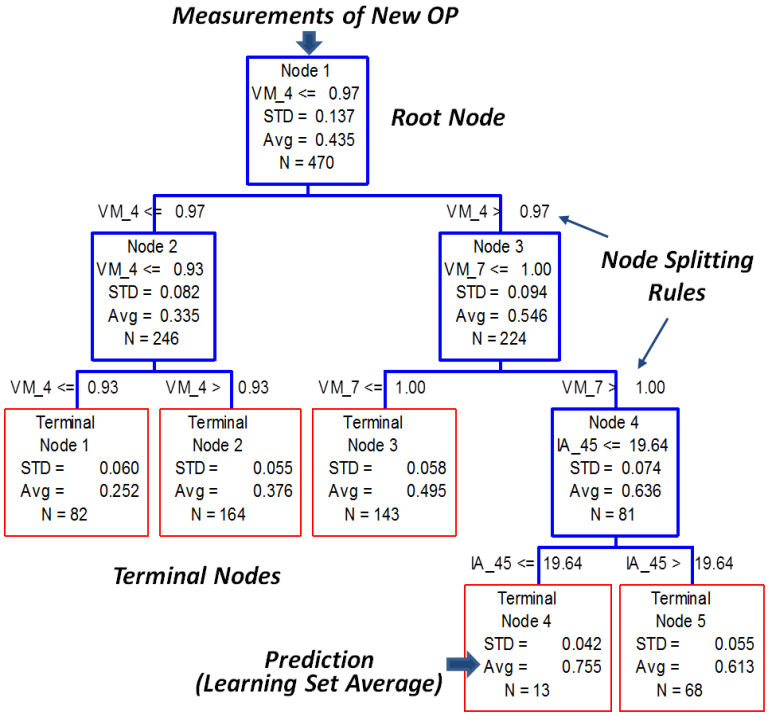


Figure 2.10 An example of the RT model structure

In regression analysis, a case refers to an instance (x, y) where x is the vector of attributes and y is the target value, to be predicted. The relationship between x and y is usually described by a regression function, through which it is possible to estimate how the target y changes when x is varied. In our proposed approach, the regression function is replaced by a binary tree structure, where x are the synchrophasor measurements and y is the system stability margin, i.e. the damping ratio or MW-distance. CART is used to develop OSM-RT and VSM-RT used for evaluating oscillatory and voltage stability margins.

The approach to build a RT entails three steps: 1) tree growing using learning dataset; 2) tree pruning using a test dataset or cross-validation; 3) selection of the best pruned tree. Experimental tests show that there is a trade-off between the tree complexity and its accuracy: a small-sized tree cannot capture enough system behavior, and a large-sized tree usually leads to imprecise prediction due to over-fitting on training data. In this work the rule of minimum cost regardless of size to search for the best pruned RT commensurate with accuracy is adopted. The complexity cost parameter in CART has been set to equal to zero. The RT growing, node splitting, tree pruning and optimal tree selection algorithms are detailed in Appendix.

2.4.1.2 Proposed Approach

The proposed framework for RT-based stability margin prediction and event detection is shown in Figure 2.11. PMU measurements from different substations are collected and time-aligned by the Phasor Data Concentrator (PDC). The synchrophasor measurements are then delivered to the Wide Area Measurement System (WAMS) server located at the central control facility. At the control center operator room, the RTs for

monitoring OSM (OSM-RT) and VSM (VSM-RT) are trained and updated periodically. The PMU data of an upcoming OP is dropped down the respective tree until it reaches a terminal node. Then the predicted stability margin is the average value of the learning set samples falling into that terminal node. Any OP with insufficient stability margin will be detected immediately by checking corresponding thresholds. Operators are alerted with the possible event and preventive control strategies can be initiated in a timely manner.

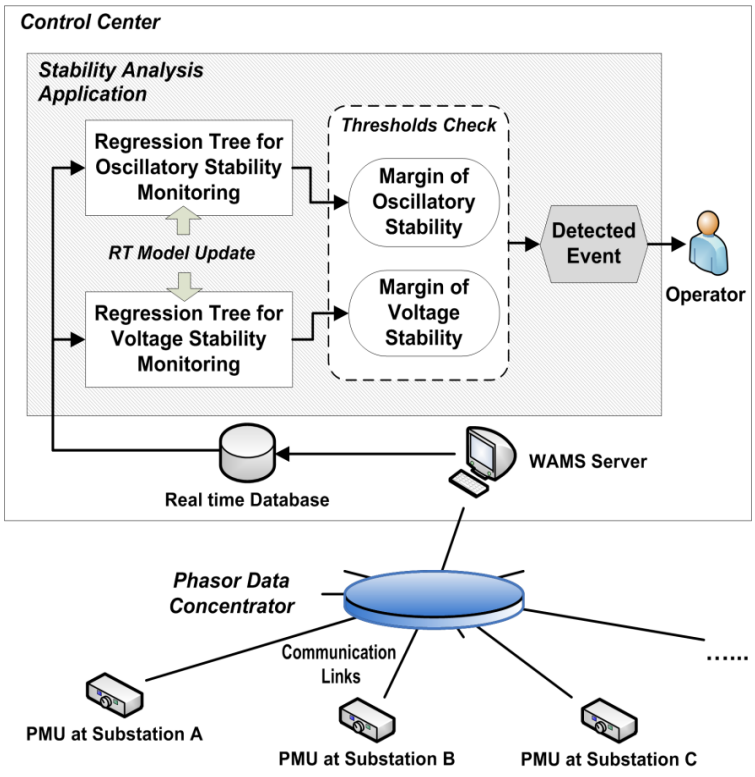


Figure 2.11 Proposed framework of the RT-based stability margin prediction and event detection

2.4.2 Knowledge Base Generation

Using the approach illustrated in previous section, the power supply at generation buses, demand at load buses, and the output of shunt capacitors were systematically

varied. A total of 1071 OPs with corresponding OSMs, and 1153 OPs with corresponding VSMs have been produced for the 9-bus system. The number of records generated for the 39-bus system knowledge base is 4276 and 3664 for the VSM and OSM tasks, respectively.

In addition, in this work the generator active/reactive power limits have been taken into account to reflect the practical stability margin. This has significant impact on the computation of VSM: when the load demand increases, a feasible load flow solution may not exist due to the limited generation capacity, even before the maximum loadability of the transmission system is reached. Therefore the derived P_{max} may be somewhere on the top half of the PV curve before the "Knee" point shown in Figure 2.6.

In order to build a sufficiently large knowledge base, in this work two stopping criteria are followed:

1) Each generator/load/shunt should be varied at least 4 times ($u \geq 4$) and the total variation should be at least 30% of the base value ($u \times C_{G/L/S} \geq 30$). The goal is to capture the most system behavior from the problem space;

2) The RT training and testing accuracy converges. The R^2, residuals squared, metric is used to measure the prediction accuracy and will be detailed in next section.

The trajectory of the 39-bus system stability margin is shown in Figure 2.12. Corresponding stability thresholds are shown as the flat planes dividing each margin space into two halves: an instability event will be immediately identified in the top half. For this power system the voltage stability threshold is put at VS_{margin}=30%. This value

can be further adjusted according to the real-time operational needs.

As it can be observed from Figure 2.12, a large imbalance in size between the stable and unstable cases exists. This is a very practical issue in power system operation since most of the time the system is in its stable state. From the classification point of view, compared with some other data mining tools that do not perform well when dealing with unbalanced data, the decision tree implemented in the CART software has the property of assuring that every class is treated equally regardless of its size. This is achieved by specifying the *Prior* of each class. From the regression point of view, there is no need to set *Priors* because each case will be treated as an equal point on the continuous stability margin space. Because of the least squares loss function for regression, as implemented in CART, large mistakes are penalized more than smaller ones, thus large errors at any OP are emphasized, be they on the stable or unstable part of the stability margin space. Once the relationship between input and output is identified, the regression model defines a mapping of an OP to its stability margins regardless of the state/class the OP belongs to.

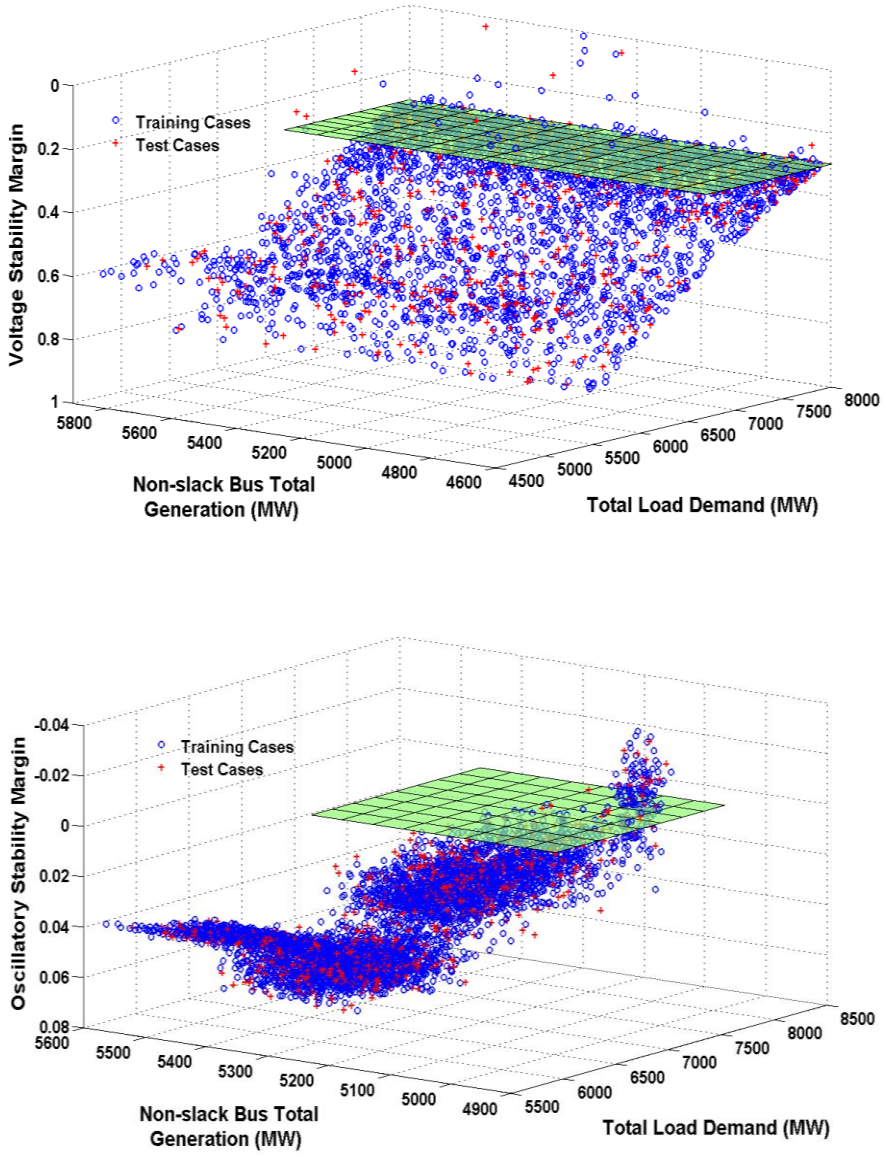


Figure 2.12 Trajectory of voltage and oscillatory stability margins of the IEEE 39-bus (New England) test system

2.4.3 Off-line Training and New Case Testing

Each knowledge base is split into two independent data sets: 80% of the records are randomly selected for training of OSM-RT and VSM-RT; the remaining 20% of the records will serve the purpose of RT testing. The 10-fold *cross validation* method is adopted to grow the RT in CART. In experiments, because of the random nature of the

splitting process, slight differences may occur between the performances of each derived RT. Therefore in this work, the process of knowledge base splitting, tree training and testing has been replicated 10 times, until the mean and standard deviation of RT accuracy become stable.

In contrast with a classification tree for which the accuracy could be directly derived from the misclassification rate, the performance of a regression tree is measured through a statistical index, termed *Residuals Squared Error* (R^2) [36]. We report the accuracy of a RT model as follows:

$$\mathrm{R}^2 = 1 - \frac{\sum_{TS} [y_i - d(x_i)]^2}{\sum_{TS} (y_i - \overline{y}_{root})^2} \qquad 2.10$$

where *TS* is the set of training samples, x_i is input, y_i is the actual stability margin, $d(x_i)$ is the RT predicted value, and y_{root} is the mean of y_i in the tree root node.

In general the closer the value of R^2 is to 1, the better the prediction is. However in practice, how good an R^2 is depends on the particular application and the way it is measured [37]. Experimental results from this work show that a quite acceptable value of R^2>0.90 can be achieved.

Sometimes the R^2 alone may not be sufficient, especially in the case when the typical difference between values predicted by RT and the actual stability margins is desired. Therefore another measure, the *Root-Mean-Square* (RMS), is utilized:

$$RMS = \sqrt{\frac{\sum_{i=1}^{n} [y_i - d(x_i)]^2}{n}} \qquad 2.11$$

where *n* is the number of test cases. The numerator stands for the sum of squared deviations of the actual stability margins around the RT predictions. The value of RMS error depends on the base magnitude of the target stability margin to be predicted. In the proposed scheme, a typical value of OSM is in the range of -0.01 to 0.1, and the VSM is usually ranging from 0.05 to 1.0. Hence the RMS errors of VSM-RT are usually several times larger than that of the OSM-RTs.

Once the training is complete, the derived RTs are evaluated using the unseen test cases. Much more emphasis must be put on the accuracy of unseen case testing because, for real-time applications, a predictive model which cannot predict the unseen system behavior well is unacceptable, even if high accuracy is obtained during the off-line training, as it lacks generalization power. The corresponding training and new case testing accuracy is summarized in Table 2.3. In addition, the results of new case testing were reported separately in terms of *Security Test* and *Reliability Test*. While the security test examines how well the stable OPs are predicted, the reliability test checks if all unstable OPs are correctly identified.

The prediction for 300 new OPs of the 39-bus system is shown in Figure 2.14. The RT-based approach has exhibited encouraging capability for system stability margin prediction.

The performances of differently sized OSM-RTs are summarized in the relative error curve shown in Figure 2.15. Among these trees, a 13-node subtree pruned from the 45-node “optimal” tree is shown in Figure 2.15 (a), and the “Largest” tree with 465 nodes is shown in Figure 2.15(b).

Table 2.3 Performance of the Regression Trees

System	Oscillatory Stability Margin (OSM-RT)				
	Train R^2	Unseen OPs Overall Accuracy		Reliability and Security Test (RMS)	
		R^2	RMS	Reliability	Security
9-bus	0.9984	0.9858	0.0023	0.00083	0.00235
39-bus	0.9617	0.9519	0.0034	0.00386	0.00328
System	Voltage Stability Margin (VSM-RT)				
	Train R^2	Unseen OPs Overall Accuracy		Reliability and Security Test (RMS)	
		R^2	RMS	Reliability	Security
9-bus	0.9928	0.9791	0.0184	0.03357	0.01480
39-bus	0.9941	0.9694	0.0211	0.02736	0.01965

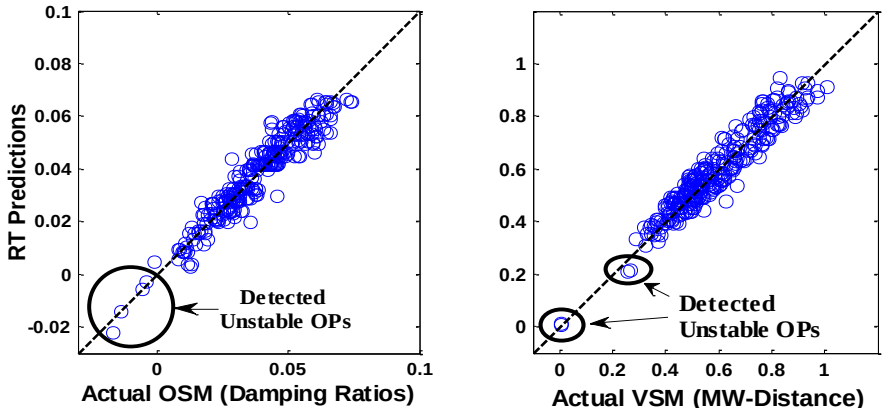


Figure 2.13 RT predicted margins versus the actual stability margins of the IEEE 39-bus system. Left: OSM-RT performance; Right: VSM-RT performance

Compared with the optimal tree, numerical results show that although the 465-node tree has boosted the training accuracy from 0.9617 to 0.9872 R^2, its accuracy in unseen case testing actually dropped from 0.9520 to 0.9407 R^2. This is because while an over-developed tree may perform well in training, but it will lose the generalization power in predicting unseen instances. The optimal tree with the lowest relative cost has the best

generalization power and should be selected.

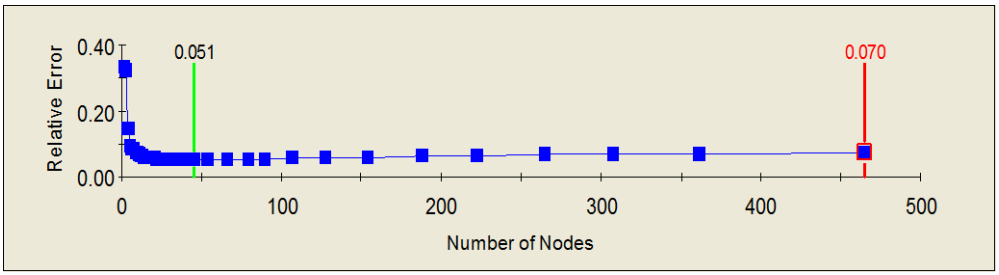
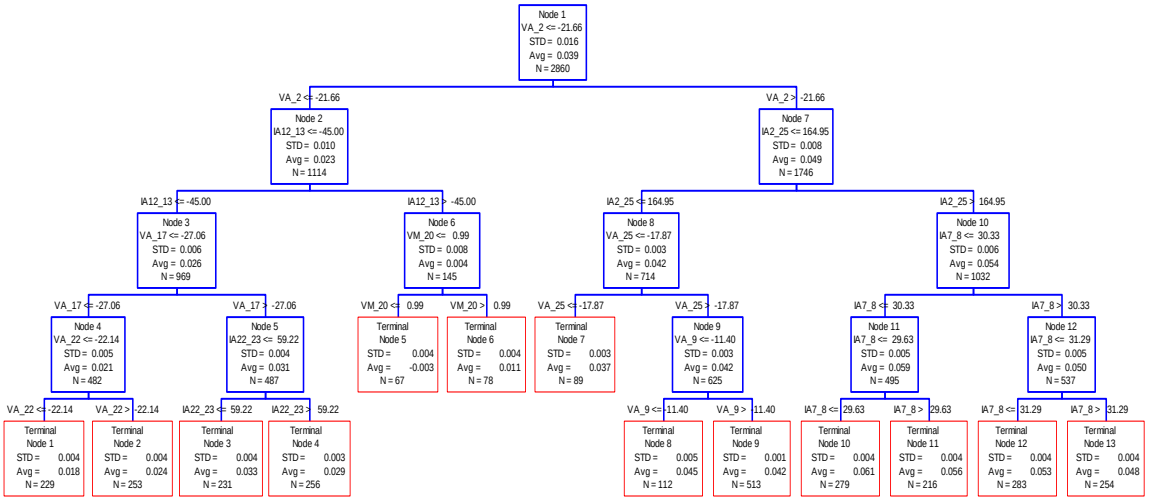
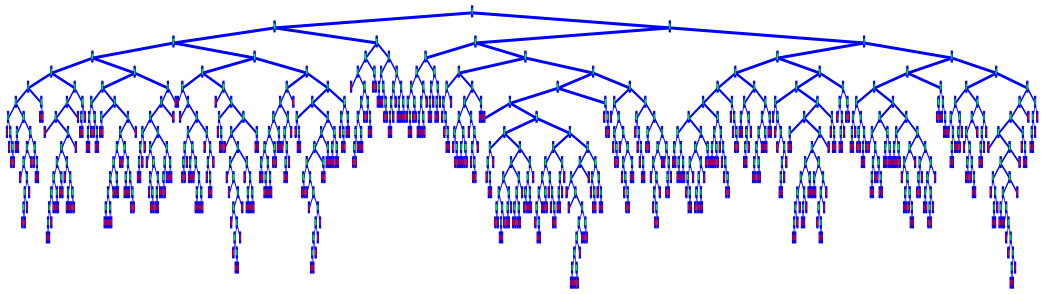


Figure 2.14 Relative cost of a series of differently sized RTs



(a) 13-node tree pruned from the optimal OSM-RT



(b) Largest RT with 465 terminal nodes

Figure 2.15 Regression trees for oscillatory stability margin prediction

2.4.4 Comparison with Other Data Mining Tools

In this work the performance of RT has been compared with two widely used data mining tools: Support Vector Machine (SVM) and Neural Network (NN). The R^2 accuracy of different data mining tools for the 39-bus system is summarized in Table 2.4.

Table 2.4 New Case Testing Accuracy using Different Data Mining Tools for the 39-bus System

Tools	Testing R^2 of OSM	Testing R^2 of VSM
RT	0.9519	0.9694
SVM	0.9591	0.9811
NN	0.9579	0.9572

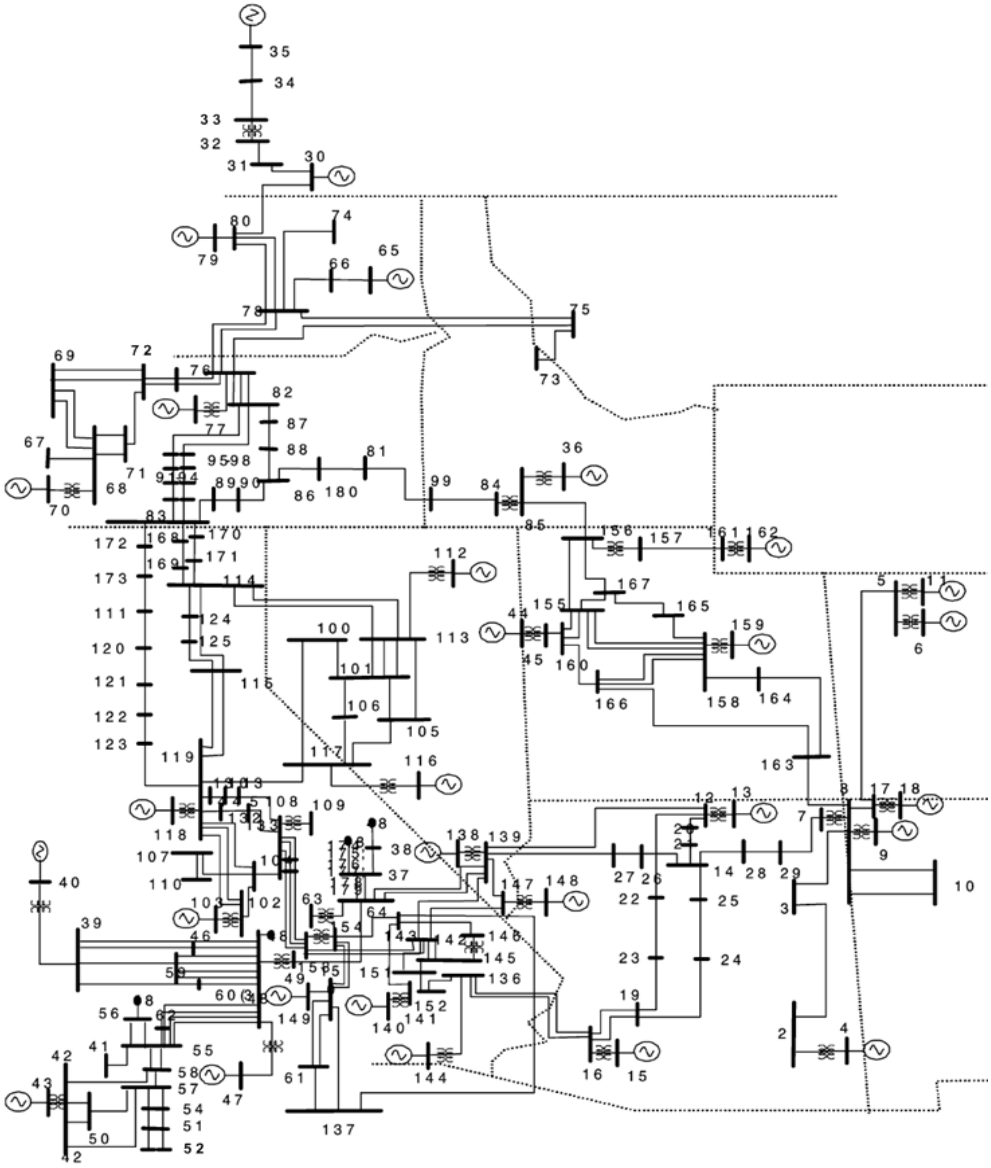


Figure 2.16 One-line diagram of the WECC 179-bus equivalent system

According to the results, the RT-based model achieved almost identical prediction accuracy as other data mining tools. Compared with some "black-box" tools, the DT piece-wise structure as shown in Figure 2.15 (a) provides system operators with a clearer cause-effect relationship of how the system variables lead to the onset of an instability event. It is possible to identify the critical variables and thresholds that need to be analyzed to gain insight into the stability margin of a system.

2.4.5 Application to a Larger System

2.4.5.1 Description of the WECC Equivalent System

The RT-based predictive model has been applied to the Western Electric Coordinating Council (WECC) equivalent system shown in Figure 2.16 [38]. This network consists of 179 buses, 29 generators, 42 shunts, and 104 loads.

2.4.5.2 Knowledge Base Generation and RT Performance

The same methodology of creating the knowledge base for the 9-bus and 39-bus systems is adopted. In addition, two practical issues have been considered: 1) the real/reactive power output limit of each generator is more stringent in this larger system and should be complied with strictly; 2) it is computationally too expensive to generate the database by varying only one component each time. For instance, if the iteration index u is set to be 4, a total of 4^{175} OPs will need to be analyzed. It may be more practical to group the loads and generators according to their geographical locations. Seven areas are formed and it is assumed that the loads/generators within each area will increase/decrease at the same rate.

A total of 12572 records have been generated for the OSM-RT and 15303 records for the VSM-RT. The impact of the size of training set on the performance of resulted RT is examined: 100%, 50%, 20%, 10%, 5%, and 2% of the training cases are used to derive RT for each task. All experiments have been replicated 10 times and the mean of unseen case prediction accuracy is summarized in Figure 2.17. It clearly shows that the prediction accuracy increased when more cases were used to train the RTs.

In order to embed the RT model into an actual online application, three aspects

need to be examined and corresponding requirements must be satisfied: 1) Eligibility for high speed analysis; 2) Robustness to measurement error; 3) Capability to accommodate topology change.

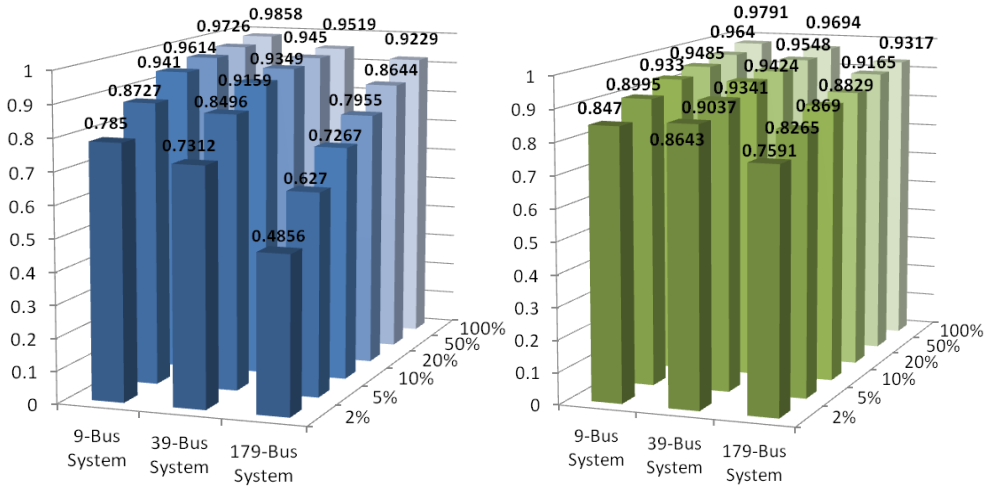


Figure 2.17 New case prediction accuracy of RTs trained with differently sized data sets. Left: OSM-RT; Right: VSM-RT

2.4.5.3 Data Processing Speed

Traditionally the data used for the stability analysis in electrical utilities are obtained from the SCADA system or state estimation functions, which are refreshed on a time scale from several seconds to several minutes. These slowly updated data can only provide limited decision making support for quickly developing situations where fast variations are present at both demand and supply side. The capability to take advantage of the quickly updated PMU data is critical in real-time applications.

In practice, the PMU measurements are updated very quickly, most likely at least 30 times per second. In order to evaluate the system stability status at each snapshot, the processing of PMU data must be less than 1/30=0.033 second.

Table 2.5 Computational Speed of Regression Trees

Type of Regression Models	IEEE 39-bus System		WECC 179-bus System	
	Off-line Training	New Case Prediction	Off-line Training	New Case Prediction
OSM-RT	36.01 s (3421 cases)	about 3 s (855 cases)	164.97 s (10058 cases)	about 5 s (2514 cases)
VSM-RT	31.38 s (2931 cases)	about 2 s (733 cases)	195.45 s (12242 cases)	about 7 s (3061 cases)

The data processing speed of RTs is summarized in Table 2.5. The computational time is estimated using the built-in clock of CART executed on an Intel Pentium IV 3.00-GHz CPU with 2 GB of RAM. It can be seen that the derived OSM-RT or VSM-RT can assess 1000 new OPs in less than 4 s for the 39-bus system, and 3000 new OPs in less than 8 s for the WECC 179-bus system. According to the results, the RTs satisfy the speed requirement of real-time applications.

2.4.5.4 Impact of Measurement Errors

The phasor estimation process may introduce errors. PMUs manufactured by multiple vendors can also yield inaccurate readings. In real-time application, the PMU measurement errors of the i^{th} OP can be expressed as:

$$VM_i^{meas} = VM_i^{real} + \Delta\varepsilon_{VMi} \qquad VA_i^{meas} = VA_i^{real} + \Delta\varepsilon_{VAi}$$

$$IM_i^{meas} = IM_i^{real} + \Delta\varepsilon_{IMi} \qquad IA_i^{meas} = IA_i^{real} + \Delta\varepsilon_{IAi} \tag{2.12}$$

where the superscript *real* means actual values of the phasor, and *meas* stands for measured values.

According to the IEEE C37.118 "*Standard for Synchrophasors for Power Systems*"

[39], PMUs that are Level 1 compliant with the standard should provide a Total Vector Error (TVE) less than 1%. This implies that the following constraints must be satisfied:

$$1\% > \left| \frac{VM_i^{meas} \angle VA_i^{meas} - VM_i^{real} \angle VA_i^{real}}{VM_i^{real} \angle VA_i^{real}} \right|$$

$$1\% > \left| \frac{IM_i^{meas} \angle IA_i^{meas} - IM_i^{real} \angle IA_i^{real}}{IM_i^{real} \angle IA_i^{real}} \right| \quad 2.13$$

Considering Eq. 2.12 and Eq. 2.13, random noise $\Delta\varepsilon$ has been added to the original phasor magnitudes and angles of the WECC 179-bus system knowledge base. In Table 2.6 two scenarios were tested. While in both scenarios errors were added to the test cases, it is shown that the RTs trained with measurement error had much better performance than the ones without the error taken into account in the training data set.

Table 2.6 Performance of the 179-Bus Regression Trees Considering PMU Measurement Error

Type of Regression Models	Add Noise Only to the Test Cases			
	Security Test		Reliability Test	
	R^2	RMS	R^2	RMS
OSM-RT	0.7906	0.00106	0.7403	0.00121
VSM-RT	0.8091	0.02785	0.7629	0.03010
Type of Regression Models	Add Noise to Both Training and Test Cases			
	Security Test		Reliability Test	
	R^2	RMS	R^2	RMS
OSM-RT	0.9170	0.00068	0.8994	0.00071
VSM-RT	0.9266	0.01789	0.9045	0.01940

2.4.5.5 Impact of Topology Variation

In this work the robustness of RT to certain system topology changes was examined. The scenarios that were evaluated and RT performances are summarized in Table 2.7.

Table 2.7 Regression Tree Performance under System Topological Variations

Scenarios of Topology Change	Type	RMS Error of OSM-RT	RMS Error of VSM-RT
Line 8-9 taken out	9 BUS N-1	0.00880	0.15481
G10 out of service	39 BUS N-1	0.00417	0.04089
G10 and Line 26-28 taken out	39 BUS N-2	0.00726	0.20702
Line 1 of 76-82 out of service	179 BUS N-1	0.00337	0.03046
Line 1 of 90-156 out of service	179 BUS N-1	0.00421	0.02654
Line 1 of 95-98 out of service	179 BUS N-1	0.00385	0.03198
Line 81-180 out of service	179 BUS N-1	0.00552	0.08325
Line 1 of 90-156 and Line 1 of 76-82 out	179 BUS N-2	0.00473	0.04830
G63-1 and Line 1 of 95-98 out of service	179 BUS N-2	0.00574	0.03792
G63-1 and Line 81-180 out of service	179 BUS N-2	0.00588	0.10736

It can be observed that OSM-RTs were able to provide somewhat acceptable predictions with low RMS errors, even under situations the network topology had changed.

On the other hand, VSM-RTs appear to be less robust and the performance varied case by case: the *N-1* test in the 9-bus system had a significant impact on the VSM

prediction due to the small size of the system; acceptable predictions were achieved for the case of generator outage in the 39-bus system; the *N-2* scenario in the 39-bus system was too severe for the VSM-RT to handle.

More case studies were conducted on the 179-bus system VSM-RT: low RMS errors were observed in experiments where slight topology changes are made, such as one of the double-circuit transmission lines out of service.

2.4.6 Discussion

2.4.6.1 Ability of RTs to Handle Evolving System Conditions

The problem of how to sustain the prediction accuracy of RT under the evolving system operating conditions is critical for its online implementation. In general, the change of system operating conditions can be categorized into two types:

- The variation of system load/generation patterns;
- The variation of system topology due to contingencies, scheduled maintenance, and system dispatch.

The work reported in previous sections tackles the first type of variation. As illustrated in the knowledge base creation process, the generator/load/shunt has been widely varied in a systematical way to capture the most system behavior from the problem space.

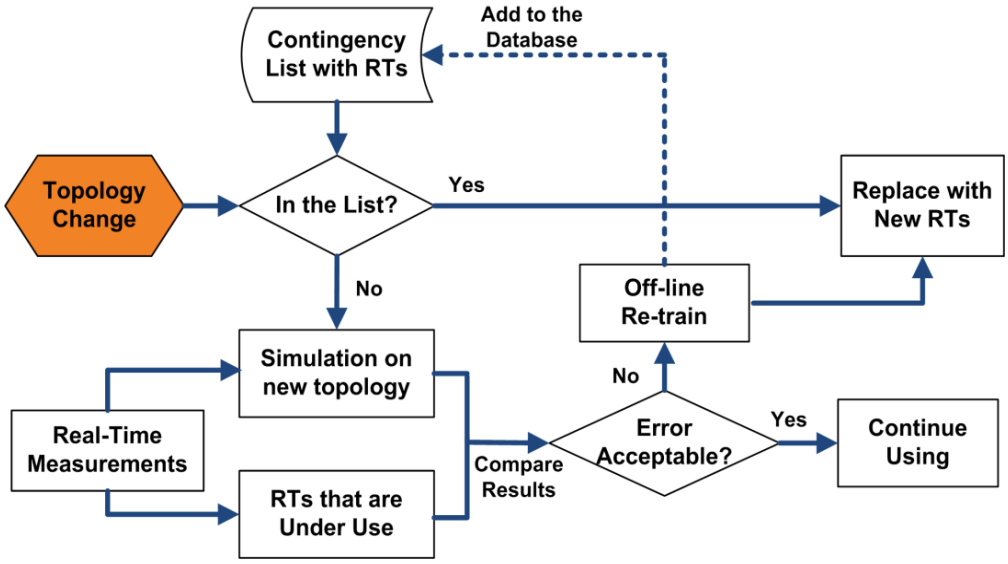


Figure 2.18 Scheme for RTs to handle system topology change

In our preliminary research we found that changes in system topology are a major reason causing a data mining tool to fail in real-time applications. The results shown in Table 2.7 indicate that the RT sensitivity to topology changes becomes less distinct in large sized networks and under milder topology changes. It is also observed that RTs are not able to accommodate certain severe contingencies, e.g. the line 81-180 out of service. In the field of data mining and machine learning, the so-called 'concept change' describes methodology for dealing with such type of topology variation. A literature search reveals that there is not a generally effective way for the data mining tool to cope with concept change incrementally, although some work has shown results, [40]. Most of the time a re-train using an updated knowledge base is necessary to reflect new topology conditions.

2.4.6.2 When and How to Update the RTs

To re-train an RT model whenever it is obsolete is time-consuming and may not satisfy the requirement of seamless on-line monitoring. An effective solution may be to prepare a knowledge base for each of the credible contingencies beforehand, and train a

series of candidate RTs accordingly. Figure 2.18 shows the proposed scheme. The list of credible contingencies is usually readily available at utility companies. If in online application an unseen contingency occurs and RT fails to provide accurate predictions, a new RT will be trained and deployed. The new contingency scenario and RTs will be added to the historical database. With the increase of contingency scenarios accumulated in database, fewer unseen topology conditions will be encountered. The obsolete models can be quickly replaced by the candidate RTs corresponding to the post-contingency condition.

2.4.7 Summary

In this work the approach of using regression trees to predict power system stability margins is explored and the following conclusions have been reached:

- Synchronized voltage and current phasors have been used as RT input feature. With a sufficiently large knowledge base, the RT model can predict the system oscillatory and voltage stability behavior with high accuracy;
- According to the test results, the RT model is fast enough to process PMU measurements, and it is robust to handle measurement errors that are within 1% TVE;
- The RT sensitivity to system topology variation becomes less distinct in large sized networks and under mild changes in topology.

2.5 Active Learning for Optimal Data Set Selection

2.5.1 Introduction

Analysis of synchrophasor measurements using data mining tools, in pursuit of precise stability assessment, requires a sufficiently large training data set. Traditionally the process of learning the underlying power system behavioral patterns introduces a significant computational burden, such that exhaustive simulations of all possible system operating conditions are necessary. Advancements in machine learning make it possible, in some cases, to reduce the amount of operating conditions that need to be analyzed while learning, without impacting the accuracy of stability assessment. By using a probabilistic learning tool in the described active learning scheme to interactively query operating conditions based on their importance, we show that significantly less data needs to be processed for accurate voltage stability and oscillatory stability estimation. Results show that the advantage of active learning is greater on more complicated power networks, where larger training data sets are involved.

Traditional power system stability assessment relies on detailed system modeling and time domain simulations to estimate the stability condition of interest. While this approach is straightforward and accurate, as long as a precise system model and adequate measurements are used, it may introduce significant computational complexity, considering the large size of modern power systems.

The recently emphasized importance of real-time stability monitoring has led to applications based on data mining tools such as classification and regression trees [11], artificial neural networks [24], and support vector machines [25]. While such tools can be used to provide near real-time stability estimation of a power system, compared to time

domain simulations, the large amount of operating conditions required for the training process is still a major obstacle to their online implementation. The occurrence of a fault event or system topology change, common in real time system operations, usually requires the data mining tools to be updated in order to reflect the evolving system configurations. In such situations, the re-training process may be an obstacle to seamless online stability monitoring.

In this project we focus on reducing the computational burden of training data mining tools by applying a pool-based active learning methodology. This approach reduces the number of operating conditions that need to be generated via time domain simulations, and consequently considered during training, without impacting the stability assessment accuracy.

2.5.2 Background

In this work two types of power system operational performance have been examined. Power system voltage stability deals with how far the system load demand is from the combined transmission and generation capability [10], while oscillatory stability is related to whether the system damping torques are sufficient to bring the system back to a steady-state operating condition after a disturbance [9]. The data comes from PMUs.

Data mining tools have been previously applied in power systems to assess the transient stability [13], system operational security [17] and system post-disturbance stability [22]; often in cases where the computational complexity of detailed modeling may be alleviated by creating highly accurate but approximate predictors. In [29] and

[41] the authors have used data mining tools to efficiently estimate the system voltage and oscillatory stability margins from system measurement data. In this work we explore a meta-learning scheme [42] aimed at reducing the computational burden of training, easing the application of data mining-based stability assessment.

Active learning has often been applied in cases where labeled examples are time consuming to obtain [43]. Pool-based active learning has been explored in situations where it is necessary to have a human expert provide labels for data [44] and classification of large amounts of networked data [45]. This kind of active learning may be used to select the optimal subset from a pool of available PMU data for which to provide labels via time domain simulation, to be used for predictor training. A detailed and recent overview of the active learning literature is given in [43].

2.5.3 Methodology

The task of power system stability assessment may be cast as a data mining classification problem [13], [17], [29]. In this case a data mining tool is used to create a mapping from the synchrophasor measurements, in our case the positive sequence voltage magnitude and angle, and the positive sequence current magnitude and angle, into one of the pre-determined stability states, or labels. The data are collected from PMUs installed at system substations, and synchronized using a satellite-based global positioning system (GPS).

The stability states are determined according to the value of the corresponding stability margin indicator. In the case of oscillatory stability the damping ratio (*DR*) of the critical oscillation mode may be used as the stability margin indicator, and two basic

stability states can be defined as: "Stable" (with critical damping ratio, $DR_{crit} > 0$) and "Unstable" (with $DR_{crit} < 0$). Similarly, the voltage stability margin (VS_{margin}) may be defined using the continuation power flow (CPF) technique [20]. The MW-distance of the current system operating conditions (OC) from the critical voltage collapse point (usually the saddle-node bifurcation point) on the P-V curve is shown in Figure 2.19. Two voltage stability states have been defined as being "Stable" or "Alert" based on VS_{margin}. In this work the voltage stability threshold is set at $\boldsymbol{\theta_{STB}}$ =30%, however value can be further adjusted according to the real-time operational needs.

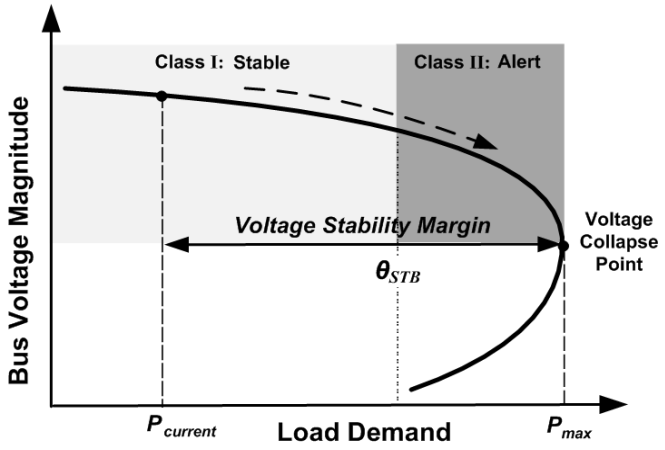


Figure 2.19 Methodology for voltage stability assessment

For simplicity of notation let us denote the synchrophasor measurements across a power system, including voltage magnitude and angle, and current magnitude and angle, characterizing the system in an OC i as $x_i = [x_{i1}, x_{i2} \ldots x_{i4P}]$, where P is the number of installed PMUs. In the case of voltage stability, for each system OC i let us label voltage stability $y_i = 1$ if $VS_{margin} > 30\%$ and -1 otherwise. Similarly, in the case of oscillatory stability let us label $y_i = 1$ if the oscillatory stability state is stable ($DR_{crit} > 0$) and -1 otherwise.

Gathering all measurements and their associated labels creates a labeled data set $\boldsymbol{D}_{\mathrm{L}} = \{(x_i, y_i,), i = 1 \ldots N\}$, where N is the total number of system operating conditions considered. A data set $\boldsymbol{D}_{\mathrm{L}}$ that may be used to train a data mining tool for either voltage or oscillatory stability margin predictions is therefore produced through extensive time-domain simulation. Let us also introduce the notation $\boldsymbol{D}_{\mathrm{U}}$ for a pool of unlabeled measurements, consisting of OCs without their associated stability margin labels.

In our previous work, [29], [41], we found that among the systematically generated OCs some are redundant and others are spurious. Spurious data can be considered outliers that should be removed from the training data set, for example by using techniques such as interquartile range measures [46].

The proposed approach is initialized by assuming all the measured data points are unlabeled, in $\boldsymbol{D}_{\mathrm{U}}$. We then apply the presented pool-based active learning methodology to incrementally select, label and include only points judged significant for learning into $\boldsymbol{D}_{\mathrm{L}}$. The procedure is iterated until a desired accuracy threshold is reached, or the budget of data points that may be included in $\boldsymbol{D}_{\mathrm{L}}$ is expended.

In the case when labels are computed beforehand for all examples the presented pool-based methodology reduces only the computational costs associated with learning. When labels for all OCs are not pre-computed a substantial reduction in both time domain simulation and learning may be possible, since not all labels may need to be computed.

Our approach uses the probabilistic and generalization properties of artificial neural networks and support vector machines to decide which system states should be labeled

and consulted during training, and which should not because they contain redundant information.

2.5.3.1 Artificial Neural Networks

Artificial neural networks (ANNs) are a biologically inspired mathematical model with significant applications in data mining. Feed-forward neural networks are composed of interconnected processing units, or neurons, each of which compute a simple transfer function, most commonly the logistic sigmoid, based on sum of their inputs, and produce an output, which may then be fed as the input into other neurons, until the output stage is reached. Therefore a neural network may be characterized by the number of neurons and connections between them.

In our case the network architecture is a directed acyclic graph having $4P$ input neurons and one output neuron, with a hidden layer of 10 neurons in between. Training is performed by adjusting the weight of connections between neurons until a close match between the inputs x_i and the desired output, y_i, is obtained through the network across all training examples i. When making a prediction the input is propagated through the network and a continuous output value is produced at the output neuron.

In traditional applications to classification tasks the output of ANNs is compared to a threshold in order to obtain a discrete prediction. For active learning, however, we will use the raw output as is typically seen in regression tasks because it can be used to provide a measure of uncertainty.

A specific property of feed-forward artificial neural networks using a logistic sigmoid transfer function is that this tool generalizes the entire possible input space even if only a few examples are available for training, and may falsely provide highly

confident predictions for unseen examples which are very dissimilar to any observed data points.

2.5.3.2 Support Vector Machines

Support vector machines (SVMs) are mathematical models which, in their simplest form, solve a linearly separable binary classification problem by finding the maximum-margin hyper-plane separating the two classes of points. More sophisticated SVM variants can make accurate predictions for non-linear problems, and are resilient to the presence of noise in data.

For the pool-based active learning methodology presented here the SVM is used in regression mode, as an implicit probabilistic classifier (see Active Learning Methodology), which may be assumed to provide the certainty of an example belonging to a class. There are several variants of SVMs distinguished by the kernel function that is employed to compute distance between data points. In our preliminary experiments we have obtained the most accurate results using the radial basis function (RBF) kernel. Unlike the logistic sigmoid based neural networks, a properly trained SVM using the RBF kernel does not provide confident predictions for points which are dissimilar to examples observed during training. For the following experiments the SVM is used as implemented in the LibSVM library [47].

2.5.3.3 Active Learning Methodology

In active learning a probabilistic data mining tool is used to interactively query a source of information (or oracle) that is assumed to always be correct, but is expensive to use. In our work the oracle is time domain simulation of a power system. With pool-

based active learning we assumed a large number of unlabeled measurements $x_i \in \boldsymbol{D}_U$ are available without their associated labels y_i. In this work we have explored an active learning methodology based on uncertainty sampling by choosing to label those examples whose class probability is closest to 0.5. Computing the uncertainty a predictor has about an unseen example, based on the output of a trained ANN or SVM requires the scalar continuous output $f(x_i)$ to be transformed into the probability of that example belonging to the positive class $p(y_i = 1|f(x_i))$. This can be accomplished by the transformation, [48],

$$p(y_i = 1 | f(x_i)) = \frac{1}{1 + \exp(Af(\mathbf{x}_i) + B)}. \quad 2.14$$

This function is monotonous and increasing for any value of B and of $A < 0$. Therefore we may conclude that the output of ANNs and SVMs can be implicitly interpreted as the class probability and used directly in active learning by considering predictions $f(\mathbf{x}_i)$ closer to 0 in absolute terms as more uncertain, or having $p(y_i = 1|f(\mathbf{x}_i))$ closer to 0.5, than those farther away from 0.

The proposed active learning procedure is initialized by asking the oracle to provide the labels for a small number of examples from $\boldsymbol{D}_U$, removing them from $\boldsymbol{D}_U$ and including them in $\boldsymbol{D}_L$. After learning on $\boldsymbol{D}_L$ the tool makes a prediction on all the examples for which labels have not yet been computed, $\boldsymbol{D}_U$, and finds those which have predictions closest to 0 in absolute terms. In other terms the unlabelled examples are sorted according to certainty the tool has about their label and those with highest uncertainty are used to query the oracle again.

PSEUDO-CODE FOR POOL-BASED ACTIVE LEARNING
1. Label and remove a small subset of examples from $\boldsymbol{D}_U$ and place into $\boldsymbol{D}_L$ 2. While stopping criteria is not met: a) Train classifier on $\boldsymbol{D}_L$ b) Make predictions on $\boldsymbol{D}_U$ c) Choose a small subset of $\boldsymbol{D}_U$ based on maximum uncertainty, remove them from $\boldsymbol{D}_U$, acquire labels for chosen examples from oracle and include them in $\boldsymbol{D}_L$

2.5.4 Experiments

Two IEEE test systems, namely the IEEE 3-machine 9-bus system and the IEEE 10-machine 39-bus (New England) system, are used to evaluate the proposed approach.

In order to create a sufficiently large training data set, different OCs have been generated by systematically varying the system generation/shunt outputs, as well as the load demands. PSS/E is used to perform load flow calculations, formulate linearized system models through numerical perturbation, and derive corresponding stability margins. MATLAB and Python add-on scripts are developed to automate this process. The procedures for creating the training data set are illustrated in Figure 2.20.

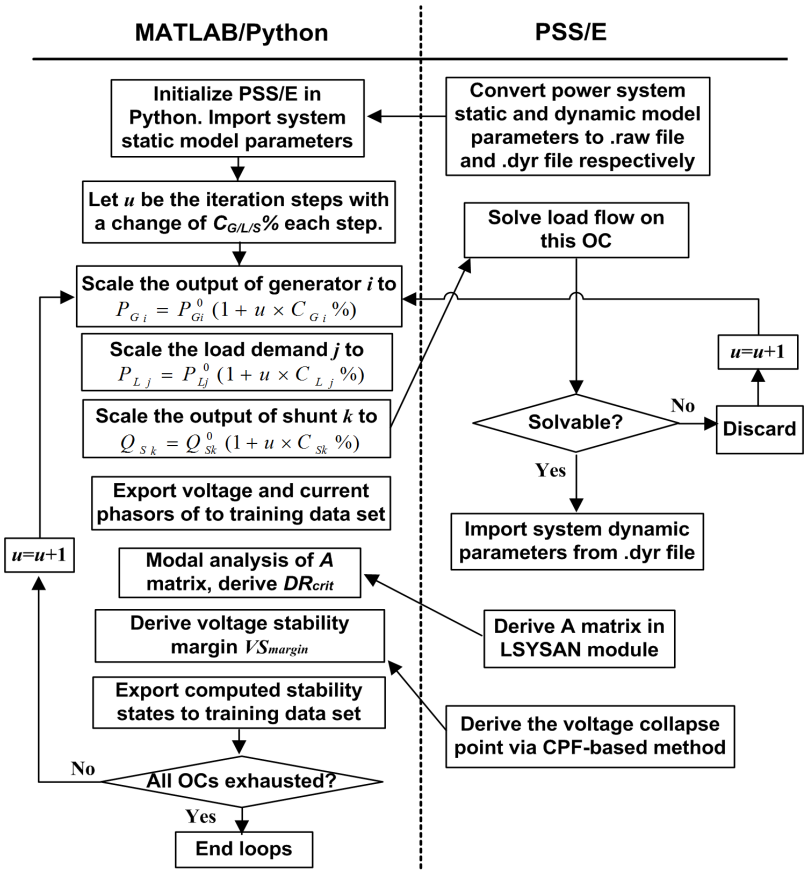


Figure 2.20 Procedures for creating the training data set

Additionally, to build a sufficiently large training data set, each generator, load, or shunt has been varied at least 6 times ($u \geq 6$) and the total variation is at least 40% of the base value ($u \times C_{G/L/S}\% \geq 40\%$). The goal is to capture the most system stability behavior from the problem space.

Using the procedures shown in Figure 2.20, and by labeling different OCs with corresponding stability states described in Section 5.3, the training data set generated for these two test systems is summarized in Table 2.8.

For the following experiments the pool-based active learning methodology was used to train SVMs and ANNs. We first performed experiments in batch-mode using 5-

fold cross-validation to obtain the optimal parameters for SVM and ANN training, and the used these parameters to test the active learning approach.

Table 2.8 Operating Points Generated for Training of Data Mining Tools

System	OPs Generated for Oscillatory Stability Estimation		OPs Generated for Voltage Stability Estimation	
	Stable OPs	Unstable OPs	Stable OPs	Unstable OPs
9-Bus	**1021**	**50**	**404**	**21**
39-Bus	**4950**	**126**	**1843**	**59**

We compared the performance of training on OPs chosen by active learning with training on random subsets of equal size. In the following figures each horizontal axis represents the number of OPs that were used for training, chosen either through active learning (full blue line) or random sampling (dashed red line), while the vertical axis represents the ratio of correctly classified examples to total examples. Because of the class imbalance we also present the results of the mean predictor (green dotted line) which always predicts the majority class, in our case the positive or stable class.

At each step of the proposed method we chose to label a single example from $\boldsymbol{D}_{\mathrm{U}}$ and include it in $\boldsymbol{D}_{\mathrm{L}}$. Testing is then performed across the entire set of generated OPs in order to illustrate the generalization power of the proposed approach; however this step is not necessary in real applications. In each experiment four initial OPs were labeled by the oracle in order to start the procedure.

2.5.4.1 Support Vector Machine Experiments

Let us first consider the 9-bus system and the problem of oscillatory stability classification. From Figure 2.21 we note that from the start of the procedure active

learning outperforms random sampling. Random sampling starts to outperform the mean predictor only after 50 examples have been labeled.

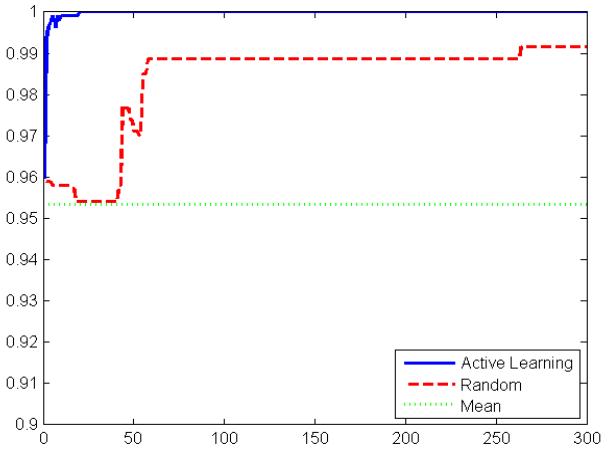


Figure 2.21 Comparison of active learning and random sampling on the 9-bus system for the oscillatory stability classification task using SVM

In Figure 2.22 we show the comparison results for the 9-bus voltage stability estimation performance comparison between active learning and random sampling. From Figure 2.22 it can be seen that active learning outperforms random sampling more than in the case of OSM prediction.

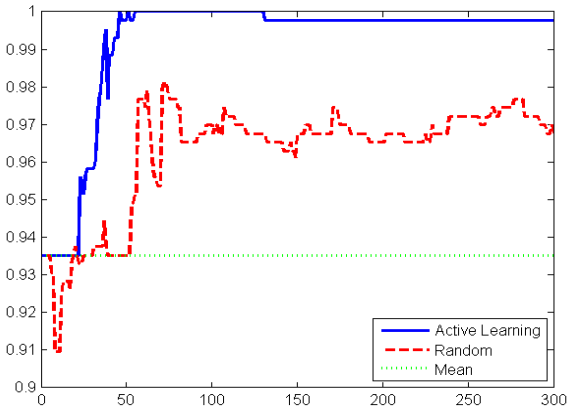


Figure 2.22 Comparison of active learning and random sampling on the 9-bus system for the voltage stability classification task using SVM

We hypothesize that this is due to the drastic difference between the sizes of the positive and negative classes. The difference in class sizes means that a greater variance may be expected when randomly sampling points because the addition of a few unstable OPs in $\boldsymbol{D}_{\mathrm{L}}$ may drastically change the decision boundary.

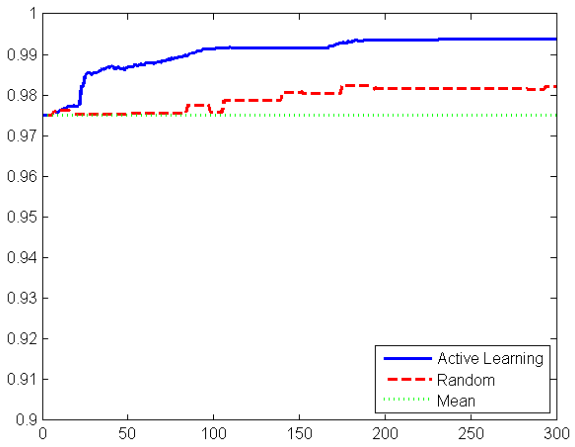


Figure 2.23 Comparison of active learning and random sampling on the 39-bus system for the oscillatory stability classification task using SVM

Next we will illustrate how the active learning approach performs on the 39-bus system oscillatory stability assessment using SVMs. From Figure 2.23 the active learning approach significantly starts to outperform random sampling after 100 examples are labeled.

In Figure 2.24 similarly to Figure 2.22 the simpler task of voltage stability margin estimation results in a smaller but still significant performance gain from using active learning.

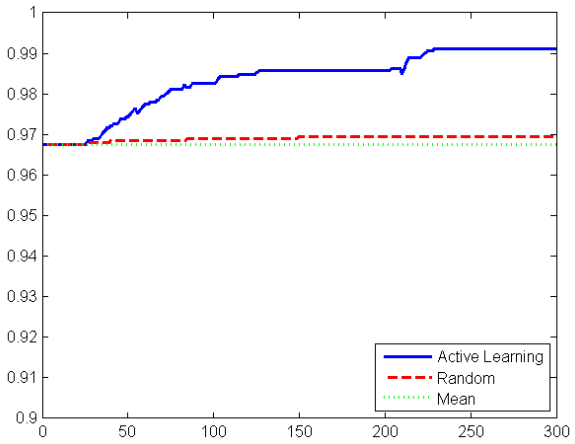


Figure 2.24 Comparison of active learning and random sampling on the 39-bus system for the voltage stability classification task using SVM

2.5.4.2 Artificial Neural Network Experiments

Unlike the SVM, in many cases the ANN using a logistic sigmoid transfer function may provide very confident predictions for data points dissimilar to those observed during training. Because of the imbalance of classes the four points used to initialize the active learning training will often of be in the positive, or stable, class. These two causes force the ANN to behave like a mean predictor, classifying the entire input space as the positive class with high confidence, until a negative example is included in $\boldsymbol{D}_{\mathrm{L}}$. To overcome this issue we included three positive and one negative point in the initialized $\boldsymbol{D}_{\mathrm{L}}$. In the resulting figures this is reflected as poor performance when very few examples are included in $\boldsymbol{D}_{\mathrm{L}}$. However, once enough points are included in DL the performance of ANN becomes closer to that of SVMs.

In Figure 2.25 we compare active learning to random sampling and the mean predictor when using ANNs on the oscillatory stability task using 9-bus system data. From Figure 2.25 the active learning provides significant improvement when few examples are observed. Interestingly, random sampling provides better results when using ANN than SVM on this task after 250 points are included in $\boldsymbol{D}_{\mathrm{L}}$.

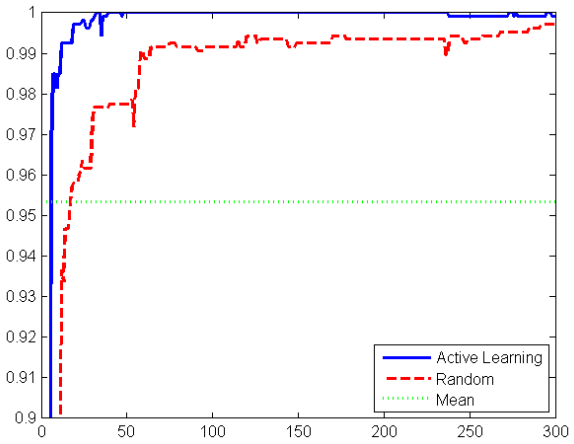


Figure 2.25 Comparison of active learning and random sampling on the 9-bus system for the oscillatory stability classification task using ANN

The next result, in Figure 2.26, shows the accuracy comparisons of using ANNs on the voltage stability task for the 9-bus system data set. Again after many labeled examples are included in DL the performance of random sampling becomes close to that of active learning.

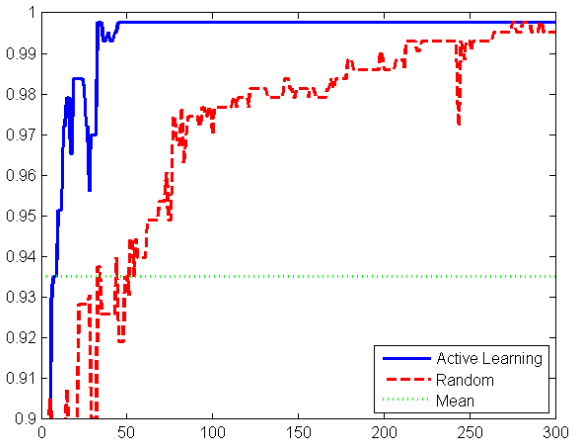


Figure 2.26 Comparison of active learning and random sampling on the 9-bus system for the voltage stability classification task using ANN

In Figure 2.27 we show the 39-bus system oscillatory stability experiment results. Here random sampling struggles to become more accurate than the mean classifier even

when 300 points are included in $\boldsymbol{D}_L$. The ANN trained using active learning provides higher accuracy than random sampling in this case as well.

Finally, Figure 2.28 we show the results of ANN using active learning and random sampling on the 39-bus system voltage stability classification task. Although initially in this case random sampling outperforms active learning, after 20 examples are included in DL the active learning trained ANN starts to outperform random sampling. Again, random sampling struggles to outperform the mean predictor.

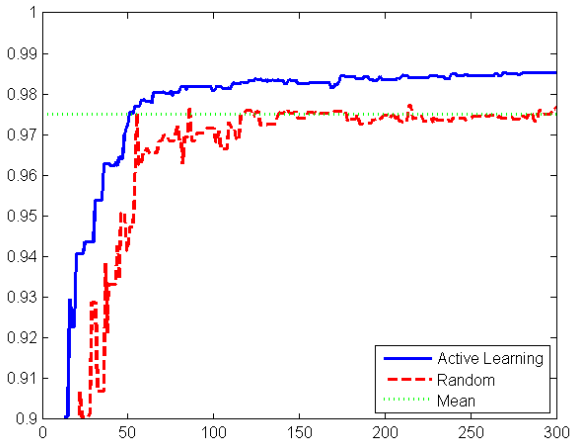


Figure 2.27 Comparison of active learning and random sampling on the 39-bus system for the voltage stability classification task using ANN

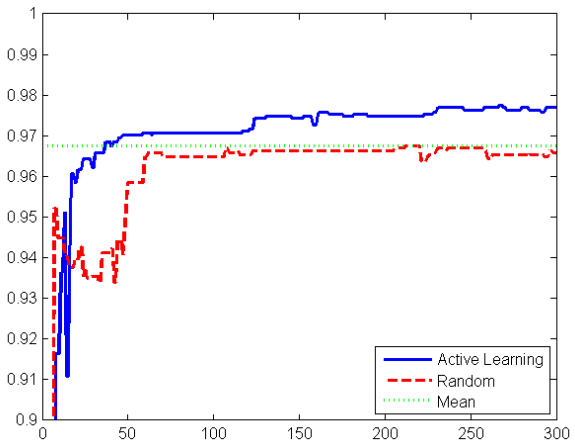


Figure 2.28 Comparison of active learning and random sampling on the 39-bus system for the oscillatory stability classification task using ANN

In Table 2.9 we summarize the accuracy of predictors on the oscillatory stability tasks and in Table 2.10 we include accuracy on the voltage stability tasks after 300 points have been included in $\boldsymbol{D}_{\mathrm{L}}$.

Table 2.9 Accuracy Results on Oscillatory Stability Task

DATA SET	ANN		SVM	
	ACTIVE LEARNING	RANDOM	ACTIVE LEARNING	RANDOM
9-BUS	**99.9%**	**99.7%**	**100%**	**99.2%**
39-BUS	**98.5%**	**97.7%**	**99.4%**	**98.2%**

Table 2.10 Accuracy Results on Voltage Stability Task

DATA SET	ANN		SVM	
	ACTIVE LEARNING	RANDOM	ACTIVE LEARNING	RANDOM
9-BUS	**99.8%**	**99.5%**	**99.8%**	**96.8%**
39-BUS	**97.6%**	**96.6%**	**99.2%**	**96.9%**

2.5.5 Conclusion

The following conclusions were reached:

- A significant improvement in accuracy can be obtained from a reduced data set by using active learning to select a subset of data to learn from. In the case of an existing labeled data set the presented methodology can be used to filter out redundant data thus reducing the computational burden of training data mining tools.

- When only a set of power system OCs is available without their associated stability states, and precise values of *DR* and VS_{margin} must be obtained through time domain simulation, the proposed method may be used to select which OCs to

query in order to create the most adequate data set to learn from. This may significantly reduce the complexity involved in time domain simulations.

- The performance improvement observed on more complex power system tasks is greater than on simpler tasks. The experiments also show that for simpler tasks the used ANNs are less sensitive to data set selection than SVMs, as can be seen from the random sampling results in Tables 2.28 and 2.24. On more complex tasks, and in all examined cases employing active learning, higher accuracy can be obtained using SVMs.

- We conclude that in the examined cases using active learning to pick which system OCs are simulated in the time domain, and afterwards used for training will lead to more accurate stability assessment, decrease the computational complexity, or both.

2.6 Feature Selection and Optimal PMU Placement

2.6.1 Introduction

In previous sections, the RTs were fed with voltage and current phasors measured at all buses. An underlying assumption is that almost every substation is equipped with a PMU. In practice this is not economically feasible, since the installation of PMUs and corresponding telecommunications path is very costly. A reasonable approach may be to install only a limited number of PMUs at the most critical substations. The problem of finding the optimal PMU location is equivalent to selecting the best reduced set of RT input features without a significant degradation in RT performance.

2.6.2 Variable Importance Derived from Decision Trees

Ideally, the optimal solution could be obtained through an exhaustive trial and comparison of all possible feature combinations. However this approach is too computationally involved. In this work we are proposing a different approach and the idea comes from a unique property of the RT model structure [41] [49]. The topology of the 9-bus system OSM-RT derived in Chapter 3 is shown in Figure 2.29. Each node has been split by an input variable, and the variable is selected as the splitter because it is the most powerful variable among all candidate features that can best split the node. The variables gain credit towards their importance by serving as primary splitters at a node, or as back-up splitters (surrogates) to be used when the primary splitter is missing. By summarizing the variables' contribution to the overall tree when all nodes are examined, the *Variable Importance* (*VI*) can be obtained.

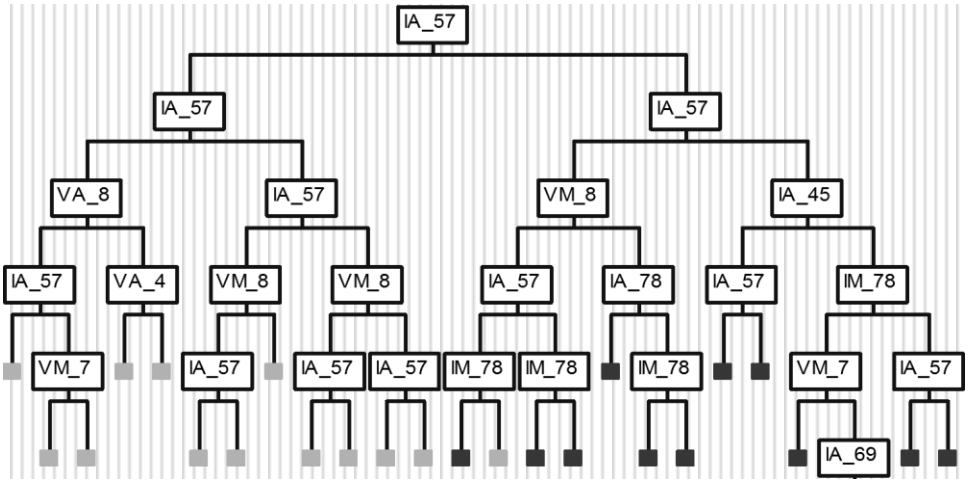


Figure 2.29 OSM-RT topology and node splitters of the 9-bus system

To calculate the *VI*, search all splits $s \in S$ on variable x_m at each tree node $t \in T$, and find the split s^*_m that gives the largest decrease in regression R [11]:

$$\Delta R(s^*_m, t) = \max_{s \in S} \Delta R(s,t) \qquad 6.1$$

Suppose s^* is the best of s^*_m , and $\tilde{s}_m$ is the split on variable x_m that has the best agreement with s^* in terms of partitioning cases, the measure of importance of variable x_m is defined as:

$$VI(x_m) = \sum_{t \in T} \Delta R(\tilde{s}_m, t) \qquad 6.2$$

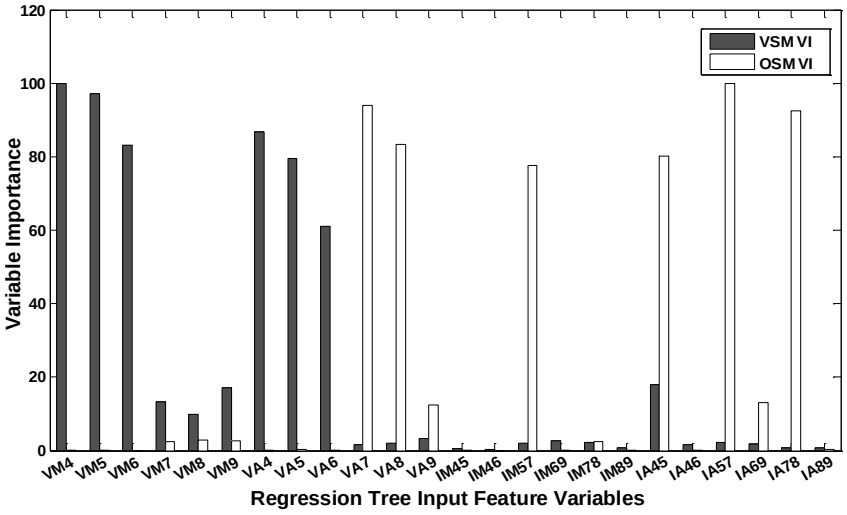


Figure 2.30 IEEE 9-bus system VSM-RT and OSM-RT variable importance

Figure 2.30 shows the computed *VI* for the OSM-RT and VSM-RT of the 9-bus system derived in previous sections. The actual measures of importance have been normalized so that the most important variable has a *VI* of 100.

2.6.3 Combined Bus Ranking

The intuition behind *Combined Bus Ranking* (CBR) is as follows: The overall contribution of each bus to the oscillatory and voltage stability evaluation can be

quantified by combining the importance of variables measured at this bus.

Mathematically the CBR of Bus i can be expressed as:

$$CBR_i = \sum_{x \in i} VI_{OSM-RT}(x) + \sum_{x \in i} VI_{VSM-RT}(x) \qquad 2.15$$

where X is the vector of RT input variables, x is the individual variable belong to X, and $VI(x)$ is its importance. By specifying $x \in i$, only the variables measured at Bus i will be counted.

2.6.4 Optimal PMU Locations

A ranking list of the bus contributions can be obtained by sorting the CBR values from high to low. The optimal PMU locations will be suggested by selecting the top ranked buses from the list. In this work the CBR of top ranked buses were computed by considering only the primary splitters, because the surrogate variables that appear to be important but rarely split nodes are almost certainly highly correlated with the primary splitters and contain similar information. Once the top ranked buses were selected, the standard *VI* considering both primary and surrogate splitters were used to rank the remaining buses. In Table 2.11, the CBR for the WECC 179-bus system was calculated and top 10 buses are listed. Also shown in the table are the 10 buses with the lowest CBR.

Suppose that a number of 4 to 20 PMUs will be installed in the WECC system. By placing them at the top ranked buses of Table 2.11, the resulting RT prediction accuracy for unseen OPs is summarized in Figure 2.31. The RT performance using the

measurements from the lowest ranked buses is also presented for the purpose of comparison.

Table 2.11 WECC 179-Bus System Combine Bus Ranking

Top Ranked Buses			Lowest Ranked Buses		
Rank	Locati	CBR	Rank	Locati	CBR
# 1	Bus	100.8	# 170	Bus	0.31
# 2	Bus	100.2	# 171	Bus	0.28
# 3	Bus	38.27	# 172	Bus	0.24
# 4	Bus	18.47	# 173	Bus	0.12
# 5	Bus	13.99	# 174	Bus	0.11
# 6	Bus	12.73	# 175	Bus	0.02
# 7	Bus	12.52	# 176	Bus	0.02
# 8	Bus	8.48	# 177	Bus	0.01
# 9	Bus 9	8.44	# 178	Bus	0.01
# 10	Bus	8.24	# 179	Bus	0.00

As shown in Figure 2.31 in contrast with the RTs fed with measurements from the lowest ranked buses, those constructed using the measurements from top ranked buses have exhibited better performance. Another conclusion could be made by comparing the R^2 of Figure 2.31 with Figure 2.17: almost identical RT prediction R^2 was achieved by using the reduced set of measurements from the PMU locations suggested by CBR. Last but not least, there is a huge decrease of the complexity in RT training since fewer features are used. The training time of the 179-bus RTs has been reduced from about 3 minutes to less than 30 seconds.

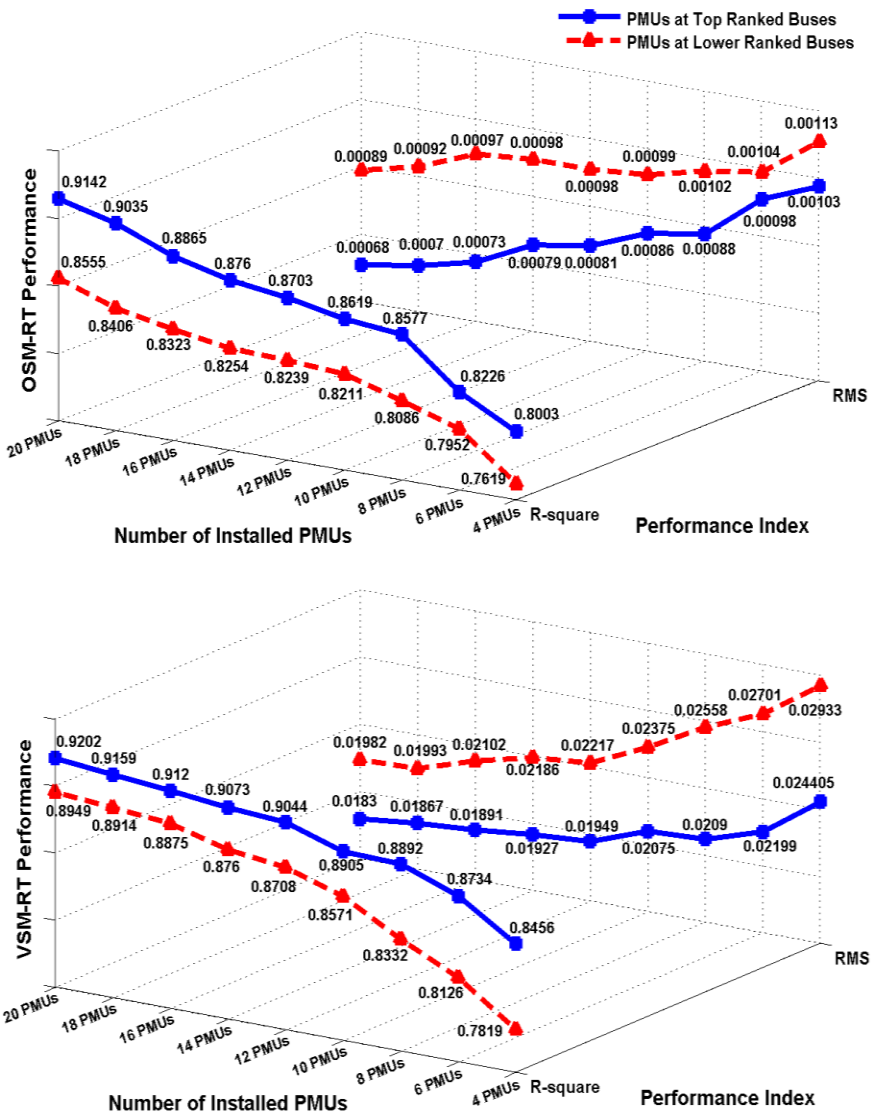


Figure 2.31 RT performance considering different PMU pacements in the 179-bus system

2.6.5 Summary

A novel methodology for optimal placement of PMUs in a power network is proposed. The variable importance of each DT input features can be derived from CART and utilized to rank the importance of network substations for stability assessment applications. The combined bus ranking derived from RT variable importance is used to

suggest optimal PMU locations. The performance of DTs using synchrophasor measurements from a limited number of PMUs was checked. Test results show that the measurements from reduced locations can still lead to satisfactory RT prediction accuracy.

2.7 Measurement-based Approach Applied to Field PMU Data

2.7.1 Introduction

Power system oscillatory stability assessment is the task of monitoring the rotor angle synchronism of generators at different locations. The recent trend in the electric power industry is to interconnect transmission lines linking small autonomous systems into large integrated systems, some of which span the entire continent. For example, in the United States and Canada generators which are located thousands of miles apart are operated simultaneously and synchronously. As a consequence inter-area electromechanical oscillations are becoming a more common occurrence. Since modern systems are optimally run near their stability threshold, the estimation of the distance of an operating point from instability region is critical for stable operation.

Traditional oscillatory stability assessment methods may not satisfy the online monitoring requirements because: 1) they are based on time-domain model simulations which are computationally intensive and time-consuming; 2) they use data collected from Supervisory Control and Data Acquisition (SCADA) systems, or state estimation functions, both of which are updated relatively infrequently.

With improved data acquisition technology, such as temporal synchronization of measurements at different locations, it may be possible to detect the onset of instability

more accurately. The ability of synchrophasors to capture system-wide dynamics shows their potential in real-time system stability monitoring applications.

The advantages of a measurement-based approach include lower computational complexity, reduced knowledge requirements about system model parameters, and the potential to provide system stability assessment in real time. Most measurement-based approaches use appropriate signal processing or spectral analysis techniques to extract information from periodically collected power systems data. One such method is Prony analysis, which has been investigated by Kumaresan et al. in exponentially damped signal analysis [50]-[51], and later applied to power systems by Hauer et al. in oscillatory stability assessment [52]-[53]. Prony analysis is a powerful tool for mode parameter identification of electromechanical oscillations. However, if noise is present in measurements it performs poorly [51]. Another shortcoming of Prony's method is that it is only suitable for transient, or ringdown, data analysis, and cannot be applied to ambient data such that the system is excited by random load variations [54]. Therefore it is termed a ringdown analyzer that operates specifically on transient portion of a measured signal.

Alternatively, several mode meters, such as the Yule-Walker method [54], autoregressive moving average (AR/ARMA) model [55], and subspace estimation method [56]-[58], have been extensively studied in the past two decades in order to estimate mode parameters from both ambient data and transient data. While in previous efforts accurate estimation has been achieved for oscillation mode frequency, the problem of identifying mode damping, a more important task in terms of stability assessment, has not been satisfactorily resolved, although encouraging results were reported under certain test scenarios [52]-[59].

In this work, a data mining approach is used to estimate oscillatory stability in real time. The decision tree (DT) method proposed by Breiman et al. is employed to map system operating points at each instant to one of several pre-defined stability states. Compared to previous research, the proposed approach casts the task as a multi-class classification problem, as detailed in Section 7.3. In Section 7.4 we show the results of the proposed method using the IEEE 39-bus test system. Finally, the data mining approach is evaluated on field PMU measurements from Salt River Project (SRP), a public electrical utility in Phoenix, Arizona, U.S.A.

2.7.2 Theoretical Formulation

2.7.2.1 Oscillatory Stability Assessment

Oscillatory stability is associated with Hopf Bifurcation. An instability event occurs when, following a small disturbance, the damping torques are insufficient to bring the system back to a steady-state operating condition, identical or close to the pre-disturbance condition.

As shown in Figure 2.32, power system oscillations may be classified into four categories in terms of frequency: 1) speed governor band, from 0.01 to 0.15 Hz; 2) inter-area electromechanical band, from 0.15 to 1.2 Hz; 3) local electromechanical band, from 1.2 to 5 Hz; and 4) torsional dynamics band, from 5 to 15 Hz. This work focuses on the second category: the low-frequency inter-area oscillations.

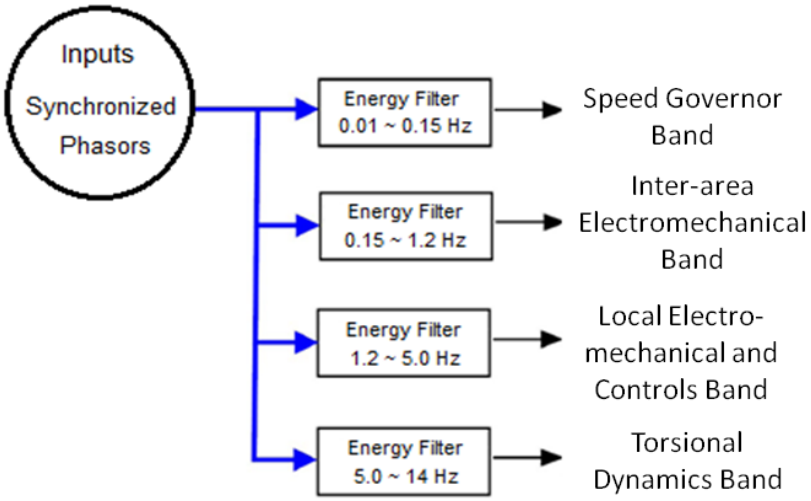


Figure 2.32 Typical frequency band of different oscillation types

2.7.2.2 Mode Identification without System Model

Traditionally, the stability of inter-area oscillations is evaluated through modal analysis of the system's non-linear differential algebraic equations (DAE) using detailed system model parameters, as detailed in Section 2.2. The inter-area oscillation modes that carry significant amount of energy but with insufficient DR are critical among all modes and need to be closely monitored.

In contrast to the model-based approach, the measurement-based approach does not require detailed system model information. Recent efforts take measurements from different locations during the same period of time, and identify oscillation mode parameters through signal processing techniques. The mode parameters that can be estimated include frequency, f, damping, σ, amplitude, A, and phase, θ, as shown in Figure 2.33.

There are three types of relevant power system measurements: ambient data, transient (ringdown) data, and probing data. Figure 2.34 shows the ambient and ringdown measurements. The probing data is beyond the scope of this work and will not be

discussed further. For ambient data an AR/ARMA model is used to derive mode parameters while Prony analysis is used for ringdown data.

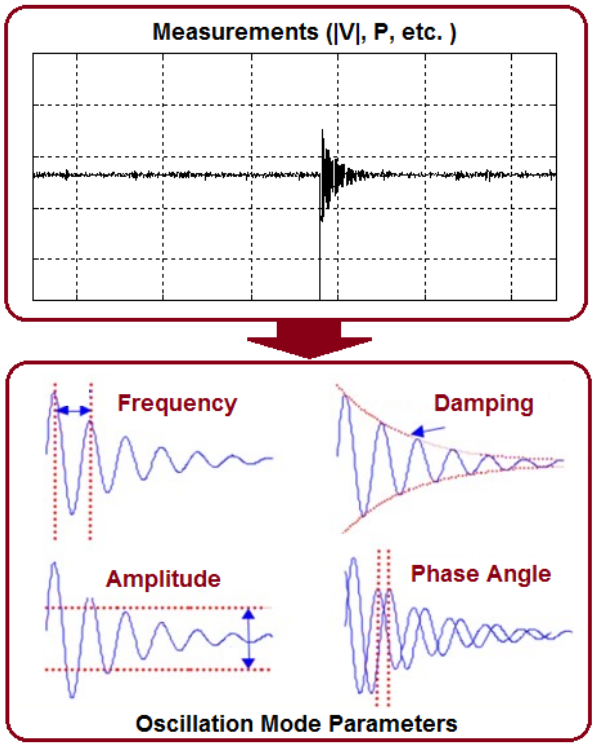


Figure 2.33 Mode parameters identified from power system measurements

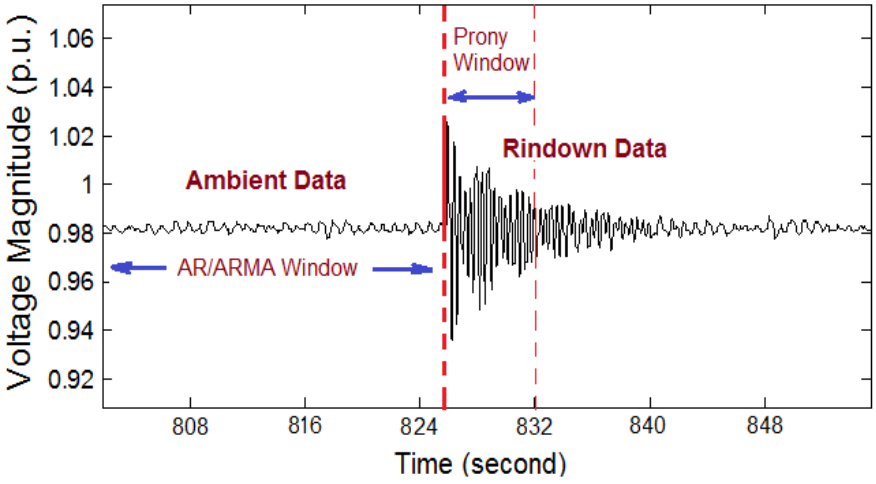


Figure 2.34 Ambient/ringdown signals and corresponding analysis windows

2.7.2.3 Oscillatory Stability Evaluation Using Prony's Method

Assume $x(t)$ is the state of a linear time-invariant (LTI) dynamic system and $u(t)$ is

the input to the system. The evolution of the state is expressed by:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \tag{2.16}$$

where A and B are constant matrices.

Suppose the LTI system is brought to an "initial" state $x(t_0)=x_0$, at time t_0, by means of the input $u(t)$. Then, if the input is removed and there are no subsequent inputs or disturbances to the system, it will "ring down" according to a differential equation of form:

$$\frac{dx(t)}{dt} = Ax(t) \tag{2.17}$$

where x is the state of the system and n is the number of components in x (i.e., the order of the system). Let x_i, p_i, q_i be respectively the eigenvalues, right eigenvectors, and left eigenvectors of matrix A (of size $n \times n$). The solution to Eq. 2.17 can be expressed as the sum of n components:

$$x(t) = \sum_{i=1}^{n} (q_i^T x_0) p_i e^{\lambda_i t} \tag{2.18}$$

Let $y(t)$ be the system response. As we have assumed the system is an LTI system, $y(t)$ can be expressed in the form:

$$\frac{dy(t)}{dt} = Cx(t) + Du(t) \tag{2.19}$$

where C and D are constant matrices. If the input is removed ($u(t)=0$), then Eq. 2.19 simplifies to:

$$\frac{dy(t)}{dt} = Cx(t) \tag{2.20}$$

Suppose an observed record for $y(t)$ is a signal consisting of N samples $y(t_k)=y(k)$, k=0, l, ..., N-l, that are evenly spaced by an amount Δt. Similarly to the Fourier series, Prony analysis builds a series of damped complex exponentials or sinusoids. Through Prony's method, valuable information such as oscillation frequency, amplitude, mode phase and decay time could be extracted from a uniformly sampled signal.

Prony's method and its recent extensions are designed to directly estimate the parameters for the exponential terms in Eq. 7.3 by fitting a function:

$$\hat{y}(t) = \sum_{i=1}^{L} A_i \, e^{\sigma_i t} \cos(2\pi f_i t + \phi_i) \tag{2.21}$$

After some manipulation utilizing Euler's formula, the following result is obtained. This allows for more direct computation of terms:

$$\hat{y}(t) = \sum_{i=1}^{L} A_i \, e^{\sigma_i t} \cos(2\pi f_i t + \phi_i) = \sum_{i=1}^{L} \frac{1}{2} A_i e^{\pm j\phi_i} e^{\lambda_i t} \tag{2.22}$$

where $\lambda_i = \sigma_i \pm j\omega_i$ are the eigenvalues of component i, σ_i are the damping coefficients, f_i are the frequency components, ϕ_i is the phase of component i, A_i is the amplitude of component i, and L is the number of damped exponential components.

The strategy for obtaining a Prony solution can be summarized as follows:

PSEUDO-CODE FOR PRONY'S ALGORITHM
1. Construct a discrete linear prediction model (LPM) that fits the record. 2. Find the roots of the characteristic polynomial associated with the LPM of step 1. 2. Using the roots of step 2 as the complex modal frequencies for the signal, determine the amplitude and initial phase for each mode.

These steps are performed in *z*-domain. For power system applications the eigenvalues would usually be translated to *s*-domain, consistent with Eq. 2.16 - Eq. 2.17

2.7.2.4 Oscillatory Stability Evaluation Using ARMA

ARMA and AR methods are a common parametric approach to spectral analysis while fast Fourier transform (FFT) methods are nonparametric approaches. The AR and ARMA models allow the direct estimation of the electromechanical oscillation modes.

The ambient noise is assumed to be relatively statistically stationary for a block of data for the frequencies of interest. For the ARMA model shown in Figure 2.35, the corresponding difference equations relating the input and output are:

$$u(n) + a_1 u(n-1) + \cdots a_N u(n-N)$$

$$= b_1 v(n) + b_1 v(n-1) + \cdots + b_M v(n-M) \qquad 2.23$$

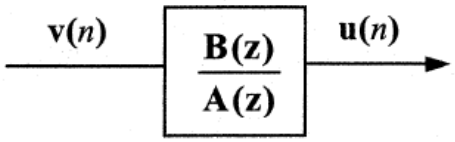


Figure 2.35 ARMA model with white noise at the input

2.7.2.5 Data Mining Approach

The DT algorithm has been used as a classification tool for online oscillatory stability estimation. The DT is created by sequentially splitting the training data set at each tree node, starting from the root. The node splitting rule is determined by searching all candidate attributes, and finding the split which gives the largest decrease in class impurity. A terminal node is reached when maximum purity has been achieved.

In the experimental section we compared results obtained using DTs with those obtained using feed-forward neural networks and support vector machines. Both techniques are well known for their powerful modeling and generalization capabilities in classification analysis.

In this work the commercial software MATLAB is used to implement the neural networks and support vector machines. Synchrophasors collected from PMUs are used as the input attributes to data mining tools.

2.7.3 Proposed Approach

2.7.3.1 Framework

A framework of the proposed measurement-based scheme has been previously shown in Figure 2.2. The model-based approach, which was investigated by the authors in [29] and [41], is also shown in the figure for comparison purposes.

For each power system, several stability thresholds are specified with respect to the typical damping ratio of the critical oscillation mode (DR_{crit}), and a set of stability states is defined accordingly. As shown in Figure 2.36, for the given oscillatory stability thresholds $\boldsymbol{\theta_{STB}}$ and $\boldsymbol{\theta_{ALT}}$ ($\boldsymbol{\theta_{STB}} > \boldsymbol{\theta_{ALT}}$), operating points (OPs) will be labeled as 'Good' if they satisfy $DR_{crit} \geq \boldsymbol{\theta_{STB}}$ $VS_{margin} > 40\%$; "Fair" if they satisfy $\boldsymbol{\theta_{STB}} > DR_{crit} \geq \boldsymbol{\theta_{ALT}}$; "Alert" if they satisfy $\boldsymbol{\theta_{ALT}} > DR_{crit} \geq 0$; and 'Unstable' when $0 > DR_{crit}$. In practice, the values of $\boldsymbol{\theta_{STB}}$ and $\boldsymbol{\theta_{ALT}}$ are usually around 10% and 5% respectively.

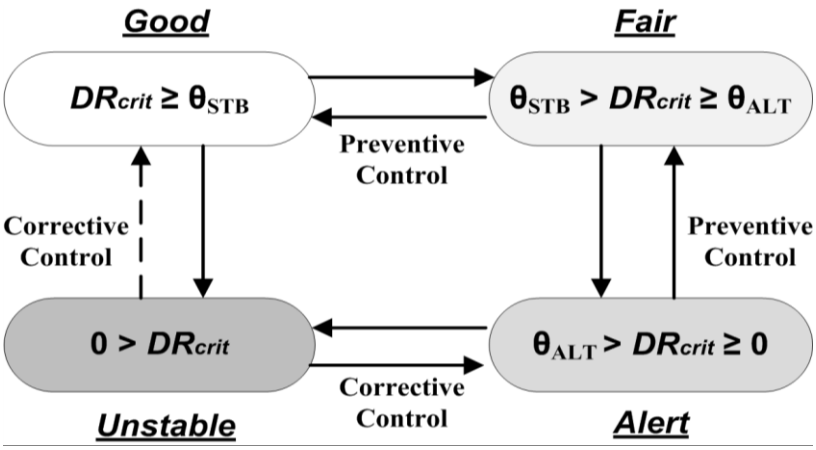


Figure 2.36 Classification of oscillatory stability states

2.7.3.2 Mode Parameter Identification

Figure 2.37 illustrates the online application procedures of the proposed scheme. As the first step, a knowledge base needs to be created in order to train the classification tree. Included in the knowledge base are the input PMU measurements at each system operating point (OP), as well as the oscillatory stability state corresponding to each OP.

The procedure is initialized with a window scanning of the historical PMU measurements. An Oscillation Detector (OD) is designed to detect whether a transient event occurs by monitoring the presence of a sudden deviation in recorded measurements. If there are no abnormal changes, the OD suggests that the system is

operated under a steady state, and an AR/ARMA model is employed to estimate the mode parameters in a sliding window manner. The required window length for ambient data analysis varies from 5 minutes to half an hour, depending on the variation level of system loads. If a sudden deviation is detected, but only limited to fewer than 5 data points, the corresponding measurements are considered outliers caused by sensor or communication error, and are discarded from consideration. If a continued deviation has been observed, the OD will report that a transient process is potentially occurring, and Prony analysis is applied to scan the transient data using a sliding window with a length of 5 to 10 seconds, depending on the critical mode frequency of the inter-area electromechanical oscillation.

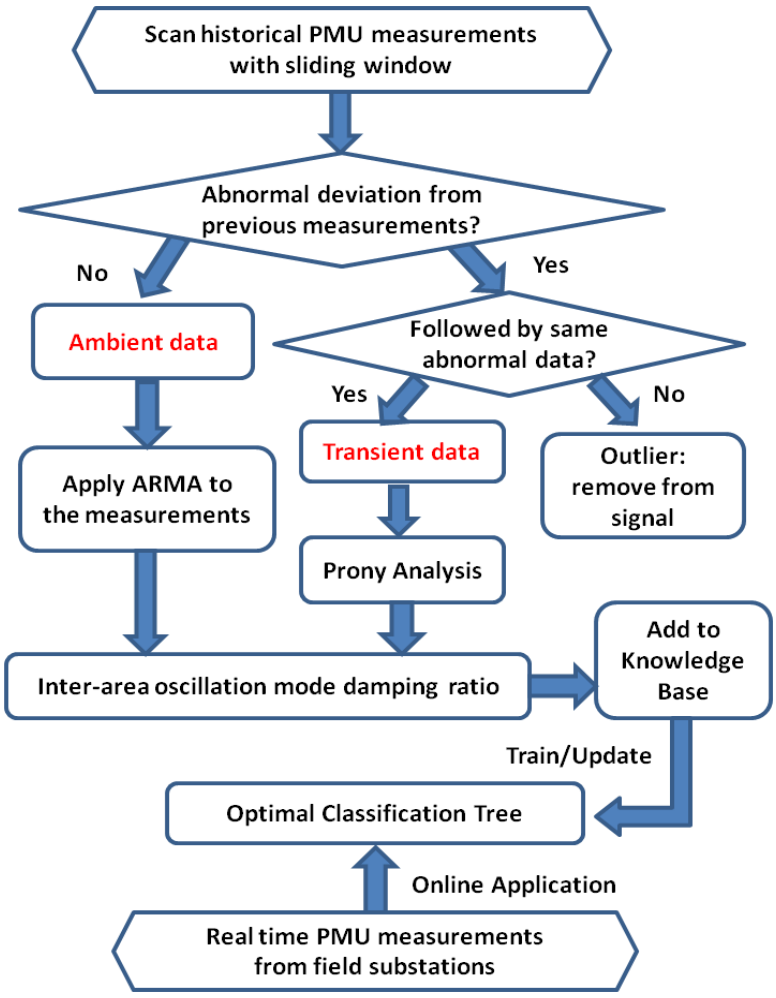


Figure 2.37 Online application of the proposed scheme

2.7.3.3 Classification Tree for Stability Assessment

In order to overcome the limitations of Prony and ARMA methods, the ringdown data is pre-processed using a low-pass filter, and the window length of AR/ARMA model is sufficiently large to assure accurate estimation. Once a sufficient number of cases have been accumulated, the knowledge base is used to train the classification trees. The derived optimal DT is then applied online. As shown in Figure 2.37, new PMU measurements are dropped down through the tree to predict the oscillatory stability status of each OP in real time.

One of the key challenges of embedding DTs in online applications is the problem of evolving system operating conditions. Due to variations in system generation and loading patterns, and changes in system topology, the DR_{crit} of inter-area electromechanical oscillations may also change. To deal with this eventuality, the classification tree derived in CART needs to be periodically refreshed in order to reflect the most current system operating conditions. This is done by updating the knowledge base using the most recent PMU measurements, and re-training the DT.

2.7.4 Case Study

The IEEE 10-machine 39-bus test system (New England system) is used to implement the proposed scheme. Its one-line diagram is shown in Figure 2.7. Firstly the oscillation mode parameters are estimated through model-based eigenvalue analysis. They will be used later to validate the results of the measurement-based approach.

The 39-bus system is modeled in MATLAB/SIMULINK. As shown in Figure 2.38, the Network Solution Module initializes the time-domain simulation, calculates power flow, and provides real time network solutions using dynamic model parameters.

The low-frequency oscillation modes with insufficient DRs are listed in Table 2.12. They are obtained from model-based eigenvalue analysis of the IEEE 39-bus system. Also listed in this table are the dominant generators that participate in the correlated oscillation modes.

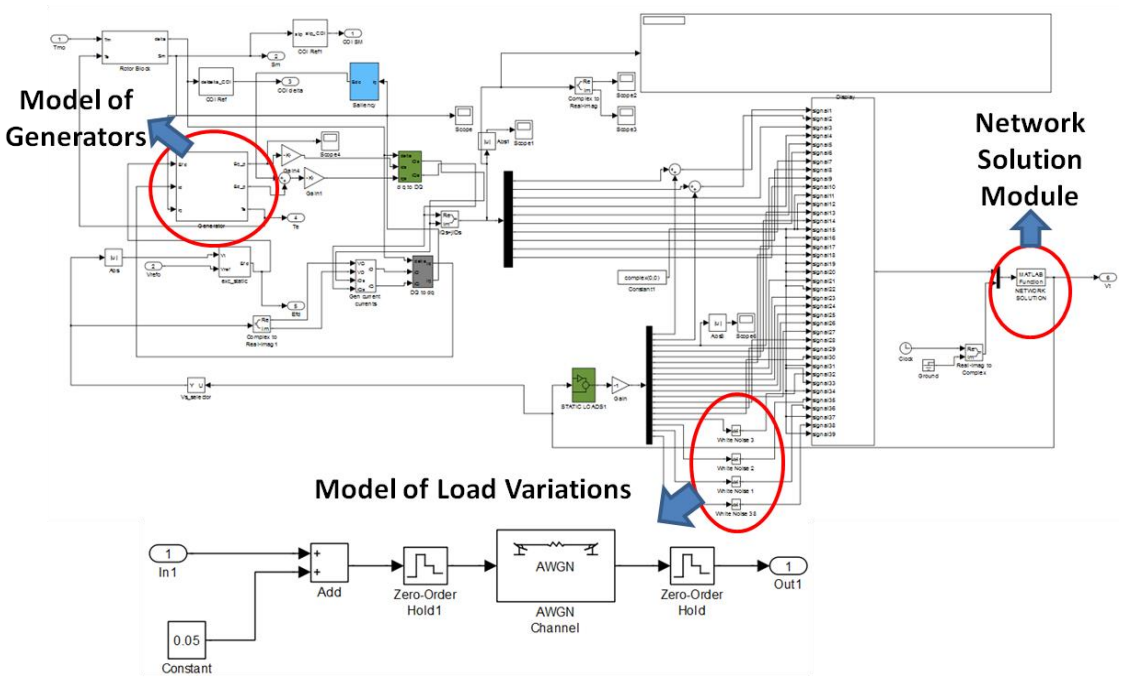


Figure 2.38 Simulink model of the IEEE 39-bus test system

In this work the Mode #5 with a frequency of 0.58 Hz is targeted for monitoring. To simulate the load variations, Gaussian noise with Mean = 0.05 and Signal to Noise Ratio (SNR) = 20 dB has been introduced to four system loads. The time-domain simulation has been performed for 15 minutes. To create transient signal, a fault that caused the line between Bus 26 and Bus 28 to trip has been simulated. The fault occurred at t = 700s, and lasted for 0.02s. The resulting measurements from all system buses are recorded. In particular, the voltage magnitudes and phase angles at Bus 7 and Bus 39 are shown in Figure 2.39 and Figure 2.40.

Table 2.12 Low-Frequency Oscillation Modes Obtained from Model Initialization

	Mode #1	Mode #2	Mode #3	Mode #4	Mode #5
Frequency (Hz)	**1.21**	**1.13**	**1.03**	**0.96**	**0.58**
Damping Ratio (%)	**1.06**	**4.62**	**1.87**	**8.81**	**6.35**
Dominant Generator	**G1, G3**	**G4, G6**	**G3**	**G10**	**G2**

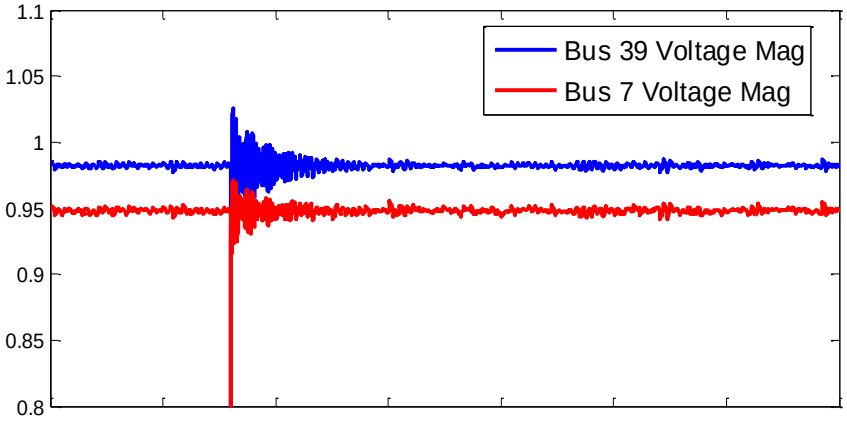


Figure 2.39 Voltage magnitude signals

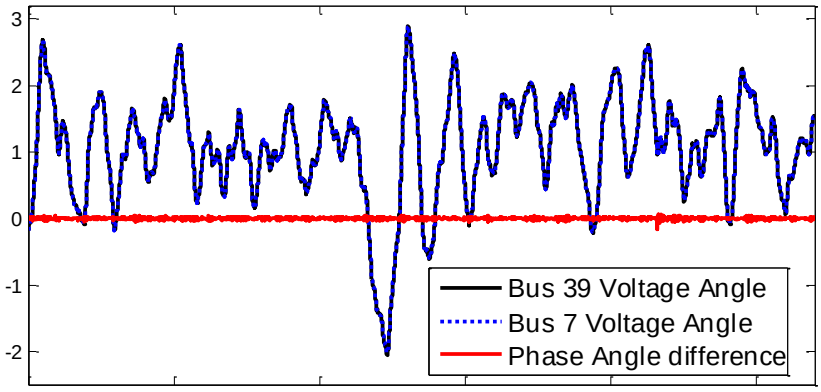


Figure 2.40 Phase angles and their difference

Prony analysis has been applied to the Bus 39 voltage magnitude signal during the transient process. The sliding window has a length of 5 seconds and the Prony model order is set to be N=30.

The AR model has been applied to the phase angle difference between Bus 7 and Bus 39, which is shown in Figure 2.40. The ambient data before the fault are treated using a sliding window with a length of 10 minutes. Different model orders have been deployed to compare the results. The mode damping ratios estimated by AR of order N=60 are drawn in Figure 2.41. The Mean of the damping ratios estimated with different model orders have been summarized in Table 2.13. Table 2.13 shows that the mode frequency estimated from AR and Prony are very close to the eigen-analysis results in Table 2.12. The damping ratio estimated by AR is approaching the actual value when increasing the model order. The DR estimated by Prony analysis is different due to the change in system topology.

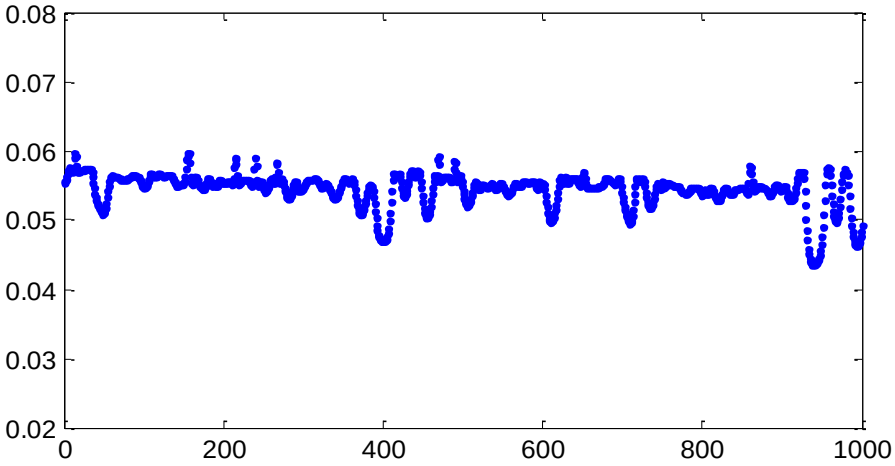


Figure 2.41 Damping ratios estimated from ambient measurements

Table 2.13 Estimate Mode #5 by Applying AR to Ambient Data

	Order	Frequency (Hz)	Damping Ratio (%)
AR	*N*=30	0.5622	4.391
	N=60	0.5819	5.637
	N=90	0.5753	6.224
Prony	*N*=30	0.5787	5.185

By varying the load disturbance level and fault scenario, the time-domain simulations have been replicated and a total of 4938 OPs with their corresponding stability states are included in the knowledge base. A classification tree has been developed in CART using 80% of the cases, and the rest 20% has been used in new case testing. The classification accuracy is evaluated as follows,

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}. \tag{7.1}$$

The DT accuracy is summarized in Table 2.14. It is observed that an overall prediction accuracy as high as 98.38% has been achieved.

Table 2.14 Classification Tree Performance

	Good	Fair	Alert	Accuracy
Good	610	8	2	0.9839
Fair	3	349	1	0.9887
Alert	0	2	13	0.8667
Accuracy	0.9951	0.9721	0.8125	0.9838

2.7.5 Application to Field PMU Measurements

The filed PMU measurements received from a public electrical utility in Phoenix, Arizona, U.S.A., the Salt River Project (SRP), have been used to evaluate the proposed scheme. The data include synchronized voltage and current phasor measurements, under both ambient and transient conditions. The transient data recorded two consecutive brake insertion applications at a major transmission substation. The voltage magnitude measured at another substation has been divided into two 5-minute signals as shown in Figure 2.42. Each of the signals includes one transient process.

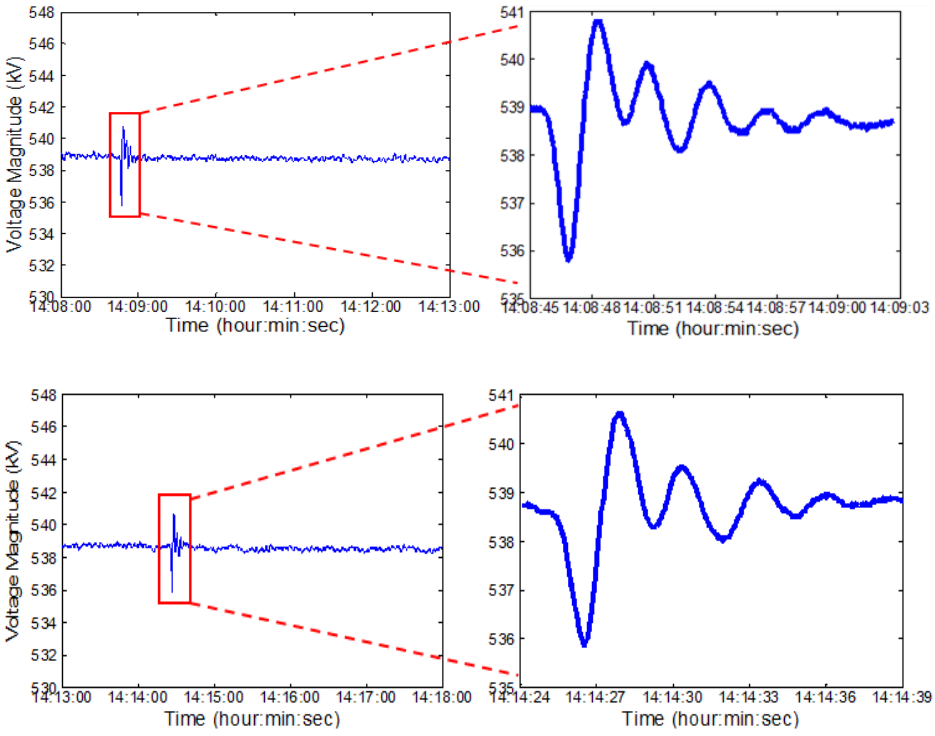


Figure 2.42 Field voltage magnitude measurements from PMUs

A knowledge base has been created by applying the same procedure introduced in Section 2.7.4 to the field measurements from PMUs. The resulting DT performance has been summarized in Table 2.15. Two other data mining tools, the artificial neural

network (ANN) and support vector machine (SVM), have also been used to compare the results.

From Table 2.15, the DT-based prediction model achieved similar accuracy to other data mining tools. Compared to some "black-box" models, however, the DT provides a more transparent structure with a clearer cause-effect relationship. Its piece-wise structure and node splitting rules enable the identification of the critical variables and thresholds that should be analyzed to gain insight into the oscillatory stability of a system.

Table 2.15 Results Comparison

Data Mining Tools	Misclassification Rate			Overall Accuracy
	Good	Fair	Alert	
DT	0.0219	0.0667	0.0737	0.9739
ANN	0.0034	0.0902	0.1852	0.9873
SVM	0.0008	0.0738	0.0602	0.9940

2.8 Summary

The use of Decision Trees for online stability assessment without the knowledge of system model parameters has been investigated in this work:

- The proposed scheme is a measurement-based method that complements the traditional model-based approach. It is particularly useful when system model parameters are not readily available;
- The proposed approach is able to provide control center operators with real time support by making use of the quickly updated PMU measurements;

- Once trained using the knowledge base, the DT-based predictor can achieve high accuracy in online oscillatory stability estimation;
- The data mining tools are capable of reflecting the evolving system operating conditions when the most recent PMU measurements and corresponding knowledge base are used;
- When the results are compared with other data mining tools such as ANN and SVM, it is observed that almost identical prediction accuracy can be achieved.

2.9 Conclusions

In this project the approach of using classification and regression trees to predict power system stability behavior from PMU measured synchrophasor data is explored. The following conclusions were reached in this work:

- The DT-based data mining model provides an accurate assessment of the stability status of each system operating point. Compared with some other data mining tools, using DTs it is possible to identify the critical variables and thresholds that need to be analyzed to gain insight into the stability margin of a power system;
- Encouraging results were obtained through performance examination using the proposed knowledge base generation methodology. With a sufficiently captured system stability behavior, the DT model can predict the system oscillatory and voltage stability status with high accuracy;

- The CT classification accuracy is related to how the tree is developed, and the setting for minimum parent node cases can alter the shape of the resulted tree as well as its performance;

- According to the test results, an RT model is fast enough to process PMU measurements, and it is robust to handle measurement errors that are within 1% TVE. The RT sensitivity to system topology variation becomes less distinct in large sized networks and under mild changes in topology. The proposed DT update methodology enables seamless online stability monitoring;

- A significant improvement in accuracy can be obtained from a reduced data set by using active learning to select a subset of data to learn from. In the case of an existing labeled data set the presented methodology can be used to filter out redundant data thus reducing the computational burden of training data mining tools. The performance improvement observed on more complex power system tasks is greater than on simpler tasks.

- The combined bus ranking derived from RT variable importance is used to suggest optimal PMU locations. Test results show that the measurements from a reduced number of locations may still lead to satisfactory RT prediction accuracy;

- The proposed measurement-based oscillatory stability assessment method complements the traditional model-based approach. It is particularly useful when system model parameters are not readily available;

- The data mining tools are capable of reflecting the evolving system operating conditions when the most recent PMU measurements and corresponding

knowledge base are used;

- When the results are compared with other data mining tools such as ANN and SVM, it is observed that almost identical prediction accuracy can be achieved by using DT. In addition, the DT model provides a more transparent structure with a clearer cause-effect relationship.

3 Data Mining for Online Dynamic Security Assessment using PMU Measurements

3.1 Introduction

Dynamic security assessment [60] can provide system operators important information regarding the transient performance of power systems under various possible contingencies. By using the real-time or near real-time measurements collected by phasor measurement units (PMUs), online DSA can produce more accurate security classification decisions for the present OC or imminent OCs. However, online DSA still constitutes a challenging task due to the computational complexity incurred by the combinatorial nature of $N-k$ ($k=1,2,\cdots$) contingencies and the massive scale of practical power systems, which makes it intractable to perform power flow analysis and time domain simulations for all contingencies in real-time.

The advent of data mining techniques provides a promising solution to handle these challenges. Cost-effective DSA schemes have been proposed by leveraging the power of data mining tools in classification, with the basic idea as follows. First, a knowledge base is prepared through comprehensive offline studies, in which a number of predicted OCs are used by DSA software packages to create a collection of training cases. Then, the knowledge base is used to train classification models that characterize the decision rules to assess system stability. Finally, the decision rules are used to map the real-time PMU measurements of pre-fault attributes to the security classification decisions of the present OC for online DSA. The data mining tools that have proven effective for DSA include decision trees [13][14][15][16][18][20], neural networks [61][62][63] and support vector machines [25][64][65]. More recently, fuzzy-logic techniques [22] and ensemble learning

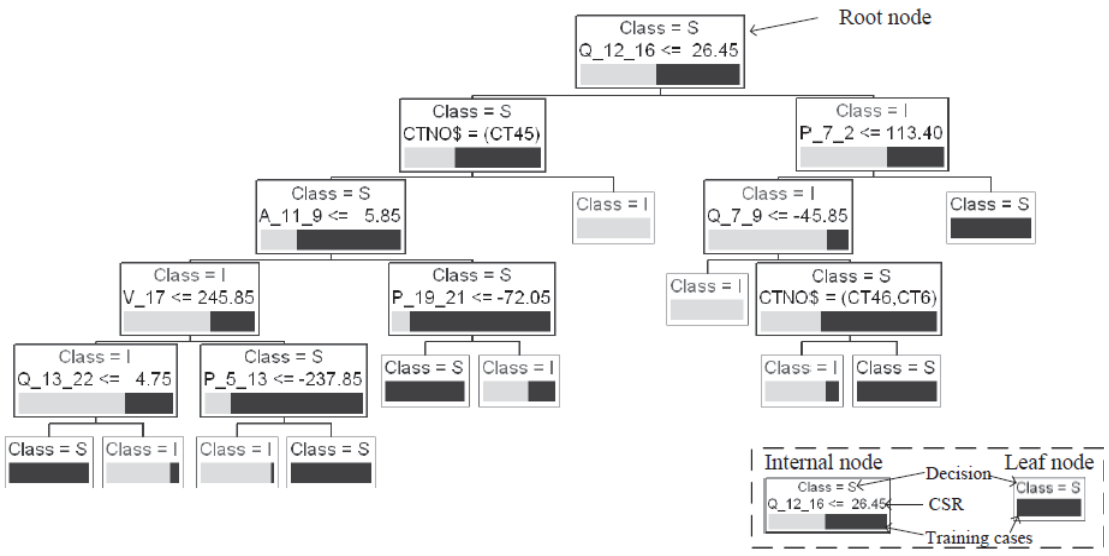


Figure 3.1 Fully-grown DT of height 5 for the WECC system using an initial knowledge base consisting of 481 OCs and three critical contingencies

techniques [19][66][67] have been utilized to enhance the performance of these data mining tools in security assessment of power systems. Among various data mining tools, DTs have good interpretability (or transparency) [68], in the sense that the secure operating boundary identified by DTs can be characterized by using only a few critical attributes and corresponding thresholds. As illustrated in Fig. 3.1, a well-trained DT can effectively and quickly produce the security classification decisions for online DSA, since only a few PMU measurements of the critical attributes are needed. The high interpretability of DTs is amenable to operator-assisted preventive and corrective actions against credible contingencies [69]. However, as discussed in [23], there exists an "accuracy versus transparency" trade-off for data mining tools. In order to obtain a more accurate classification model from DTs, one possible approach is to use an ensemble of DTs at the cost of reduced interpretability. Examples of ensembles of DTs for DSA are the multiple optimal DTs [18], random forest [19] and boosting DTs [66].

When applying data-mining-based approaches to online DSA with PMU measurements, there are two main issues that can compromise the performance of the classification model trained offline, as listed below:

- The realized OCs in online DSA can be dissimilar to those in the initial knowledge base prepared offline, since the predicted OCs might not be accurate and the OCs can change rapidly over time. Further, it is possible that a system topology change may occur during the operating horizon due to the forced outage of generators, transformers and transmission lines.
- In online DSA, PMU measurements can become unavailable due to the unexpected failure of the PMUs or phasor data concentrators (PDCs), or due to loss of the communication links.

However, there have been limited efforts directed towards handling OC variations and topology changes. In the scheme proposed in [18], when the built DT fails to classify the changed OCs correctly, a new DT is built from scratch or a sub-tree of the DT is replaced by a newly built corrective DT. Aiming to deal with possible topology changes, references [62], [67] suggest creating an "overall" knowledge base that covers all possible system topologies and choosing the attributes that are independent of topology for data mining. Further, reliable PMU measurement is usually assumed in literature, and the issue of missing PMU measurements in online DSA has not been considered.

To develop a robust data-mining-based online DSA scheme, the initial knowledge base and the classification model have to be updated in a timely manner to track these changed situations. Therefore, the two main objectives of our study in this project are:

- to develop data-mining-based online DSA schemes that are robust to power system dynamics, including OC variations and topology changes;
- to develop data-mining-based online DSA schemes that are robust to WAMS failures, including missing PMU measurements.

To these ends, state-of-the-art adaptive ensemble DT learning techniques have been developed and applied to online DSA, and shown to be effective through case studies with practical test systems. In what follows, technical background of adaptive ensemble DT learning is first introduced in Section 3.2. Then, the proposed approaches are presented in Section 3.3 and Section 3.4, respectively.

3.2 Background on Adaptive Ensemble DT Learning

The data-mining framework for DSA was originally developed in [13], in which DTs were introduced to perform DSA for power systems. A DT, as illustrated in Fig. 3.1, is a tree-structured predictive model that maps the measurements of an attribute vector $\mathbf{x}$ to a predicted value $\hat{y}$. When DTs are used for online DSA, the attribute vector can consist of various PMU-measured variables and other system information, and the binary decision given by DTs represents the security classification decision of an OC for a critical contingency (e.g., $\hat{y} = +1$ represents the insecure case, and $\hat{y} = -1$ for the secure case). Usually, bus voltage phase angles, bus voltage magnitudes and branch power/current flows that are directly measured by PMUs are used as numerical attributes. Fig. 3.1 illustrates the numerical and categorical attributes used in a trained DT, in which an attribute with initial 'V' stand for a bus voltage magnitude, the attributes with initials 'P', 'Q', and 'A' stand for an active power flow, a reactive power flow, and a voltage

phase angle difference between two buses, respectively (the bus numbers in attribute names are different from their real ones), 'CTNO$' stands for the index of contingency.

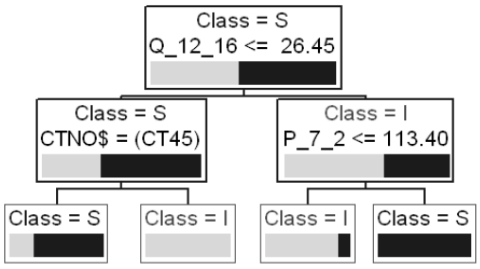
In a DT, each non-leaf node tests the measurement of an attribute and decides which child node to drop the measurements into, and each leaf node corresponds to a predicted value. As shown in Fig. 3.1, in a DT for DSA, the predictive value of each leaf node is either 'S' or 'I', in which 'S' stands for secure cases and 'I' for insecure cases. Fig. 3.1 also illustrates the training cases that fall into each node, by using dark bars for secure cases and bright bars for insecure cases. The number of non-leaf nodes along the longest downward path from the root node to a leaf node is defined as the height of a DT. Given a collection of training cases $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$, the objective of DT induction is to find a DT that can fit the training data and accurately predict the decisions for new cases. State-of-the-art DT induction algorithms are often based on greedy search. For example, in the classification and regression tree (CART) algorithm [11], the DT grows by recursively splitting the training set and choosing the critical attributes (numerical or categorical) and critical splitting rules (CSR) with the least splitting costs until some predefined stopping criterion (e.g., the size of tree or the number of training cases in a leaf node) is satisfied. In general, a fully-grown DT that accurately classifies the training cases might misclassify new cases outside the knowledge base. This feature of fully-grown DTs is usually referred to as "overfitting" [68]. In order to avoid overfitting, DTs are usually pruned by collapsing unnecessary sub-trees into leaf nodes. As illustrated in Fig. 3.1, in a pruned DT, some leaf nodes do not have pure training cases, which is a result of either tree pruning or early termination of tree growing [68]. By removing the nodes that may have grown based on noisy or erroneous data, the pruned DT is more resistant to

overfitting than a fully-grown DT without pruning, and thus can give more accurate security decisions.

A major advancement in DT-based DSA schemes was made in [20], in which the authors proposed to build a single DT to handle multiple contingencies, by using the index of contingencies as a categorical attribute of the DT. It is worth noting that a DT built by using such an approach can give the security classification decisions of an OC concurrently for all the critical contingencies in the knowledge base, which is more efficient and can identify the critical attributes that are independent of contingencies. For example, the DT in Fig. 3.1, using CTNO$ as a categorical attribute, can give security classification decisions of an OC for three critical contingencies, i.e., CT6, CT45 and CT46, at the same time, and the critical attributes $Q_{12,16}$, $P_{7,2}$, $Q_{7,9}$, $A_{11,9}$, $A_{12,19}$, $A_{5,12}$ and $P_{36,7}$ can give security classification decisions independent of contingence type for some cases.

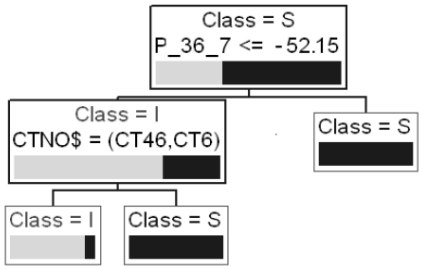
3.2.1 Small DTs

A small DT with tree height J is obtained by stopping the splitting of any leaf node if the downward path from the root node to that leaf node has exactly J non-leaf nodes. According to [70], a small DT is much less prone to overfitting compared to a fully-grown DT; therefore, the small DTs used in the proposed scheme are built without pruning. Examples of small DTs are given in Fig. 3.2 with J=2. It can be seen that the non-leaf nodes of h_1 are exactly the same as the corresponding nodes of the DT in Fig. 3.1. It is worth noting that the optimal choice of J is highly dependent on the knowledge base, and should be decided based on a bias-variance analysis [68], which will be discussed in the case study. Note also that different from [68], the tree height, instead of



(a) Small DT h_1

(b) Small DT h_2



(c) Small DT h_3

Figure 3.2 The first three small DTs (J=2) for the WECC system, the voting weights of which are 4.38, 3.04 and 0.93, respectively

the number of nodes, is used as the metric to quantify the tree size. The reason, which will be soon apparent, is to restrict the number of nodes that will be revised when updating DTs to a value less than *J*.

3.2.2 Ensemble DT Learning

In ensemble-DT-based DSA schemes, the security classification decision of an OC vector $\mathbf{x}$, denoted by $H_L(\mathbf{x})$, is made based on the voting of multiple DTs. For an ensemble of DTs h_l $(l = 1, 2, \cdots, L)$, there are two approaches to DSA classification: deterministic and probabilistic. For the deterministic approach, the security classification decision is given by:

$$H_L(\mathbf{x}) = \begin{cases} +1, & \text{if } \sum_{l=1}^{L} a_l h_l(\mathbf{x}) \geq 0 \\ -1, & \text{o.w.} \end{cases} \tag{3.1}$$

where a_l $(l = 1, 2, \cdots, L)$ are the voting weights of DTs. To obtain probabilistic classification decisions, the "logistic correction" technique [71] can be applied. Then, the probability of an "Insecure" classification decision is given by:

$$\Pr(H_L(\mathbf{x}) = +1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-\sum_{l=1}^{L} a_l h_l(\mathbf{x}))} \tag{3.2}$$

In this report, deterministic classification decision is used to calculate the misclassification rate for case studies.

The existing methods for ensemble DT learning include bagging, random subspace method, boosting and random forest. A comparison of these state-of-the-art methods can be found in [72]. In previous work by the authors [66], an algorithm for boosting DTs is developed in the context of avoiding overfitting to noisy training data. In this project, the boosting algorithm is employed in online DSA to deal with OC variations and possible topology changes. The algorithm for building the small DTs and calculating the voting weights will be discussed in Section 3.3.1. Further, it is shown that random subspace methods can lead to improved accuracy and generalization capability, if the DTs are trained from a variety of compact and non-redundant attribute subsets. Usually, the attribute subsets used by DTs are selected in a randomized manner. For example, in the random forest algorithm, each DT is built by using an attribute subset that is randomly selected from all possible candidate attribute subsets with equal weights. For online DSA, it is observed that additional system information on the attributes could be utilized to create and select the attribute subsets. First, the candidate attribute subsets could be significantly refined by exploiting the locational information of attributes. Further, by

putting more weights on the attribute subsets that have higher availability when randomly selecting attribute subsets, the resulting small DTs would be more likely to be robust to possibly missing PMU measurements.

3.2.3 Updating DTs

One existing approach for updating a DT without rebuilding it from scratch is the efficient tree restructuring algorithm [73], with the main idea summarized as follows. When incorporating a new case, the DT remains unchanged if the new case is classified correctly; otherwise, the non-leaf nodes along the path which the new case passes are revised in a top-down manner. Specifically, for each non-leaf node to be revised, a new test is first identified by using the new case as well as the existing cases that fall into the non-leaf node. If different from the original test, the newly identified test is then installed at the non-leaf node, followed by tree restructuring operations recursively applied on the sub-tree corresponding to that non-leaf node (there are six slightly different restructuring operations for various structures of the sub-tree, which are not discussed here). The motivation for these restructuring operations is that the original test at the non-leaf node is highly likely to be the optimal tests for the two child nodes after restructuring, which is usually the case when categorical attributes are used by the test [73]; in this scenario, the two child nodes are exempted from further update.

3.3 Proposed Robust Online DSA for OC Variations and Topology Changes

In this project, a robust data-mining-based DSA scheme using adaptive ensemble DT learning is proposed to handle OC Variations and Topology Changes in an efficient manner. The proposed scheme for online DSA, as illustrated in Fig. 3.3, consists of three

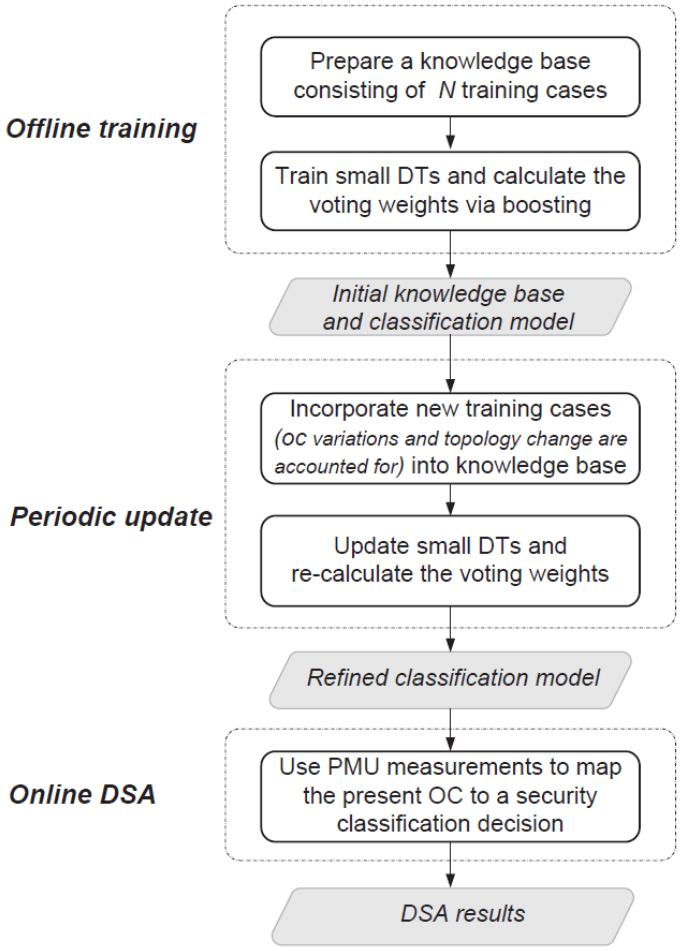


Figure 3.3 Proposed online DSA using adaptive ensemble DT learning

steps, with the details described below. Specifically, the classification model for DSA is based on boosting multiple unpruned small-height DTs. Generally, the height of a DT is the maximal number of tests that is needed for the DT to classify a case. For the sake of brevity, small-height DTs are referred to as small DTs throughout. In offline training, the small DTs and their voting weights are sequentially identified in a "gradient-descent" manner to minimize the misclassification cost. The small DTs, together with their voting weights, are then periodically updated throughout the operating horizon by using new training cases that are created to account for any change in OC or network topology.

Different from existing DT-based DSA schemes, the training cases are assigned different data weights by each small DT; and higher data weights are assigned to a new training case if it is misclassified by the small DTs. The aforementioned techniques are utilized to minimize the misclassification cost as new training cases are added to the knowledge base, so that the classification model could smoothly track the changes in OCs or system topology.

3.3.1 Offline Training

3.3.1.1 Initial Knowledge Base Preparation

First, N_{OC} predicted OCs are generated day ahead for each period of the future operating horizon (e.g., the next 24 hours) based on day-ahead load forecast and generation schedules; each period may span several hours, and can be divided according to the hours of peak load, shoulder load and off-peak load. Then, for each of the N_{OC} day-ahead predicted OCs, detailed power flow analysis and time-domain simulations are performed for K critical contingencies that are selected by the system operator or based on prior experience. It is worth noting that the key focus here is on dealing with OC variations and possible topology change, and thus the selection or screening of critical contingencies is beyond the scope of this project. By using specified dynamic security criteria (e.g., transient stability, damping performance, transient voltage drop/rise, transient frequency, relay margin), the day-ahead predicted OCs are labeled as "Secure" or "Insecure" for each critical contingency.

As a result, an initial knowledge base that consists of $N = N_{OC} \times K$ training cases is obtained, in which each case is represented by a vector $\{x_1, \cdots, x_P, y\}$, where x_1 is the

index of a critical contingency, $\{x_2, \cdots, x_P\}$ are the values of numerical attributes obtained from power flow analysis of an OC, and y is the transient security classification decision of the OC for the critical contingency x_1. Based on the previous studies [16][18][20], the following PMU-measured variables are selected as numerical attributes:

- Branch active power flows $\{P_{ij}; i \in \mathrm{B} \ or \ j \in \mathrm{B}\}$
- Branch reactive power flows $\{Q_{ij}; i \in \mathrm{B} \ or \ j \in \mathrm{B}\}$
- Branch current flows (magnitude) $\{I_{ij}; i \in \mathrm{B} \ or \ j \in \mathrm{B}\}$
- Bus voltage magnitudes $\{V_i; i \in \mathrm{B}\}$
- Bus voltage phase angle differences $\{A_{ij} \triangleq A_i - A_j; i, j \in \mathrm{B} \ and \ i > j\}$

where B denotes the set of PMU buses in the system. It is worth noting that only raw measurements reported by PMUs are used as the numerical attributes in this work; more generally, the variables computed using other system information may also be used, e.g., the voltage at the bus connected to a PMU bus when the branch impedance is constant [16].

3.3.1.2 Boosting Small DTs

The basic algorithmic flowchart of boosting small DTs is illustrated in Fig. 3.4. For convenience, define H_J as the class of small DTs with height J, define F_L as the score of the weighted voting of the ensemble of small DTs, i.e., $F_L(\mathbf{x}) = \sum_{l=1}^{L} a_l h_l(\mathbf{x})$, and define $C_N(F_L)$ as the cost function of F_L on the N training cases, given by:

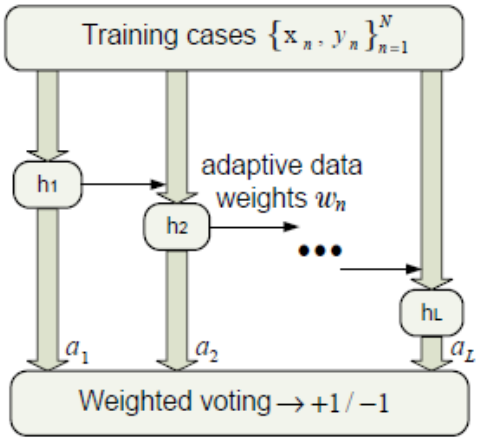


Figure 3.4 Boosting small DTs

$$C_N(F_L) = \frac{1}{N}\sum_{n=1}^{N} \log_2\left(1+\mathrm{e}^{-y_n F_L(\mathbf{x_n})}\right). \tag{3.3}$$

It is observed from (3.1) and (3.3) that $C_N\left(F_L\right)$ lies strictly above the misclassification error rate of H_L. Then, a primary objective of boosting is to minimize $C_N\left(F_L\right)$, by identifying the small DTs $h_l \in \mathrm{H}_J$ and their voting weights $a_l \in \mathrm{R}^+$. An analytical formulation is provided as follows:

$$\mathrm{P}_F : \min_{\substack{h_1,\cdots,h_L \in \mathrm{H}_J \\ a_1,\cdots,a_L \in \mathrm{R}^+}} C_N(F_L). \tag{3.4}$$

The convexity and the differentiability of $C_N\left(F_L\right)$ with regard to F_L make it possible to solve P_F in (3.4) by using a line search strategy [74], the details of which are summarized as follows. A small DT h_l is chosen to be the "gradient" of $C_N\left(.\right)$ at F_{l-1} projected onto H_J, and the voting weight a_l is computed as the "step size" that minimizes $C_N(F_{l-1}+a_l h_l)$. Then, the small DT h_l is added to F_{l-1} to obtain $F_l = F_{l-1}+a_l h_l$. The

above steps are iterated, for $l = 1, 2, \cdots, L$, by using F_0 as a zero function. More specifically, it is shown in [66] that the small DT h_l can be obtained by solving the following problem:

$$\mathrm{P}_h^{(l)} : \min_{h_l \in \mathrm{H}_J} \frac{1}{N} \sum_{n=1}^{N} w_n^{(l)} \mathbf{1}_{\{y_n \neq h_l(\mathbf{x}_n)\}} \tag{3.5}$$

where $w_n^{(l)} \underline{\underline{\Delta}} (1 + \mathrm{e}^{y_n F_{l-1}(\mathbf{x}_n)})^{-1}$ is the positive data weight of the training case $\{\mathbf{x}_n, y_n\}$, and $\mathbf{1}_{\{y_n \neq h_l(\mathbf{x}_n)\}}$ takes value 0 if the training case $\{\mathbf{x}_n, y_n\}$ is correctly classified by the small DT h_l (otherwise, it takes value 1). By definition of $w_n^{(l)}$, it is easy to observe that the data weights are assigned adaptively by small DTs, in the sense that if the training case $\{\mathbf{x}_n, y_n\}$ is misclassified by the small DT h_l, then $w_n^{(l+1)} > w_n^{(l)}$, i.e., the training case has a higher data weight in the next round of the boosting process. It is worth noting that highly skewed training data (e.g., the case in [19]) can be handled by scaling up the weights of under-represented cases, such that $\sum_{y=+1} w_n^{(l)} = \sum_{y=-1} w_n^{(l)}$. As suggested in (3.5), the objective of $\mathrm{P}_h^{(l)}$ is to determine the small DT that has the least misclassification error rate on the weighted training data. Thus, the small DT h_l can be obtained by employing the standard CART algorithm [11] subject to the tree height J, and by using misclassification error rate as the splitting cost when building the DT. Then, its positive voting weight is obtained by solving the following problem:

$$\mathrm{P}_a^{(l)} : \min_{a \in \mathrm{R}^+} G_N^{(l)}(a) \tag{3.6}$$

where $G_N^{(l)}(a) \underline{\underline{\Delta}} C_N(F_{l-1} + ah_l)$. Under the condition that h_l is a "descent direction" of $C_N(F_{l-1})$, it is easy to verify that $G_N^{(l)'}(0) < 0$ and $G_N^{(l)''}(a) > 0$ holds for any $a \in \mathrm{R}^+$. Therefore, $G_N^{(l)}(a)$ has a unique minimum in R^+ that can be found using standard numerical solution methods (e.g., Newton's method).

3.3.2 Periodic Updates

3.3.2.1 New Training Case Creation

In the initial knowledge base prepared offline, the predicted OCs generated using day-ahead forecast may not reflect the actual system conditions, which is very likely to be the case for power systems with high penetration of variable renewable generation and distributed generation. Therefore, as the operating horizon is approached and the data available to system operators is updated, it will be necessary to utilize short-term forecast and schedules to generate newly changed OCs and add them to the knowledge base on a slot-by-slot basis (one slot may span several minutes depending on the processing speed [16]). Further, in case of topology change, the post-disturbance OCs should also be incorporated into the knowledge base. After power flow analysis of these newly changed OCs, new training cases are generated as described in Section 3.3.2.1. It is worth noting that during the operating horizon, it is also likely that the knowledge base may need to be updated by incorporating new contingencies of interest. The solution to this problem has been discussed in [66]. In this work, the critical contingency list is assumed to remain unchanged during the operating horizon.

3.3.2.2 Updating the Classification Model

Given the newly created training cases, the classification model is updated by using one new case at a time. Specifically, for the *k*-th new training case $\{\mathbf{x}_{N+k}, y_{N+k}\}$, the classification model is updated by incorporating $\{\mathbf{x}_{N+k}, y_{N+k}\}$ with a data weight $w_{N+k}^{(l)} = (1+\mathrm{e}^{y_{N+k}F_{l-1}(\mathbf{x}_{N+k})})^{-1}$ into the small DT h_l and recalculating the voting weight a_l, iteratively for $l = 1, 2, \cdots, L$.

A key step for incorporating a new training case into a small DT is to adopt the method described in Section 3.2.3. Since misclassification error rate is used as the metric of splitting cost, as suggested in (3.5), it is easy to observe that there exists a even simpler solution for updating the small DTs. Specifically, a small DT remains unchanged if the new case is correctly classified; otherwise, only the sub-tree corresponding to the first non-leaf node that has a different decision for the new case is subject to update. It is worth noting that, since the tree height is J, the total number of non-leaf nodes to be revised is at most J. After the small DT h_l is updated, its voting weight al is recalculated by minimizing $G_{N+k}^{(l)}(a)$.

The process of updating the classification model is summarized in Algorithm 3.1. It is useful to note that when the *k*-th new training case is used to update the small DTs, the data weights of the previous $N+k-1$ training cases calculated in Step 4 of Algorithm 3.1 are different from the data weights that were used in building or updating the small DTs in the past rounds. Therefore, unlike the case in offline training, it is possible that the updated small DT h_l is not a "descent direction" of C_{N+k} at F_{l-1} any more. In order to

detect and handle this situation, an extra step is used in Algorithm 3.1. Specifically, if $\sum_{n=1}^{N+k} w_n^{(l)} y_n h_l(\mathbf{x}_n) < 0$, then h_l is a "descent direction" and used for weighted voting.

Algorithm 3.1: Periodic updates using a new training case

Input: A new training case $\{\mathbf{x}_{N+k}, y_{N+k}\}$.

Initialization: $F_0 = \mathbf{0}$

For $l = 1$ **to** L **do**

Recalculate the data weights of $\{\mathbf{x}_n, y_n\}_{n=1}^{N+k-1}$.

Incorporate $\{\mathbf{x}_{N+k}, y_{N+k}\}$ with weight $w_{N+k}^{(l)}$ into h_l.

Calculate $\varepsilon = \sum_{n=1}^{N+k} w_n^{(l)} y_n h_l(\mathbf{x}_n)$.

If $\varepsilon < 0$, **then**

$h_l \leftarrow -h_l$

End if

Recalculate a_l by minimizing $G_{N+k}^{(l)}(a)$.

$F_l \leftarrow F_{l-1} + a_l h_l$

End For

3.3.3 Online DSA using PMU Measurements

In real-time, when the synchronized PMU measurements are received, the pre-fault values of the numerical attributes are retrieved and combined with the indices of all critical contingencies to create K unlabeled cases, which will be used by the classification model to give security classification decisions of the present OC for the K critical contingencies. Specifically, when an unlabeled case is processed by the

classification model, each of the small DTs uses the values of the attribute vector and its CSRs to produce a binary decision. Finally, the binary decisions of all small DTs are collected and used to give the security classification decisions of the present OC, according to (3.1). It is worth noting that distributed processing technologies [75] can be leveraged to speed up online DSA. Specifically, the K unlabeled cases can be classified separately by using K duplicates of the classification model, and in each classification model, all small DTs can process the attribute vector of an unlabeled case in a parallel manner.

From the above development, it can be seen that the proposed scheme illustrated in Fig. 3.3 is derived from those in previous work [16][18][20], with the following major modifications. 1) The classification model is obtained via boosting multiple small unpruned DTs instead of a single fully-grown DT after pruning. It is suggested that boosting algorithms can lead to better model fitting and the produced classification model is quite resistant to overfitting [70]. Thus, boosting small DTs has great potential to deliver better performance in terms of classification accuracy. 2) Unequal data weights are assigned to the training cases adaptively by small DTs. In periodic updates, misclassified new training cases can have higher data weights than those classified correctly. This will speed up adapting the small DTs to newly changed OCs. 3) The small DTs are gracefully updated by incorporating new cases one at a time, whereas rebuilding DTs is used in [16][18][20]. 4) The DT and the knowledge base are updated only when the new cases are misclassified in [16][18][20]; whereas all new training cases are incorporated into the knowledge base in the proposed scheme.

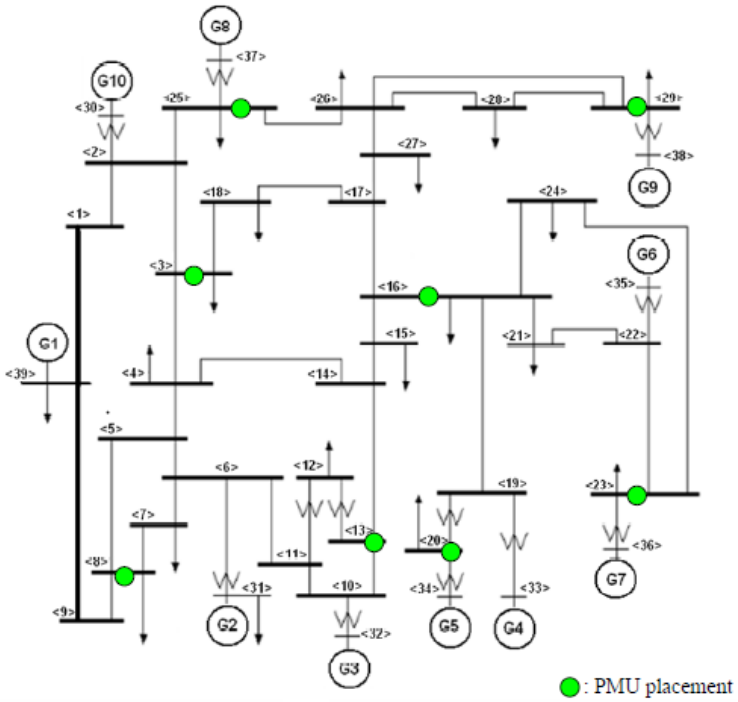


Figure 3.5 The IEEE 39-bus system with 8 PMUs

3.3.4 An Illustrative Example

The IEEE 39-bus test system [35] is used as an illustrative small system. As illustrated in Fig. 3.5, 8 PMUs are installed in the system, according to the placement design provided in [76]. In what follows, the main steps of the proposed approach, including attribute selection, knowledge base preparation and ensemble small DT learning, will be demonstrated by using the IEEE 39-bus test system. Finally, the results of robustness test on changed OCs will be presented.

3.3.4.1 Knowledge Base

3.3.4.1.1 Attribute Selection

Based on the PMU placement and system topology in Fig. 3.5, 111 numerical attributes are selected according to the rules described in Section 3.3.1.1, including:

- 8 bus voltage magnitudes at the 8 PMU buses;
- 75 branch active/reactive power flows and current flows, which take any of the 8 PMU buses as either a from-bus or a to-bus of the branch;
- 28 bus voltage phase angle differences, which are computed from the 8(8-1)/2 pairs of phase angles.

3.3.4.1.2 OC Generation and Contingencies

The OC specified in [35] is used as the base OC. To enrich the knowledge base, more OCs are generated by randomly changing the bus loads (both active and reactive) within 90% to 110% of their original values in the base OC. For each generated OC, limit checking is carried out by using the power flow and short circuit analysis tool (PSAT) [77], so that any generated OC with pre-contingency overloading or violation of voltage magnitude/angle limits is not included in the knowledge base. Further, transient stability assessment is carried out for the 30 *N-2* contingencies listed in [78, Table II]. These *N-2* contingencies, which can lead to stressed system conditions, are identified by exhaustive search among all possible *N-2* contingencies.

3.3.4.1.3 Transient Stability Assessment Tool and Criteria

The transient security assessment tool (TSAT) [77] is used to assess the transient performance of the generated OCs. The time-domain simulation is executed for 10

seconds with a step size of 0.5 cycle. The power angle-based stability margin is used as the transient stability index (TSI), defined as

$$\eta = \frac{360 - \delta_{max}}{360 + \delta_{max}} \times 100, \quad -100 < \eta < 100 \tag{3.7}$$

where δ_{max} is the maximum angle separation of any two generators in the system at the same time in the post-fault response. In case of islanding, the above value is evaluated for each island and the smallest value is taken as the TSI. During the simulation time, whenever the margin η turns out to be negative, i.e., the rotor angle difference of any two generators exceeds 360 degree, the case is labeled as transiently insecure.

3.3.4.2 Offline Training

3.3.4.2.1 Choice of *J* and *L*

V -fold cross validation (V =10) is carried out to determine the optimal tree height J and the optimal number of small DTs L. Specifically, the training cases in the initial knowledge base are randomly partitioned into V subsets of equal size. For given fixed J and L, a classification model is trained by using $V-1$ subsets, and tested using the other subset. The training process is then repeated V times in total, with each of the V subsets used exactly once as the test data. Finally, the misclassification error rate obtained by V -fold cross validation is calculated by averaging over the V classification models. The results of the above procedure for different tree heights (J =1, 2, 3) are illustrated in Fig. 3.6. It can be seen that as L increases, the misclassification error rate of each classification model decreases and reaches a plateau at some L. Then, when L grows larger, each classification model incurs a larger variance and hence a higher

misclassification error rate. On the other hand, a larger tree height J implies a larger variance of classification model [68], which is also observed in Fig. 3.6. Based on these observations, J =2 is chosen, and L =15 at which the misclassification error rate drops below 1% and reaches a plateau is selected.

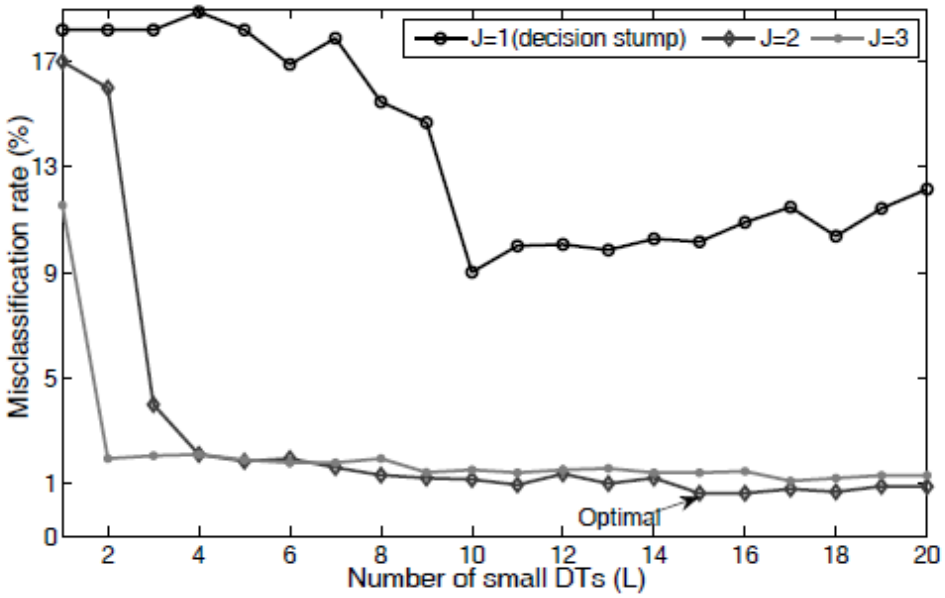


Figure 3.6 Ensemble small DT learning with different tree heights for the IEEE 39-bus test system

3.3.4.2.2 Ensemble Small DT Learning

When the optimal tree height J and the optimal number of small DTs L are determined, the algorithm described in Section 3.3.1.2 is used to build the ensemble of small DTs. Specifically, for $l = 1, 2, \cdots, L$, the data weights $w_n^{(l)}$ are first computed according to (3.5). Then, the training cases together with their data weights are used by the CART algorithm to build a small DT h_l with height J , by using weighted misclassification rate as the cost function, as shown in (3.5). Note that each small DT gives security classification decisions for all critical contingencies. Further, the voting

weight of h_l is calculated by numerically solving (3.6). Then, the ensemble of small DTs is obtained. It is worth noting that, different from the V -fold cross validation procedure, the entire training set (not a subset) is used by each small DT of the ensemble.

3.3.4.3 Robustness Testing

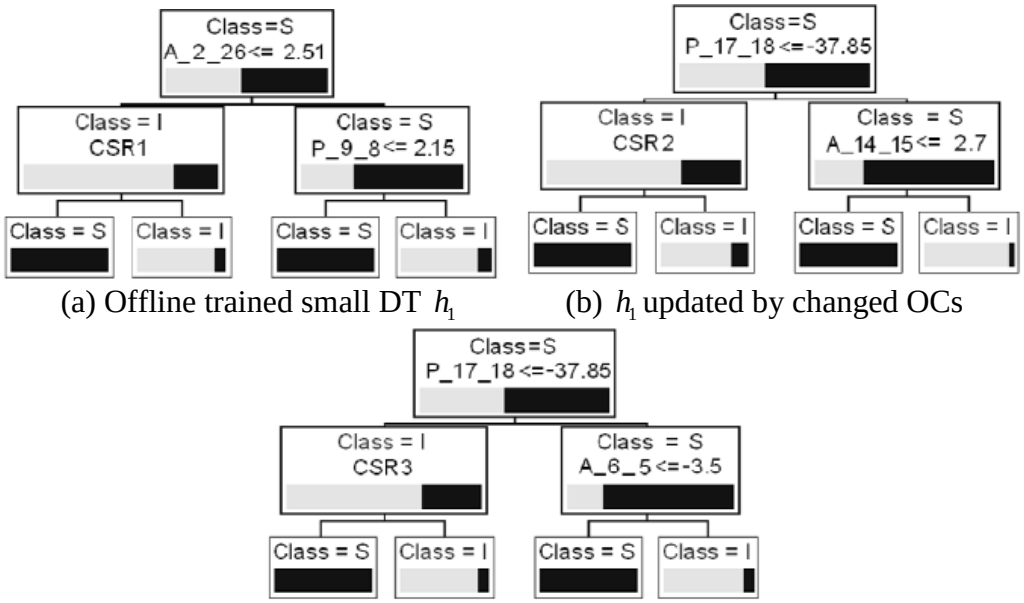
3.3.4.3.1 Changed OCs

In the IEEE 39-bus test system, generator G1, together with transmission lines (39, 9) and (39, 1), represents the equivalent to the external system of the New England area [35]. It is now assumed that the capacity of G1 reduces from 1100 MW to 900 MW, which could be the result of either the loss of a transmission corridor or a generator tripping outside the New England area. Therefore, the OCs will change due to generation rescheduling. By setting the capacity of G1 to 900 MW, changed OCs are generated by rescheduling generation and re-solving power flows for each OC in the initial knowledge base. These changed OCs will be utilized to test the robustness of the proposed approach.

3.3.4.3.2 Robustness Testing Results

First, 200 OCs are generated to create the initial knowledge base consisting of 6000 (200 OCs × 30 contingencies) training cases. Accordingly, another 200 changed OCs are generated, in which 100 OCs are used to update the small DTs and the other 100 OCs are used for robustness testing. In the proposed approach, Algorithm 3.1 is applied to update each of the 15 small DTs by using the 3000 (100 OCs × 30 contingencies) new cases. To illustrate the change of small DTs, the first small DT h_1 is used as an example. Specifically, h_1 obtained in offline training and updated with the 100 changed OCs by using the proposed approach are illustrated in Fig. 3.7(a) and Fig. 3.7(b), respectively. It is observed that due to the changed OCs and generation rescheduling, the critical attribute

in the root node of h_1 changes from the voltage phase angle difference between bus 2 and bus 26, *A_2_26*, to the active power flow between bus 17 and bus 18, *P_17_18*. The CSRs of the non-root nodes change accordingly, as a result of the recursive procedure of the CART algorithm. The small DT h_1 rebuilt with the 100 changed OCs is illustrated in



(a) Offline trained small DT h_1 (b) h_1 updated by changed OCs

(c) Small DT h_1 rebuilt with changed OCs

Figure 3.7 The first small DT h_1 (J =2) for the IEEE 39-bus test system

Fig. 3.7(c), which has the same CSR at the root node as the small DT updated by using the proposed approach. Since the small DTs h_1 obtained by updating and rebuilding are different at non-root nodes, the other small DTs, h_2 to h_{15} are also different. This is because the ensemble DT learning algorithm sequentially updates/builds the small DTs, in which each small DT depends on the previous small DTs.

The proposed approach is compared with two benchmark approaches: 1) small DTs rebuilt by using the 100 changed OCs together with the initial 200 OCs, 2) small DTs without updating. The test results of the three approaches are presented in Table 3.1. It

can be seen that the proposed approach achieves comparable performance to the benchmark approach by rebuilding small DTs. The test results also suggest that when OCs change, the small DTs have to be updated in order to track the variation of OCs.

Table 3.1 Misclassification error rate of robustness testing

	Secure cases	Insecure cases	Overall
Proposed	0.68%	0.36%	0.55%
Small DTs (rebuilding)	0.59%	0.38%	0.54%
Small DTs (no updating)	10.68%	6.85%	9.57%

3.3.5 Application to the WECC System

The test power system used in this case study is part of the WECC system. It consists of over 600 buses (of which 33 are PMU buses), 700 transmission lines and 100 generators.

3.3.5.1 Knowledge Base

3.3.5.1.1 OC Generation

The OCs used in the case study are generated by using real-life data of power flows, bus loads and generator power outputs that were recorded every 15 minutes during a 2008 summer peak day. The overall load profile is illustrated in Fig. 3.8. Based on the variations of the aggregate load, each period for offline training is chosen to span 8 hours, and the peak load period 1200 Hrs-2000 Hrs is investigated in this case study. Basically, there are three sets of generated OCs used in this case study: day-ahead predicted OCs, short-term predicted OCs and realized OCs. The day-ahead predicted OCs are used to create the initial knowledge base, the short-term predicted OCs are used to create the new training cases to update the knowledge base and the classification model, and the realized OCs are used for testing purposes only.

In what follows, the procedure for generating the three OC sets is discussed in detail. The realized OCs include the 33 recorded OCs and another 448 OCs that are generated by interpolation, as illustrated in Fig. 3.8. Specifically, following the method in

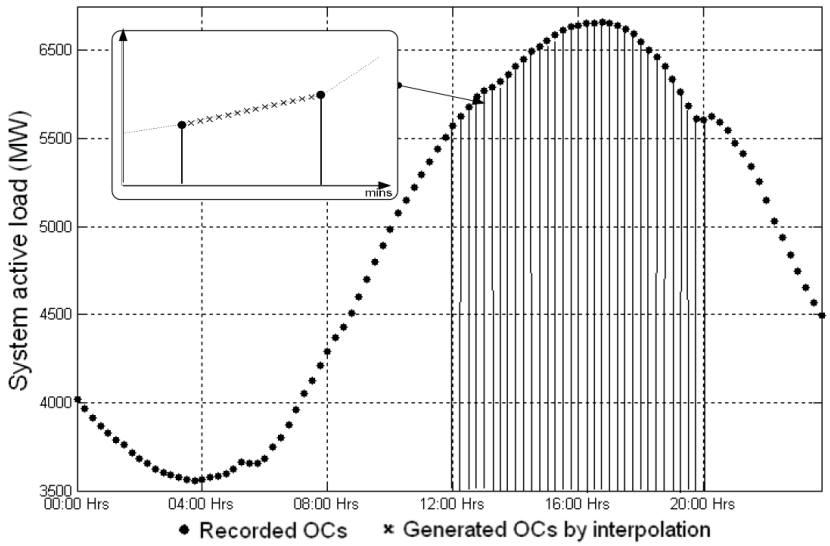


Figure 3.8 Aggregate load of recorded OCs and generated OCs by interpolation

[20], both the active and reactive load of each load bus for every minute of the investigated period are obtained by linear interpolation based on the two closest recorded OCs, and the generator power outputs are adjusted as needed to ensure valid OCs. To enrich the initial knowledge base, a day-ahead predicted OC is obtained by randomly changing the bus loads within 90% to 110% of the loads of the corresponding realized OC, by using a uniform distribution. Similarly, a short-term predicted OC is generated by uniformly randomly changing the bus loads within 97% to 103% of the loads of the corresponding realized OC. After solving the power flows for each OC using the power flow and short circuit analysis tool (PSAT) [77], 481 OCs are generated for each of the three OC sets. It is worth noting that different from the day-ahead predicted OCs, the short-term predicted OCs and the realized OCs are time-stamped.

3.3.5.1.2 Critical Contingency Selection

A contingency list, which was created by the regional grid operator to account for possible outages of transmission lines, three-winding transformers and generators that could have significant impact, is used here. Specifically, the contingency list consists of 1 *N*-4 contingency, 8 *N*-3 contingencies, 172 *N*-2 contingencies, and 0 *N*-1 contingencies (i.e., no *N*-1 contingencies lead to insecure conditions). The power angle-based stability margin defined in (3.7) is used as the transient stability index. After performing transient security assessment by using TSAT for all realized OCs and adhering to the above security criteria, three *N*-2 contingencies which lead to transiently insecure cases are selected as the critical contingencies in the knowledge base. Each of the three *N*-2 critical contingencies is initiated by a "three-phase short circuit to ground" fault at a bus which is cleared after 5 cycles, by tripping a transmission line that connects the bus and by disconnecting a generator that will go out of step as a result of the line tripping.

3.3.5.1.3 Case Creation

Combining the three sets of generated OCs with their transient security classification decisions for the three critical contingencies, N=1443 cases are created for the initial knowledge base, for updating and for testing, respectively. Based on the interconnection structure of the 33 PMU buses, 799 numerical attributes are identified using the rules described in Section 3.3.1.1; thus P=800. For each case, the values of the 799 numerical attributes are obtained from the power flow solutions. Then, the initial knowledge base is organized into an $N\times(P+1)$ array.

3.3.5.2 Offline Training

The initial knowledge base as an $N\times(P+1)$ array is first used by the CART algorithm to build the small DTs. Following the procedure described in Section 3.3.4.2, it is found that J =2 and L =35 give the best results of V -fold cross validation. The first three small DTs built from the initial knowledge base are illustrated in Fig. 3.2. For comparison, a fully-grown single DT with pruning is also built, as illustrated in Fig. 3.1.

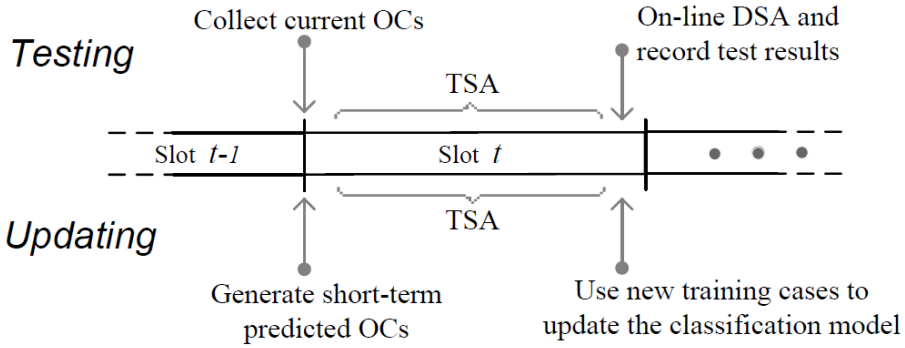


Figure 3.9 Flowchart for testing online DSA with periodic updates

3.3.5.3 Online DSA Simulation

The online DSA is simulated iteratively on a slot-by-slot basis, as illustrated in Fig. 3.9. Generally, each slot spans M minutes. Since it is sufficient to perform security assessment of a short-term predicted OC for the three *N-2* critical contingencies, M =1 is chosen here. In case of more critical contingencies or a larger test system, a longer slot can be chosen. In online DSA, a third scheme in which the classification model is obtained by boosting small DTs but updated by rebuilding is compared with the two aforementioned schemes.

3.3.5.3.1 OC variations in sub-period 1200 Hrs-1600 Hrs

In each slot of this sub-period, the $3M$ test cases created from the M realized OCs with time-stamps falling into this slot are collected, and then used as the present OCs for online DSA to assess the performance of the classification model updated so far. Meanwhile, another $3M$ new training cases created from the short-term predicted OC for the next slot are incorporated into the knowledge base to update the classification model.

3.3.5.3.2 Topology change in sub-period 1600 Hrs-2000 Hrs

At the peak hour 1600 Hrs, a topology change is imposed on the test system, and assumed to last for the remaining hours of the day. Specifically, among the 178 contingencies that do not incur transient instability for all realized OCs, the contingency which has the least positive margin averaged over all realized OCs is chosen; as a result, a transmission line is removed and a generator is disconnected from the test system. Then, the new training cases and test cases during the latter sub-period are created using an approach similar to those used in the former sub-period, but by using a different system topology.

3.3.5.4 Test Results and Discussion

Throughout the entire horizon of the above online DSA simulations, the misclassification error rate and the computation time for updating in each slot are recorded and summarized in Table 3.2 and Fig. 3.10, respectively.

3.3.5.4.1 Classification Accuracy

As illustrated in Table 3.2, the two boosting-based schemes turn out to be more accurate than the single-DT-based scheme for both simulation sub-periods, and the

performance of the proposed scheme is quite close to the scheme based on boosting small DTs with rebuilding.

Table 3.2 Misclassification error rate of online DSA

Scheme	Sub-period 1200-1600 Hrs			Sub-period 1600-2000 Hrs		
	Secure cases	Insecure cases	Overall	Secure cases	Insecure cases	Overall
Proposed	2.41%	1.03%	1.67%	2.54%	1.08%	1.74%
A single DT (rebuilding)	2.71%	1.80%	2.22%	2.26%	2.73%	2.5%
Boosting (rebuilding)	1.81%	1.03%	1.39%	2.26%	0.82%	1.5%

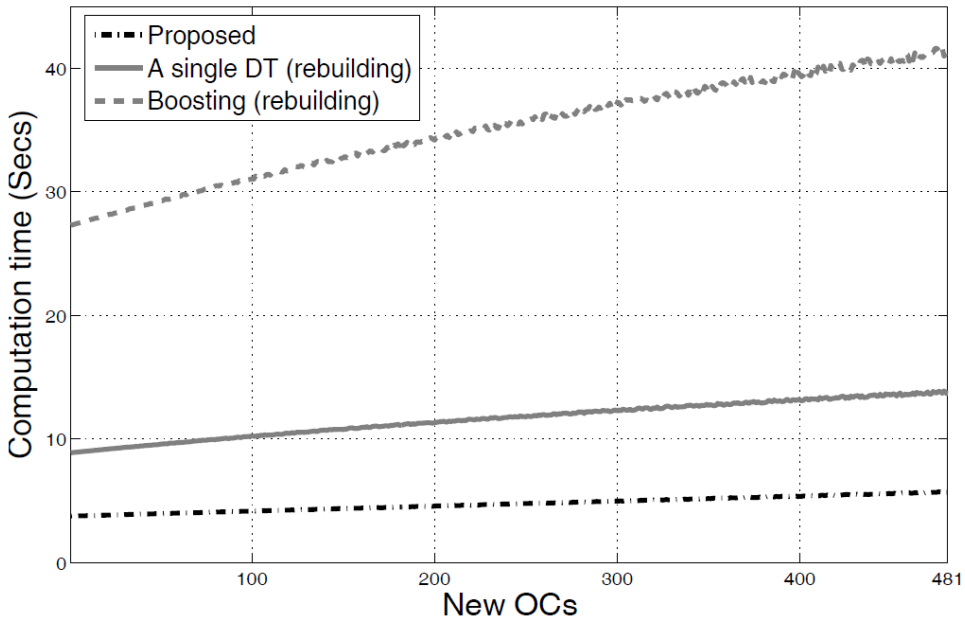


Figure 3.10 Computation time for updating/rebuilding (executed in MATLAB on a workstation with an Intel Pentium IV 3.20 GHz CPU and 4GB RAM)

3.3.5.4.2 Computation Requirement

The computation time required by updating the classification models using new OCs is illustrated in Fig. 3.10. It is clear that the proposed scheme requires the lowest computation time. Further, as the number of new OCs increases, the proposed scheme

becomes less time-consuming than the other two schemes. The reason is that for each new OC, the two benchmark schemes rebuild DTs from scratch, while the graceful update of small DTs is carried out in the proposed scheme. Further, according to the CART algorithm [11], it is known that the sorting operation of the CART algorithm dominates the computational burden of DT building/rebuilding. When updating small DTs, the sorting operation is skipped [73]. Therefore, the proposed scheme has a much lower computational burden.

3.4 Proposed Robust Online DSA for Missing PMU Measurements

Previous studies on PMU measurement-based online DSA implicitly assume that wide area monitoring systems (WAMS) provide reliable measurements. However, in online DSA, PMU measurements can become unavailable due to the unexpected failure of the PMUs or phasor data concentrators (PDCs), or due to loss of the communication links. Recently, it has been widely recognized that PMU failure can be an important factor that impacts the performance of WAMS. For example, AESO's newest rules on implementing PMUs [79] require that the loss or malfunction of PMUs, together with the cause and the expected repair time, has to be reported to the system operator in a timely manner. In the report [80], the deployment of redundancy is suggested by PMU manufacturers to reduce the impact of single PMU failure. Loss of PMUs has also been taken into account when designing WAMS and PMU placement [81]. Moreover, the delivery of PMU measurements from multiple remote locations of power grids to monitoring centers could experience high latency when communication networks are heavily congested, which could also result in the unavailability of PMU measurements.

Therefore, it is urgent to design DT-based online DSA approaches that are robust to missing PMU measurements.

Intuitively, one possible approach to handle missing PMU measurements is to estimate the missing values by using other PMU measurements and the system model. However, with existing nonlinear state estimators in supervisory control and data acquisition (SCADA) systems, this approach may compromise the performance of DTs. First, the scan rate of SCADA systems is far from commensurate with the data rate of PMU measurements, and thus using estimated values from SCADA data may result in a large delay for decision making. Second, SCADA systems collect data from remote terminal units (RTUs) utilizing a polling approach. Following a disturbance, it is possible that some post-contingency values are used due to the lack of synchronization, which can lead to inaccurate security classification decisions of DTs. It is worth noting that future fully PMU-based linear state estimators [82] can overcome the aforementioned limitations; but this is possible only when there is a sufficient number of PMUs placed in system. With this motivation, data-mining based approaches are investigated in this paper, aiming to use alternative viable measurements for decision making in case of missing data.

In DTs built by the classification and regression tree (CART) algorithm [11], missing data can be handled by using surrogate. However, a critical observation in this project is that when PMU measurements are used as attributes, most viable surrogate attributes have low associations with the primary attributes. Clearly, the accuracy of DSA would degrade if surrogate is used. This is because a DT is essentially a sequential processing method, and thus the wrong decisions made in earlier stages may have

significant impact on the correctness of the final decisions. Thus motivated, this paper studies applying ensemble DT learning techniques (including random subspace methods and boosting), so as to improve the robustness to missing PMU measurements.

Aiming to develop a robust and accurate online DSA scheme, the proposed approach consists of three processing stages, as illustrated in Fig. 3.11. Specifically, given a collection of training cases, multiple small DTs are trained offline by using randomly selected attribute subsets. In near real-time, new cases are used to re-check the performance of small DTs. The re-check results are then utilized by a boosting algorithm

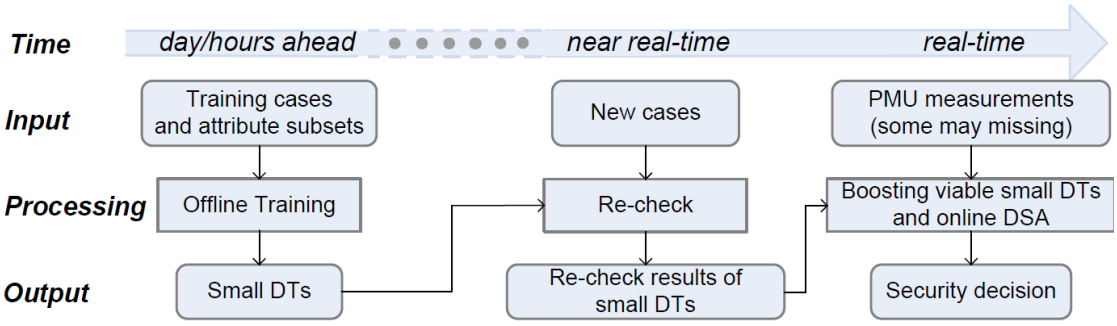


Figure 3.11 A three-stage ensemble DT-based approach to online DSA with missing PMU measurements

to quantify the voting weights of a few viable small DTs (i.e., the DTs without missing data from their attribute subsets). Finally, security classification decisions of online DSA are obtained via a weighted voting of viable small DTs. More specifically, a random subspace method for selecting attribute subsets is developed by exploiting the locational information of attributes and the availability of PMU measurements. Conventionally, the availability of a WAMS is defined as the probability that the system is operating normally at a specified time instant [83]. In this project, the availability of PMU measurements is defined similarly, i.e., as the probability that PMU measurements are

successfully collected and delivered to a monitoring center. The developed random subspace method guarantees that a significant portion of small DTs are viable for online DSA with high likelihood. Further, a boosting algorithm is employed to assign the viable small DTs with proper voting weights that are quantified by using the results from performance re-check, leading to the high robustness and accuracy of the proposed approach in case of missing PMU measurements. The proposed approach is applied to the IEEE 39-bus system with 9 PMUs. Compared to off-the-shelf DT-based techniques (including random forests (RFs) with and without using surrogate), the proposed ensemble DT-based approach can achieve better performance in case of missing PMU measurements.

3.4.1 Handling Missing Data by using Surrogate in DTs

A surrogate split at an internal node is the one that "mimics" the primary split most closely, i.e., gives the most similar splitting results for the training cases. Usually, the similarity is quantified by the association between the surrogate split and the primary split [11]. The significance of a surrogate split that has a high association (i.e., over 0.9) with the primary split is that the DT could still use the surrogate split at this internal node to give almost the same decisions when the PMU measurement of the primary attribute is missing.

The performance of surrogate in DT-based DSA is evaluated via a case study, in which a single DT is built by using the same knowledge base for voltage magnitude violation analysis as in [66]. It is observed that co-located attributes (i.e., the attributes measured by the same PMU) would often be unavailable at the same time when the PMU fails, which implies that co-located attributes cannot be used as surrogate for each other

in online DSA. Therefore, a modified CART algorithm in which co-located attributes are excluded from surrogate searching is used to build a single DT and identify the surrogate attributes. The results regarding the performance of the surrogates identified by both the modified CART algorithm and the CART algorithm are given in Table 3.3.

Two key observations are drawn. First, the results obtained by the modified CART algorithm suggest that all non-colocated surrogates have relatively low associations with the primary ones. The low association could be explained by the complex coupling

Table 3.3 Surrogates of the DT for the WECC system

Node	Primary Attribute	By modified CART		By CART	
		Surrogate	Association	Surrogate	Association
1	$V_{\{217\}}$	$V_{\{207\}}$	0.76	$V_{\{207\}}$	0.76
2	$Q_{\{204;207\}}$	$Q_{\{212;216\}}$	0.33	$Q_{\{207;209\}}$	0.50
3	$Q_{\{204;207\}}$	$V_{\{209\}}$	0.28	$Q_{\{207;209\}}$	0.64
4	$I_{\{211;204\}}$	$P_{\{008;011\}}$	0.62	$P_{\{209;211\}}$	0.83
5	$P_{\{210;201\}}$	$P_{\{211;062\}}$	0.87	$P_{\{231;201\}}$	0.87
6	$Q_{\{005;033\}}$	$Q_{\{801;999\}}$	0.71	$Q_{\{801;999\}}$	0.71
7	$P_{\{213;222\}}$	$Q_{\{207;211\}}$	0.85	$P_{\{222;223\}}$	0.85
8	$Q_{\{041;060\}}$	$I_{\{011;051\}}$	0.50	$I_{\{011;051\}}$	0.50
9	$P_{\{211;062\}}$	$P_{\{213;216\}}$	0.50	$I_{\{062;211\}}$	0.75
10	$P_{\{236;219\}}$	$Q_{\{230;052\}}$	0.42	$P_{\{236;207\}}$	0.68

structure of the attributes in power systems. According to the definition of surrogate, high association relies on the dependency between the surrogate and the primary attributes, i.e., the surrogate attribute gives similar decisions to the primary attribute on all the training cases regardless of any other attribute. However, in power systems, one attribute (i.e., voltage magnitude, voltage phase angle or power/current flow) is coupled with many other non-co-located attributes, as dictated by the AC power flow equations and the

network interconnection structure. Second, it is observed in Table 3.3 that the surrogate attributes found by the CART algorithm are mostly co-located with the primary attributes. This observation signifies the redundancy between co-located attributes when used for splitting the training cases, and thus sheds lights on exploiting the locational information to create the attribute subsets, as described in Section 3.4.2.

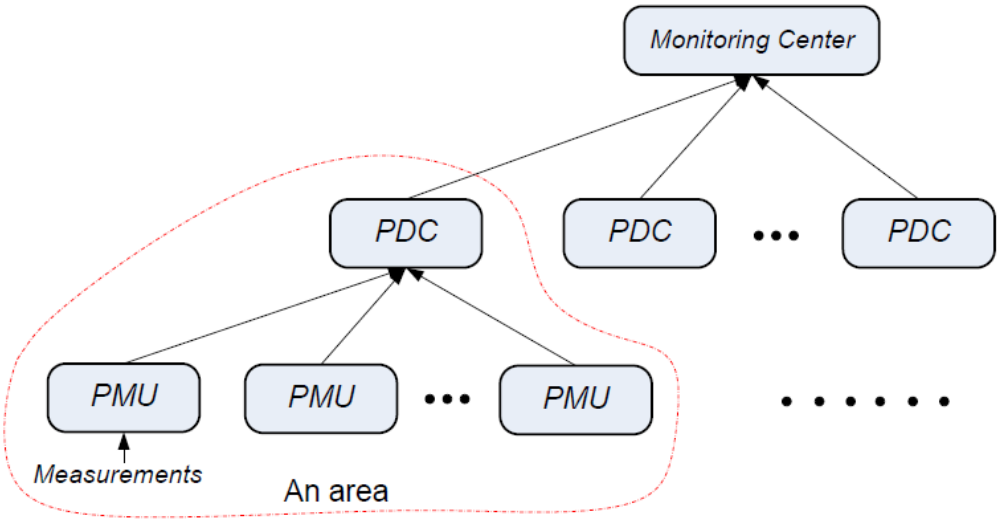


Figure 3.12 Wide area monitoring system consisting of multiple areas

3.4.2 Proposed Random Subspace Method for Selecting Attribute Subsets

A key step of the random subspace method is to identify a collection of candidate attribute subsets S and determine the weight p_s that dictates how likely a candidate attribute subset $\mathbf{s} \in S$ is to be selected. In this project, by exploiting the locational information of attributes and the availability of PMU measurements, the random subspace method adheres to the following two guidelines:

- **G1**: Co-located attributes do not co-exist within an attribute subset.

- **G2**: The average availability of the selected attribute subsets should be sufficiently high.

Further, for a power system consisting of K areas, the corresponding WAMS is assumed to have a hierarchical architecture [86]. As illustrated in Fig. 3.12, each area of the power system has a PDC that concentrates the PMU measurements of this area and submits them to the monitoring center.

3.4.2.1 Candidate Attribute Subsets

The candidate attribute subsets are created based on the three following specific rules: 1) Within a candidate attribute subset, all the attributes are from the same area. 2) In area k ($k = 1, 2, \cdots, K$), three categories of pre-fault quantities measured by PMUs are used as the numerical attributes:

- Category 1: voltage magnitude V_i, for $i \in \mathrm{I}_k^{PMU}$
- Category 2: active power flow P_{ij}, reactive power flow Q_{ij} and current magnitude I_{ij}, for $i \in \mathrm{I}_k^{PMU}$ and $j \in \mathrm{N}(i)$
- Category 3: phase angle difference θ_{ij}, for $i, j \in \mathrm{I}_k^{PMU}$

where I_k^{PMU} denotes the collection of the buses with PMU installation within area k, and $\mathrm{N}(i)$ denotes the collection of the neighbor buses of bus i. An attribute subset of area k is created by including one voltage or flow measurement from each bus $i \in \mathrm{I}_k^{PMU}$ and all phase angle difference measurements from this area. 3) The index of contingencies is included as a categorical attribute in any attribute subset.

The criteria used in creating the attribute sets are elaborated below. By restricting the attributes of a subset to be the PMU measurements within the same area, the impact of some scenarios, i.e., when a PDC that concentrates PMU measurements within an area fails, is significantly reduced, since the small DTs using the PMU measurements from the other areas could still be viable. For a given bus, since Category 1 and Category 2 PMU measurements are co-located, it suffices to include only one of them in an attribute subset so that the redundancy within an attribute subset is minimal. Further, all measurable phase angle differences are included. This is because theoretical and empirical results (e.g., in [18]) suggest that angle differences contain important information regarding the level of stress in OCs, and thus are more likely to be the attributes critical to assessing transient instability. It is also worth noting that the Category 2 attributes from two different buses are unlikely to be redundant, in the sense that they are the measurements from different transmission lines, given the fact that PMUs could provide power flow measurements and it is usually unnecessary to place PMUs at both ends of a transmission line to achieve the full observability of power grids.

For convenience, let S_k denote the collection of candidate attribute subsets of area k. Then, the size of S_k is given by

$$M_k = \prod_{i \in \mathrm{I}_k^{PMU}} (3\deg(i) + 1) \tag{3.8}$$

where $\deg(i)$ denotes the degree of bus i, i.e., the number of buses that connect with bus i. Then, $\mathrm{S} = \bigcup_{k=1}^{K} \mathrm{S}_k$ is the collection of candidate attribute subsets.

3.4.2.2 Randomized Algorithm for Selecting Attribute Subsets

It is plausible to develop the randomized algorithm so as to achieve maximum randomness of the selected attribute subsets by maximizing the entropy of the weight distribution $\{p_{\mathbf{s}}, \mathbf{s} \in \mathrm{S}\}$. Without any other information of attribute, equal weights are usually used by existing random subspace methods (e.g., [87], [88]). Here, by adhering to guideline **G2**, an additional constraint is that the average availability of the randomly selected attribute subsets is above an acceptable level A_0. As a result, the weight distribution can be determined by solving the following problem:

$$\mathrm{P}_{\mathbf{s}}: \max_{\{p_{\mathbf{s}}, \mathbf{s} \in \mathrm{S}\}} \sum_{\mathbf{s} \in \mathrm{S}} p_{\mathbf{s}} \log_2 p_{\mathbf{s}}^{-1} \tag{3.9}$$

$$\text{s.t.} \quad \sum_{\mathbf{s} \in \mathrm{S}} p_{\mathbf{s}} A_{\mathbf{s}} \geq A_0$$

$$\sum_{\mathbf{s} \in \mathrm{S}} p_{\mathbf{s}} = 1$$

where $A_{\mathbf{s}}$ denotes the availability of an attribute subset $\mathbf{s}$. According to the rules for creating the candidate attribute subsets, it is easy to see that each of the attribute subsets of an area consists of exactly two measurements from each PMU within this area. Therefore, the availability of an attribute subset s of area k, which was formally defined in Section I as the probability that the measurements of s are successfully delivered to the monitoring center, equals that of the WAMS within area k, i.e.,

$$A_{\mathbf{s}} = A_k, \ \forall \mathbf{s} \in \mathrm{S}_k \tag{3.10}$$

In availability analysis of WAMS (e.g., in [83]), it is usually assumed that the availability of PMUs, PDCs and communication links are known (e.g., estimated from

past operating data) and independent from each other. Under these assumptions, the availability of the WAMS within area k is given by:

$$A_k = \prod_{i \in \mathrm{I}_k^{PMU}} (A_i^{PMU} A_i^{link}) \tilde{A}_k^{PDC} \tilde{A}_k^{link} \tag{3.11}$$

where A_i^{PMU}, A_i^{link}, $\tilde{A}_k^{PDC}$ and $\tilde{A}_k^{link}$ denote the availability of the PMU at bus i, the communication link from the PMU at bus i to the PDC, the PDC and the communication link from the PDC to the monitoring center, respectively. It is worth noting that (3.10) and (3.11) are derived for the case illustrated in Fig. 3.12, and thus may not be directly applicable to the cases with measurement redundancy. For example, when multiple dual use PMU/line relays are utilized in substations, the availability of bus voltage phasor measurements can be enhanced. The procedure for analyzing the availability of WAMS in case of redundancy can be found in the literature (e.g., [89]).

By taking (3.10) into account, it follows that the solution to problem $\mathrm{P}_\mathbf{s}$ in (3.9) has the following property.

Proposition 3.1: The optimal solution to $\mathrm{P}_\mathbf{s}$ in (3.9) takes the following form:

$$p_\mathbf{s}^* = p_k^* / M_k, \ \forall \mathbf{s} \in \mathrm{S}_k \tag{3.12}$$

where M_k is the size of S_k as defined in (3.8), and $\{p_k^*, k = 1, 2, \cdots, K\}$ is the solution to the following problem:

$$\tilde{\mathrm{P}}_\mathbf{s}: \min_{p_1, \cdots p_K} \sum_{k=1}^{K} p_k \log_2 (p_k / M_k) \tag{3.13}$$

$$\text{s.t.} \sum_{k=1}^{K} p_k A_k \geq A_0$$

$$\sum_{k=1}^{K} p_k = 1$$

Proof: Since $\mathrm{P_s}$ maximizes a concave function with affine constraints, the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for a solution to be optimal. Therefore,

$$(1+\ln p_{\mathbf{s}}^*)/\ln 2 - \lambda^* A_{\mathbf{s}} + \mu^* = 0, \ \forall \mathbf{s} \in \mathrm{S} \tag{3.14}$$

where λ^* and μ^* are the KKT multipliers for the two constraints of $\mathrm{P_s}$. Then, by taking the equality in (3.10) into account, it is easy to verify that $\mathrm{P_s}$ have the same value for all $\mathbf{s} \in \mathrm{S}_k$. Define $p_k = M_k p_{\mathbf{s}}$ for $\mathbf{s} \in \mathrm{S}_k$, then $\mathrm{P_s}$ reduces to $\tilde{\mathrm{P}}_{\mathbf{s}}$.

The above result leads to the following implementation of the randomized algorithm, as summarized in Algorithm 3.2. Further, it is also observed from (3.14) that the attribute subsets which have higher availability are assigned higher weights.

Algorithm 3.2: Randomized algorithm for selecting an attribute subset

---

1. Calculate M_k and A_k according to (3.8) and (3.11), respectively, for $k = 1, \cdots, K$.
2. Find $\{p_k, k = 1, \cdots, K\}$ by solving $\tilde{\mathrm{P}}_{\mathbf{s}}$ in (3.13).
3. Select an area k among the K areas with weight p_k.
4. For the chosen area k, select an attribute subset s from S_k with weight M_k^{-1}.

3.4.3 Proposed Approach for Online DSA with Missing PMU Measurements

First, L small DTs are trained offline by using randomly selected attribute subsets. In case of missing PMU measurements in online DSA, $\tilde{L}$ ($\tilde{L} \le L$) viable small DTs are identified, and are assigned different voting weights. Specifically, the results of

performance re-check in near real-time are utilized to quantify these voting weights. Finally, the security classification decisions for the new OCs in online DSA are obtained via weight voting of the $\tilde{L}$ viable small DTs.

3.4.3.1 Offline Training

Given a collection of training cases $\{\mathbf{x}_n, y_n\}_{n=1}^N$ and candidate attribute subsets S , a primary objective of offline training is to obtain small DTs $\{h_1, \cdots, h_L\}$ so that the majority voting of them, i.e., $F_L(\mathbf{x}) = \sum_{l=1}^{L} h_l(\mathbf{x})$ could fit the training data. The iterative process to obtain an F_L is summarized in Algorithm 3.3.

Algorithm 3.3: Offline training using the random subspace method

---

Input: Training cases $\{\mathbf{x}_n, y_n\}_{n=1}^N$, $\varepsilon_0 \in (0,1)$

Initialization: $F_0 = \mathbf{0}$

For $l = 1$ **to** L **do**

Select an attribute subset $\mathbf{s}_l$ by using Algorithm 3.2.

Find a small DT h_l by solving $\mathrm{P}_{DT}^{(l)}$ in (3.15) using the CART algorithm.

$F_l \leftarrow F_{l-1} + h_l$

End For

In the l-th iteration, a small DT h_1 is first obtained by solving the following problem:

$$\mathrm{P}_{DT}^{(l)} : \min_{h_l} \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}_{\{y_n \neq h_l(\mathbf{x}_n^l)\}} \tag{3.15}$$

where $\mathbf{x}_n^l$ denotes the measurements of the attribute subset $\mathbf{s}_l$. It is well-known that the problem in (3.15) is NP-complete [90]. Here, the CART [11] algorithm is employed to find a sub-optimal DT, by using misclassification error rate as the splitting cost function. It is clear from (3.15) that equal weights, i.e., $1/N$, are assigned to all training data. When historical data that identifies potential weak spots of the system is available, these data can be integrated by assigning higher weights, and by replacing $1/N$ with unequal data weights.

3.4.3.2 Near Real-time Performance Re-check

In near real-time, a more accurate prediction of the imminent OC in online DSA can be made. Then, a collection of new cases $\{\tilde{\mathbf{x}}_n, \tilde{y}_n\}_{n=1}^{\tilde{N}}$ are created in a similar manner to that in offline training and used to re-evaluate the accuracy of the L small DTs. The re-check results are then utilized by the boosting process in online DSA. In case of variations between the OCs used in offline training and the new OCs in online DSA, near real-time re-check is also a critical step to make sure that the small DTs still work well.

3.4.3.3 Online DSA

The results of near real-time re-check $\{\tilde{h}_l(\tilde{\mathbf{x}}_n^l), \tilde{y}_n\}_{n=1}^{\tilde{N}}, \forall l = 1, \cdots, \tilde{L}$, are utilized to choose a few viable small DTs to be used in online DSA and calculate the corresponding voting weights via a process of boosting small DTs. In order to make best use of existing DTs, the viable small DTs in online DSA include the small DTs without any missing PMU measurement and non-empty degenerate small DTs.

3.4.3.3.1 Degenerate Small DTs

A degenerate small DT is obtained by collapsing the subtree of an internal node with missing PMU measurement into a leaf node. Specifically, a small DT degenerates to a non-empty tree if the PMU measurements used by the internal nodes other than the root node are missing, an example of which is illustrated in Fig. 3.13. Further, since each internal node of the original small DT is also assigned a decision in building the DT, the

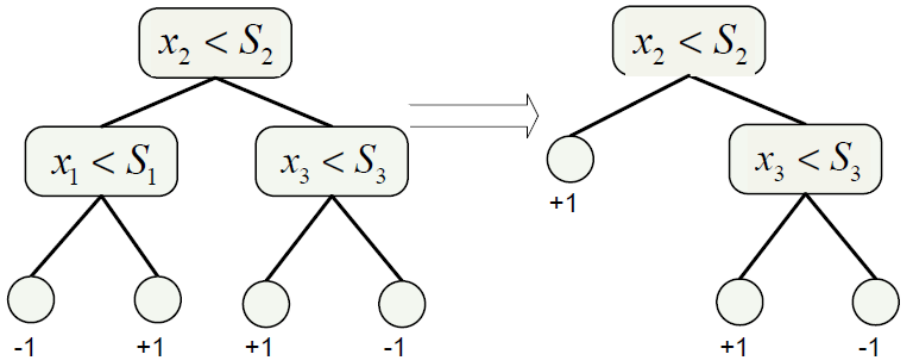


Figure 3.13 Degeneration of a small DT as a result of missing PMU measurements of attribute x_1 when node ($x_1 < S_1$) is originally assigned +1.

new leaf node of the degenerate small DT is assigned the same decision as the original internal node. Therefore, for a non-empty degenerate small DT, the re-check results on the $\tilde{N}$ new cases could be easily obtained.

3.4.3.3.2 Weighted Voting of Viable Small DTs

Let $\tilde{\mathrm{H}}$ be the collection of viable small DTs. Then, weighted voting of the viable small DTs in $\tilde{\mathrm{H}}$ is utilized to obtain the security classification decisions of online DSA, due to the following two reasons. First, in case that some small DTs degenerate to empty trees and the accuracy of non-empty degenerate small DTs degrades, weighted voting could improve the overall accuracy compared to majority voting, provided that the voting

weights are carefully assigned based on the re-check results of the viable small DTs. Second, even though all the small DTs are viable, choosing the small DTs with proper voting weights based on their accuracy can still be a critical step to guarantee accurate decisions. This is because small DTs trained offline fit the training cases that are created based on day-ahead prediction, while the re-check results on the $\tilde{N}$ new cases contain more relevant information on assessing the security of the imminent OCs in online DSA.

In the proposed approach, weighted voting of small DTs in $\tilde{\mathrm{H}}$ is implemented via a boosting process. Following the method in [66], initially with $\tilde{F}_0$ as a zero function, a small DT $\tilde{h}_l \in \tilde{\mathrm{H}}$ is first identified and added to $\tilde{F}_{l-1}$, i.e.,

$$\tilde{F}_l = \tilde{F}_{l-1} + a_l \tilde{h}_l \tag{3.16}$$

iteratively for $l = 1, 2, \cdots, \tilde{L}$, so that the cost function, i.e.,

$$\hat{C}(\tilde{F}_{\tilde{L}}) = \frac{1}{\tilde{N}} \sum_{n=1}^{\tilde{N}} \log_2 (1 + \mathrm{e}^{-\tilde{y}_n \tilde{F}_{\tilde{L}}(\tilde{\mathbf{x}}_\mathbf{n})}) \tag{3.17}$$

is minimized in a gradient descent manner. In the boosting process, $\tilde{h}_l$ is identified by solving the following problem:

$$\tilde{\mathrm{P}}_{DT}^{(l)} : \min_{h_l \in \tilde{\mathrm{H}}} \frac{1}{\tilde{N}} \sum_{n=1}^{\tilde{N}} w_n^{(l)} \mathbf{1}_{\{\tilde{y}_n \neq \tilde{h}_l(\tilde{\mathbf{x}}_n^l)\}} \tag{3.18}$$

and the data weights and voting weight are given by

$$\begin{cases} w_n^{(l)} = \dfrac{1}{1 + \mathrm{e}^{\tilde{y}_n \tilde{F}_{l-1}(\tilde{\mathbf{x}}_n)}} & n = 1, \cdots, \tilde{N} \\ a_l = \underset{a \in \mathrm{R}^+}{\operatorname{argmin}}\, g_l(a) & l = 1, \cdots, \tilde{L} \end{cases} \tag{3.19}$$

where $g_\ell(a) \underline{\underline{\Delta}} \hat{C}\left(\tilde{F}_{\ell-1} + a\tilde{h}_\ell\right)$. Boosting viable small DTs in online DSA is summarized in Algorithm 3.4.

Algorithm 3.4: Boosting viable small DTs for online DSA

Input: Re-check results $\{\tilde{h}_l(\tilde{\mathbf{x}}_n^l), \tilde{y}_n\}_{n=1}^{\tilde{N}}$, $\forall l = 1, \cdots, \tilde{L}$

Initialization: $\tilde{F}_0 = \mathbf{0}$

For $l = 1$ **to** $\tilde{L}$ **do**

Calculate the data weights according to (3.19).

Find a small DT $\tilde{h}_l$ by solving $\tilde{\mathrm{P}}_{DT}^{(l)}$ in (3.18) using the CART algorithm.

Calculate the voting weight a_l according to (3.18).

$\tilde{F}_l \leftarrow \tilde{F}_{l-1} + a_l \tilde{h}_l$.

End For

3.4.3.4 Further Discussion

Through detailed complexity analysis, it is shown that the low computational complexity of the online processing renders that the time criticality of online DSA would not be compromised when the proposed approach is used. Specifically, the computationally intensive part of the online processing stage is the boosting process that consists of calculating the data weights $w_n^{(l)}$, solving $\tilde{\mathrm{P}}_{DT}^{(l)}$ and calculating the voting weights a_l of small DTs. According to (3.19), calculating the data weights requires evaluating $\tilde{F}_l$ for the new cases, which could be easily obtained from the re-check results of the small DTs. Therefore, it is easy to see that the complexity in calculating the data

weights is $\mathrm{O}(\tilde{N})$. Solving $\tilde{\mathrm{P}}_{DT}^{(l)}$ boils down to searching for the small DT in $\tilde{\mathrm{H}}$ that has the least weighted misclassification error. Since the re-check results of the small DTs in $\tilde{\mathrm{H}}$ for the new cases are already known, the optimal small DT could be found by comparing the weighted misclassification errors of the small DTs in $\tilde{\mathrm{H}}$. Therefore, the complexity in solving $\tilde{\mathrm{P}}_{DT}^{(l)}$ is $\mathrm{O}(\tilde{L}\tilde{N})$. In the l-th iteration of the boosting process, the voting weight is obtained by minimizing $g_l(a)$. It is easy to verify that $g_l'(0) < 0$ and $g_l''(a) > 0$ holds for $a \in \mathrm{R}^+$. Therefore, $g_l(a)$ has a unique minimum in R^+ that could be found by using standard numerical methods (e.g., Newton's methods). Further, since $g_l(a)$ is convex, standard numerical methods could find the minimum in a few iterations. In each iteration, $\tilde{F}_{l-1} + a\tilde{h}_l$ needs to be evaluated for all the $\tilde{N}$ new cases. Therefore, the complexity in calculating the voting weight for a small DT is $\mathrm{O}(\tilde{N})$. Summarizing, the overall computational complexity of the boosting process is $\mathrm{O}(\tilde{L}^2\tilde{N})$.

The proposed approach above relates to that in [66] in the following sense: small DTs are utilized in both approaches; new cases are used in near real-time for accuracy guarantee by both approaches; the security classification decisions of online DSA are both obtained via a weighted voting of small DTs. However, the two approaches are tailored towards different application scenarios. The approach proposed here is more robust to missing PMU measurements, while the approach in [66] could give accurate decisions with less effort in offline training when the availability of PMU measurements is sufficiently high. The major differences of the two approaches are outlined as follows. First, the small DTs in the proposed approach are trained by using attribute subsets for

robustness, whereas the entire set of attributes is used in [66]. Second, the usage of new cases in near real-time is different. In [66], the new cases are used to update the small DTs, whereas in the proposed approach, the new cases are only used to re-check the performance of viable small DTs so as to quantify the voting weights.

3.4.4 Case Study

3.4.4.1 Test System

The IEEE 39-bus system [35] is used as the test system which contains 39 buses, 10 generators, 34 transmission lines and 12 transformers. Particularly, G1 represents the

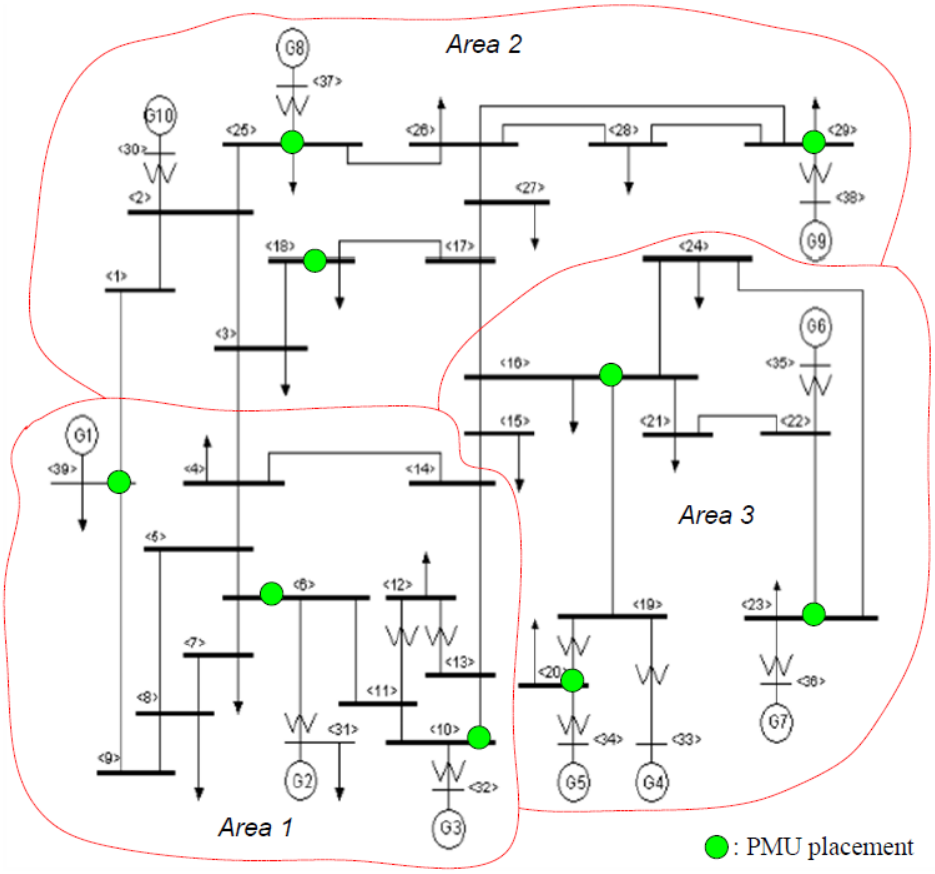


Figure 3.14 The IEEE 39-bus system in three areas and PMU placement

aggregated generation from the rest of eastern interconnection [35]. In this case study, the

test system is assumed to consist of three areas. The three areas together with the PMU placement are illustrated in Fig. 3.14. It is worth noting that the PMU placement guarantees the full observability of the test system when the constraints at zero-injection buses are taken into account.

3.4.4.2 Knowledge base

The knowledge base only consists of the OCs that are both pre-contingency secure and $N-1$ secure. The cases in the knowledge base are created from the combinations of the PMU measurements of the OCs and their transient security classification decisions for a few selected $N-2$ contingencies. In this case study, the power flow solutions of an OC are used as the "PMU measurements."

3.4.4.2.1 OC Generation

The OC given in [35] is used as the base OC. Following the method in [19], more OCs are generated for offline training, by randomly changing the bus loads (both active and reactive power) within 90% to 110% of their original values in the base OC; for the OCs generated for near real-time re-check and online DSA test, the bus loads varies from 97% to 103% of their original values in the base OC. The rationale for the above percentage values is that offline training is usually carried out day/hours ahead, and thus the predicted OCs can have a larger prediction error than those in near real-time. The power flows of each generated OC are solved using the power flow and short circuit analysis tool (PSAT) [77], followed by a limit check such that the generated OCs with any pre-contingency overloading or voltage/angle limit violations are excluded from the knowledge base.

3.4.4.2.2 Critical Contingencies

The loss of any of the 46 components (i.e., the 34 transmission lines, the 9 generator-transformer pairs and the 3 transformers at (11,12), (12,13) and (19,20)) is considered as an $N-1$ contingency. Due to the large number of possible $N-2$ contingencies, only a few of them are selected. Intuitively, a severe impact on the security of power systems is more likely if a second component gets overloaded after the loss of the first component. As such, the $N-2$ contingencies are selected in the following manner. First, each of the aforementioned 46 components is removed from the test system. Then, power flows are re-solved and limit check is rerun for the base OC using PSAT. The first removed component and any overloaded component are regarded as the removed pair of an $N-2$ contingency. As a result, 15 pairs are identified, as listed in Table 3.4.

Table 3.4 1st and 2nd removed components of the selected N-2 contingencies

line(4,14), line(6,11)	line(6,11), line(4,14)	line(6,11), line(13,14)
line(6,11), line(10,13)	line(10,11), line(10,13)	line(10,13), line(6,11)
line(10,13), line(10,11)	line(13,14), line(6,11)	line(13,14), line(10,11)
line(16,21), line(23,24)	line(21,22), line(23,24)	line(21,22), line(22,23)
line(21,22), line(16,24)	line(23,24), line(16,21)	line(23,24), line(21,22)

3.4.4.2.3 Transient Security Assessment

Transient security assessment tool (TSAT) [77] is used to assess the transient performance of the OCs that are pre-contingency secure. To create a contingency in TSAT, the "three-phase short circuit to ground" fault is applied at either of the two terminal buses of the first removed component with a primary clearing time of 4 cycles.

Therefore, 92 $N-1$ contingencies and 30 $N-2$ contingencies are created. The "power angle-based stability margin" defined in TSAT [77] is used as the transient stability index.

3.4.4.3 Test Results and Discussion

Three other approaches are used as benchmarks, including a single DT using surrogate, an RF using surrogate, and an RF without using surrogate. Following [88], unpruned DTs are used in RFs; in RFs, all training cases are used to build a single DT; in each split of DTs, a number of $\log_2 P+1$ attributes are randomly selected (where P =96 according to Column 3 of Table 3.5); the optimal number of DTs in the forest is determined through out-of-bag validation [88]. Specifically, for the former two benchmark approaches, surrogate attributes are obtained from those which are not co-located with the primary attributes; for the third benchmark approach, degenerated DTs are used.

Table 3.5 Data used by Algorithm 3.2 for the IEEE 39-bus test system

Area	Placement	Number of attributes			M_k	A_k	p_k
		Category 1	Category 2	Category 3			
1	8, 13, 39	3	24	3	700	*b*	0.28
2	18, 25, 29	3	24	3	700	*b*	0.28
3	16, 20, 23	3	30	3	1120	*b*	0.44

3.4.4.3.1 Attribute Subsets

The hypothetical WAMS for the test system has a hierarchical architecture similar to that in Fig. 3.12. Based on the evaluation results of the reference [91], it is assumed that all the PMUs have the same availability a ($a \in$ [0.979975, 0.998920]), and all the communication links from PMUs to PDC have the same availability A^{link} =0:999. Further, the availability of the PDC and the communication link from the PDC to the monitoring center is assumed to be 1. Let $b = (0.999a)^3$, and thus $b \in$ [0.938299,0.993776]. Then, it follows that when $A_0 \le b$, the solution to $\tilde{P}_s$ in (3.13) exists, as given in Table 3.5. In what follows, the data in Table 3.5 is explained in detail. Specifically, Column 2 provides the indices of PMU buses, which can also be seen from Fig. 3.14. Column 3 contains the number of attributes for the three categories defined in Section 3.4.2.1. Take area 1 for example, there are 3 voltage magnitude attributes, 24 transmission line (including power flow and current magnitude) attributes, and another 3 attributes from voltage phase angle difference. Given the system topology and availability information, M_k in column 4 and A_k in column 5 are calculated using (3.8) and (3.11), respectively. Then, p_k is obtained by solving (3.13).

3.4.4.3.2 Offline Training

N_{OC} =200 generated OCs which are both pre-contingency and $N-1$ contingency secure are used for offline training. Combining the generated OCs with their transient security classification decisions for the N_C =30 selected $N-2$ contingencies, are used to generate the N =6000 cases in the knowledge base. The size and the number of small DTs are determined by bias-variance analysis [68] and V -fold cross validation [18]. In

this case study, L =40 and J =3 are used by the proposed approach; 45 DTs are used in the two RF-based approaches.

3.4.4.3.3 Near Real-time Re-check

By following the procedure described in Section 3.4.3.2, 100 OCs are generated for performance re-check. The DTs trained offline are applied to the new cases; the classification results are compared with the actual security classification decisions of the new cases. Then, these re-check results are used by Algorithm 3.4 to quantify the voting weights of DTs

3.4.4.3.4 Online DSA Test

Another 100 OCs are generated for testing, by following the procedure described in Section V.B. Recall that the availability of PDC and the communication links for PDCs is 1, and then it can be seen from Fig. 3.13 and Fig. 3.14 that the total number of failure scenarios of all PMUs and links can be reduced to 512 (2^9, since there are 9 pairs of PMUs and links). The online DSA test is repeated for all failure scenarios, by identifying the missing PMU measurements and viable small DTs, calculating the voting weights of viable small DTs, and evaluating the misclassification error rate. The misclassification error in online DSA is calculated by:

$$\bar{e}(\tilde{F}) = \sum_{k=1}^{512} \text{Prob}(\Omega(k))\text{e}(\tilde{F} \mid \Omega(k)) \tag{3.20}$$

where, $\Omega(k)$ denote the k-th failure scenario; $\text{Prob}(\Omega(k))$ denotes the probability to happen of $\Omega(k)$, which can be easily calculated by using the assumed availability; $\text{e}(\tilde{F} \mid \Omega(k))$ denotes the misclassification error of $\tilde{F}$ under failure scenario $\Omega(k)$

($e(\tilde{F}\,|\,\Omega)$ is set to be 1 when all PMUs fail).

The test are performed for various values of availability a, the test results of which are illustrated in Fig. 3.15. It is observed that the performance of the benchmark approaches is comparable to that of the proposed approach only around b =1. However, the gap becomes more significant as b decreases.

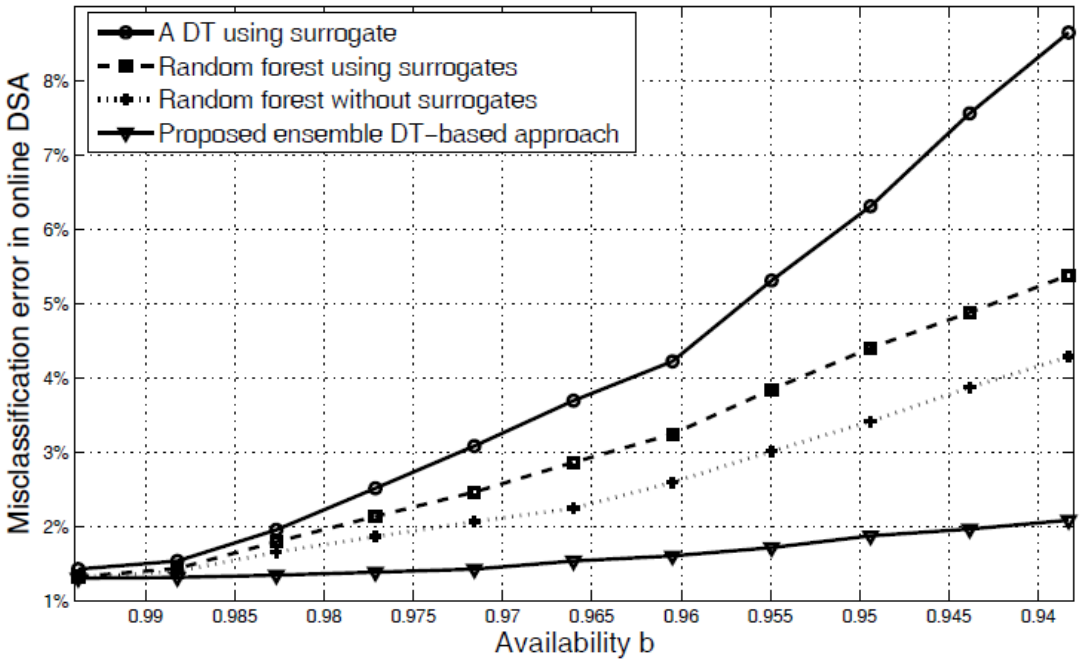


Figure 3.15 Performance on online DSA in case of missing PMU measurements

3.4.4.3.5 Impact of Measurement Noise

In reality, PMU data may contain measurement noise. Actually, besides missing PMU data, noisy PMU data can be another important issue to online DSA and many other PMU measurement-based applications. Following the approach in [41], numerical experiment is carried out to study the impact of measurement noise on the performance of the proposed approach.

For convenience, let $V\angle\theta_V$ and $I\angle\theta_I$ denote a voltage/ current phasor respectively; let $\tilde{V}\angle\tilde{\theta}_V$ and $\tilde{I}\angle\tilde{\theta}_I$ be the corresponding measurement. For PMUs complying with IEEE C37.118 standard [39], a measurement should have a total vector error (TVE) less than 1%, i.e.,

$$\left|\frac{\tilde{V}\angle\tilde{\theta}_V - V\angle\theta_V}{V\angle\theta_V}\right| < 1\% \tag{3.21}$$

$$\left|\frac{\tilde{I}\angle\tilde{\theta}_I - I\angle\theta_I}{I\angle\theta_I}\right| < 1\% \tag{3.22}$$

For convenience, let n_V and n_I denote the measurement noise in $\tilde{V}\angle\tilde{\theta}_V$ and $\tilde{I}\angle\tilde{\theta}_I$ respectively. In order to generate measurement complying with the above specifications, the complex noise n_V and n_I are randomly generated, by using the following density functions (note that other density functions can be also used) properly scaled and truncated from standard complexity Gaussian distributions:

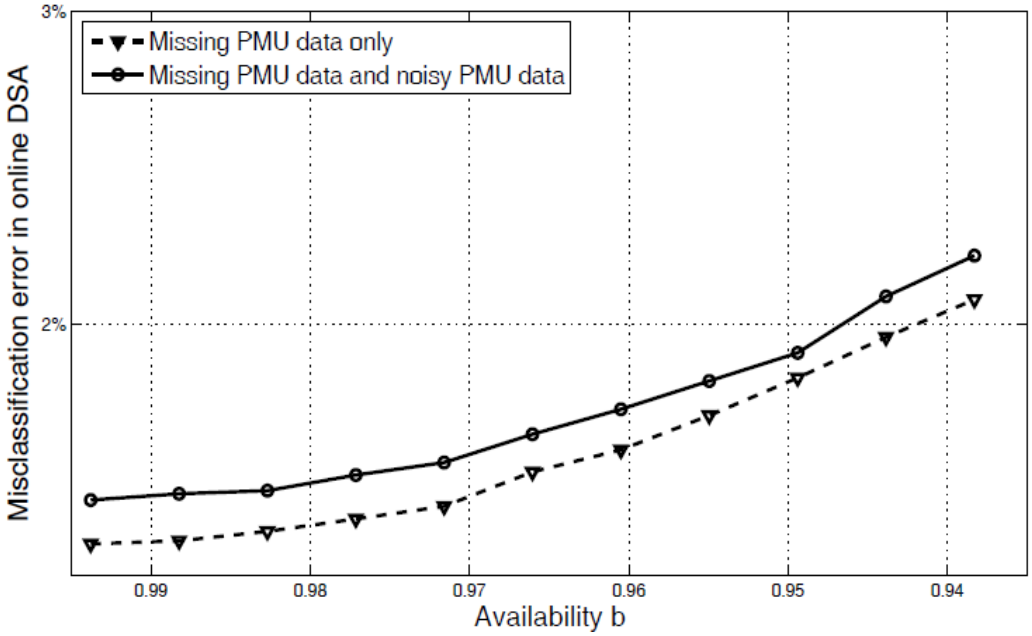


Figure 3.16 Impact of measurement noise

$$f(n_V)=\begin{cases}\dfrac{9}{\pi\left(1-e^{-9}\right)10^{-4}V^2}e^{-\frac{9|n_V|^2}{10^{-4}V^2}} & if\ \ |n_V|\le 10^{-2}V \\ 0 & o.w.\end{cases} \tag{3.23}$$

$$f(n_I)=\begin{cases}\dfrac{9}{\pi\left(1-e^{-9}\right)10^{-4}I^2}e^{-\frac{9|n_I|^2}{10^{-4}I^2}} & if\ \ |n_I|\le 10^{-2}I \\ 0 & o.w.\end{cases} \tag{3.24}$$

Then, it is clear that all noisy measurements have TVE not more than 1%, and are complex Gaussian distributed within their support. The generated random measurement noise is added to the both training and testing data. The test results are provided in Fig. 3.16.

3.5 Conclusions

In this project, an online DSA scheme based on ensemble DT learning is proposed to handle the OC variations and topology changes that are likely to occur during the operating horizon. The proposed scheme is applied to a practical power system, and the results of a case study demonstrate the performance improvement brought by boosting unpruned small DTs over a single DT. Compared to single DTs, the classification model obtained from ensemble DT learning often have higher accuracy, and lend themselves to cost-effective incorporation of new training cases. The results presented here also provide an insight into the possibilities of other ensemble DT learning techniques, e.g., random forest, in handling the challenges of online DSA.

Further, in order to mitigate the impact of missing PMU measurements in online DSA, a random subspace method that utilizes the topological information of WAMS and the availability of PMU measurement has been developed and incorporated into the ensemble DT learning. In particular, the various possibilities of missing PMU measurements in online DSA can make off-the-shelf DT-based techniques (a single DT, RF, etc) fail to deliver the same performance as expected. The proposed ensemble DT-based approach exploits the locational information and the availability of PMU measurements in randomly selecting attribute subsets, and utilizes the re-check results to re-weight the DTs in the ensemble. These special treatments developed from a better understanding of power system dynamics guarantee that the proposed approach can achieve better performance than directly applying off-the-shelf DT-based techniques.

4 References

[1] Bittencourt, H.R.; Clarke, R.T., "Use of classification and regression trees (CART) to classify remotely-sensed digital images," *Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International* 21-25 July 2003

[2] Galvan, F.; Wells, C.H., "Detecting and managing the electrical island created in the aftermath of Hurricane Gustav using Phasor Measurement Units (PMUs)," *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, 19-22 April 2010

[3] Wells, C.H., "Redundancy and Reliability of Wide-Area Measurement Synchrophasor Archivers." *OSIsoft, LLC and Schweitzer Engineering Laboratories, Inc.* 2011

[4] Kolluri, S.; Mandal, S.; Galvan, F.; Thomas, M., "Island Formation in Entergy Power Grid during Hurricane Gustav," *Power & Energy Society General Meeting 2009. PES '09. IEEE*, 26-30 July 2009

[5] Jervis, R., "Hurricane Isaac pounds Louisiana, water pours over levee," USA Today 29 August 2012

[6] P. Kundur, *Power System Stability and Control*. New York: McGraw-Hill, 1994.

[7] M. Kezunovic, C. Zheng, and C. Pang, "Merging PMU, operational, and non-operational data for interpreting alarms, locating faults and preventing cascades," 43rd Hawaii International Conference on System Sciences (HICSS), Jan. 2010.

[8] C. Zheng, Y. Dong, O. Gonen, and M. Kezunovic "Data integration used in new applications and control center visualization tools," IEEE PES General Meeting, Minneapolis, USA, July 2010.

[9] G. Rogers, *Power System Oscillations*. Boston: Kluwer Academic Publishers, 2000.

[10] T. V. Cutsem and C. Vournas, *Voltage Stability of Electric Power Systems*. Boston: Kluwer Academic Publishers, 1998.

[11] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Pacific Grove: Wadsworth, 1984.

[12] Dan Steinberg and Mikhail Golovnya, *CART 6.0 User's Manual*. San Diego, CA: Salford Systems, 2006.

[13] L. Wehenkel, T. V. Cutsem, and M. Ribbens-Pavella, "An artificial intelligence framework for on-line transient stability assessment of power systems," *IEEE Trans. Power Syst.*, Vol. 4 No. 2, pp. 789-800, May 1989.

[14] L. Wehenkel, M. Pavella, E. Euxibie, and B. Heilronn, "Decision tree based transient stability method a case study," *IEEE Trans. Power Syst*., Vol. 9, No. 1, pp. 459-469, Feb. 1994.

[15] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, "Decision trees for real-time transient stability prediction," *IEEE Trans. Power Syst.*, Vol. 9, No. 3, pp. 1417-1426, Aug. 1994.

[16] K. Sun, S. Likhate, V. Vittal, V. S. Kolluri, and S. Mandal, "An online dynamic security assessment scheme using phasor measurements and decision trees," *IEEE Trans. Power Syst.*, Vol. 9, No. 1, pp. 459-469, Nov. 2007.

[17] T. V. Cutsem, L. Wehenkel, M. Pavella, B. Heilbronn, and M. Goubin, "Decision tree approaches to voltage security assessment," *Proc. Inst. Elect. Eng.*, Vol. 140, No. 3, pp. 189-198, May 1993.
[18] R. Diao, K. Sun, V. Vittal, et al., "Decision tree-based online voltage security assessment using PMU measurements," *IEEE Trans. Power Syst.*, Vol. 24, No. 2, pp. 832-839, May 2009.
[19] I. Kamwa, S. R. Samantaray, and G. Joos, "Catastrophe predictors from ensemble decision-tree learning of wide-area severity indices," *IEEE Trans. Smart Grid*, Vol. 1, No. 2, pp. 144-158, Sep. 2010.
[20] R. Diao, V. Vittal, and N. Logic, "Design of a real-time security assessment tool for situational awareness enhancement in modern power systems," *IEEE Trans. Power Syst.*, Vol. 25, No. 2, pp. 957-965, May 2010.
[21] S. P. Teeuwsen, I. Erlich, M. A. El-Sharkawi, and U. Bachmann, "Genetic algorithm and decision tree-based oscillatory stability assessment," *IEEE Trans. Power Syst.*, Vol. 21, No. 2, pp. 746-753, May 2006.
[22] I. Kamwa, S. R. Samantaray, and G. Joos, "Development of rule-based classifiers for rapid stability assessment of wide-area post-disturbance records," *IEEE Trans. Power Syst.*, Vol. 24, No. 1, pp. 258-270, Feb. 2009.
[23] I. Kamwa, S. R. Samantaray, and G. Joos, "On the accuracy versus transparency trade-off of data-mining models for fast-response PMU-based catastrophe predictors," *IEEE Trans. Smart Grid*, Vol. 3, No. 1, pp. 152-161, 2012.
[24] D. Q. Zhou, U. D. Annakkage, and A. D. Rajapakse, "Online monitoring of voltage stability margin using an artificial neural network," *IEEE Trans. Power Syst.*, Vol. 25, No. 3, pp. 1566-1574, August 2010.
[25] F. R. Gomez, A. D. Rajapakse, U. D. Annakkage, and I. T. Fernando, "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements," *IEEE Trans. Power Syst.*, Vol. 26, No. 3, pp. 1474-1483, August 2011.
[26] C. Zheng and M. Kezunovic, "Impact of wind generation uncertainty on power system small disturbance voltage stability: a PCM-based approach," *Electric Power Systems Research*, Vol. 84, No. 1, pp 10-19, March 2012.
[27] C. Zheng and M. Kezunovic "Distribution system voltage stability analysis with wind farms integration," IEEE PES 42nd North American Power Symposium (NAPS), Arlington, USA, September, 2010.
[28] V. Ajjarapu and C. Christy, "The continuation power flow: a tool for steady state voltage stability analysis," *IEEE Trans. Power Syst.*, Vol. 7, No. 1, pp. 416–423, Feb. 1992.
[29] C. Zheng, V. Malbasa, and M. Kezunovic, "A fast stability analysis scheme based on classification and regression tree," IEEE Conference on Power System Technology (POWERCON), Auckland, New Zealand, October 2012.
[30] Y. Zhou and V. Ajjarapu, "A fast algorithm for identification and tracing of voltage and oscillatory stability margin boundaries," *Proceedings of IEEE*, Vol. 93, No. 5, pp. 934-946, 2005.
[31] PSS/E-32 Program Operation Manual, USA: Power Technologies, Inc., Oct. 2010.
[32] Mathworks Inc. MATLAB R2012b User's Guide. [Online]. Available: http://www.mathworks.com.

[33] S. P. Teeuwsen, I. Erlich, M. A. El-Sharkawi, and U. Bachmann, "Genetic algorithm and decision tree-based oscillatory stability assessment," *IEEE Trans. Power Syst*., Vol. 21, No. 2, pp 746-753, May 2006.
[34] P. M. Anderson and A. A. Fouad, *Power System Control and Stability*. pp. 38, The Iowa State University Press, Ames, Iowa, 1977.
[35] M. A. Pai, *Energy Function Analysis for Power System Stability*. Boston, MA: Kluwer, 1989.
[36] R. G. D. Steel and J. H. Torrie, *Principles and Procedures of Statistics*. New York: McGraw-Hill, 1960.
[37] R. F. Nau. (2005, February). Forecasting [Online]. Available: http://www.duke.edu/~rnau/rsquared.htm.
[38] Electric Power Research Institute, "DC multi-infeed study," EPRI TR-104586s, Projects 2675-04-05, Final Report 1994.
[39] IEEE Standard for Synchrophasors for Power Systems, IEEE Std. C37.118-2005, 2005.
[40] Y. Yang, X. Wu, and X. Zhu, "Mining in anticipation for concept change: proactive-reactive prediction in data streams," *Data Mining and Knowledge Discovery*, no. 13, pp. 261-289, 2006.
[41] C. Zheng, V. Malbasa, and M. Kezunovic, "Regression tree for stability margin prediction using synchrophasor measurements," *IEEE Trans. Power Syst*., to be published.
[42] V. Malbasa, C. Zheng, M. Kezunovic, "Power system online stability margin estimation using active learning and synchrophasor data," manuscript accepted by PowerTech 2013, to be presented.
[43] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin-Madison, 2009.
[44] D. D. Lewis, W. A. Gale, "A sequential algorithm for training text classifiers," Proceedings of the 17th ACM SIGIR Conference on Research and Development in Information retrieval, pp. 3-12, 1994.
[45] L. Shi, Y. Zhao, J. Tang, "Batch mode active learning for networked data," *ACM Transactions on Intelligent Systems and Technology*, Vol. 3, No. 2, pp. 33, 2012.
[46] V. Hodge, J. Austin, "A survey of outlier detection methodologies," Artificial Intelligence Review, Vol. 22, No. 2, pp. 85-126, 2004.
[47] C. Chang, and C. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, pp. 27:1-27:27, 2011.
[48] J. Platt, "Probablistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers,* Vol. 10, No. 3, pp. 61-74, 1999.
[49] Y. Dong, C. Zheng, and M. Kezunovic, "Enhancing accuracy while reducing computation complexity for voltage-sag-based distribution fault location," *IEEE Trans. Power Delivery*, Vol. 28, No. 2, pp. 1202-1212, April 2013.
[50] R. Kumaresan and D.W. Tufts, "Estimating the parameters of exponentially damped sinusoids and pole-zero modeling in noise," *IEEE Trans. Acoustics, Speech, and Signal Processing*, pp. 833-840, Dec. 1982.

[51] R. Kumaresan, D.W. Tufts, and L.L. Scharf, "A Prony method for noisy data: choosing the signal components and selecting the order in exponential signal models," *Proc. IEEE*, pp. 230-233, February 1984.
[52] J. F. Hauer, C. J. Demeure, and L. L. Scharf, "Initial results in Prony analysis of power system response signals," *IEEE Trans. Power Syst.*, vol. 5, pp. 80-89, Feb. 1990.
[53] J. F. Hauer, "Applications of Prony analysis to the determination of modal content and equivalent models for measured power system response," *IEEE Trans. Power Syst.*, vol. 6, pp. 1062–1068, Aug. 1991.
[54] J. W. Pierre, D. J. Trudnowski, and M. K. Donnelly, "Initial results in electromechanical mode identification from ambient data," *IEEE Trans. Power Syst.*, vol. 12, no. 3, pp. 1245–1251, Aug. 1997.
[55] R. W. Wies, J. W. Pierre, and D. J. Trudnowski, "Use of ARMA block processing for estimating stationary low-frequency electromechanical modes of power systems," *IEEE Trans. Power Syst.*, vol. 18, no. 1, pp. 167–173, Feb. 2003.
[56] I. Kamwa, G. Trudel, and L. Gerin-Lajoie, "Low-order black-box models for control system design in large power systems," *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 303–311, Feb. 1996.
[57] C. Zheng, V. Malbasa, and M. Kezunovic, "Online estimation of oscillatory stability using synchrophasors and a measurement-based approach," submitted to 17th International Conference on Intelligent System Applications to Power Systems (ISAP 2013), under review.
[58] N. Zhou, J. W. Pierre, and J. Hauer, "Initial results in power system identification from injected probing signals using a subspace method," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1296–1302, Aug. 2006.
[59] D. J. Trudnowski, J. M. Johnson, and J. F. Hauer, "Making Prony analysis more accurate using multiple signals," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 226–231, Feb. 1999.
[60] P. Sauer, K. L. Tomsovic, and V. Vittal, Dynamic Security Assessment, 2nd ed., ser. The Electric Power Engineering Handbook. CRC Press, 2007, chapter 15, pp. 1–10.
[61] V. Miranda, J. Fidalgo, J. Lopes, and L. Almeida, "Real time preventive actions for transient stability enhancement with a hybrid neural network optimization approach," *IEEE Trans. Power Syst.*, vol. 10, no. 2, pp. 1029–1035, May 1995.
[62] C. Jensen, M. El-Sharkawi, and R. Marks, "Power system security assessment using neural networks: feature selection using Fisher discrimination," *IEEE Trans. Power Syst.*, vol. 16, no. 4, pp. 757–763, Nov. 2001.
[63] Kamwa, R. Grondin, and L. Loud, "Time-varying contingency screening for dynamic security assessment using intelligent-systems techniques," *IEEE Trans. Power Syst.*, vol. 16, no. 3, pp. 526–536, Aug. 2001.
[64] L. Moulin, A. da Silva, M. El-Sharkawi, and I. Marks, R.J., "Support vector machines for transient stability analysis of large-scale power systems," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 818–825, May 2004.
[65] Rajapakse, F. Gomez, K. Nanayakkara, P. Crossley, and V. Terzija, "Rotor angle instability prediction using post-disturbance voltage trajectories," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 947–956, May 2010.

[66] M. He, J. Zhang, and V. Vittal, "A data mining framework for online dynamic security assessment: decision trees, boosting, and complexity analysis," in *IEEE PES Innovative Smart Grid Technologies (ISGT)*, Jan. 2012, pp. 1–8.
[67] Y. Xu, Z. Y. Dong, J. H. Zhao, P. Zhang, and K. P. Wong, "A reliable intelligent system for real-time dynamic security assessment of power systems," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1253–1263, Aug. 2012.
[68] T. Hastie, R. Tibshirani, and J. Friedman, the Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, ser. Springer Series in Statistics. Springer-Verlag, 2008.
[69] Genc, R. Diao, V. Vittal, S. Kolluri, and S. Mandal, "Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1611–1619, Aug. 2010.
[70] Y. Freund and R. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.
[71] Niculescu-mizil and R. Caruana, "Obtaining calibrated probabilities from boosting," in Proc. 21st Conference on Uncertainty in Artificial Intelligence (UAI 05), AUAI Press. AUAI Press, 2005.
[72] R. Banfield, L. Hall, K. Bowyer, and W. Kegelmeyer, "A comparison of decision tree ensemble creation techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 173–180, Jan. 2007.
[73] P. Utgoff, N. Berkman, and J. Clouse, "Decision tree induction based on efficient tree restructuring," *Mach. Learn.*, vol. 29, pp. 5–44, Oct 1997.
[74] M. Box, D. Davies, and W. Swann, Nonlinear optimisation Techniques. Oliver and Boyd, 1969.
[75] R. Schainker, G. Zhang, P. Hirsch, and C. Jing, "Online dynamic stability analysis using distributed computing," in *Power and Energy Society General Meeting, 2008 IEEE*, July 2008, pp. 1–7.
[76] S. Chakrabarti and E. Kyriakides, "Optimal placement of phasor measurement units for power system observability," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1433–1440, Aug. 2008.
[77] Powertech Labs, "DSATools: Dynamic Security Assessment Software," http://www.dsatools.com.
[78] M. He, V. Vittal, and J. Zhang, "Online dynamic security assessment with missing pmu measurements: A data mining approach," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1969–1977, 2013.
[79] Alberta Electric System Operator Rules, "Section 502.9: synchrophasor measurement unit technical requirements," Aug. 2012, [Available] online: http://www.aeso.ca/downloads/2012-08-30 Section 502-9 phasor.pdf.
[80] Schweitzer Engineering Laboratories Technical Report, "Improving the availability of synchrophasor data," Aug. 2011, [Available] online: https://www.selinc.com/TheSynchrophasorReport.aspx?id=98004
[81] R. Emami and A. Abur, "Robust measurement design by placing synchronized phasor measurements on network branches," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 38–43, Feb. 2010.

[82] Gomez-Exposito, A. Abur, P. Rousseaux, A. de la Villa Jaen, and C. Gomez-Quiles, "On the use of PMUs in power system state estimation," in *Proc. 17th Power Systems Computation Conference,* Stockholm, Sweden, Aug. 2011.
[83] Y. Wang, W. Li, and J. Lu, "Reliability analysis of wide-area measurement system," *IEEE Trans. Power Del.*, vol. 25, no. 3, pp. 1483–1491, July 2010.
[84] R. Bryll, R. Gutierrez-osuna, and F. K. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, vol. 36, no. 6, pp. 1291–1302, June 2003.
[85] T. K. Ho, "Random decision forests," in *Proc. Third Intl Conf. Document Analysis and Recognition*, Montreal, Canada, Aug. 1995, pp. 278–282.
[86] Phadke and J. Thorp, Synchronized Phasor Measurements and Their Applications. New York: Springer, 2008.
[87] T. K. Ho, "Random decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.
[88] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[89] V. Khiabani, O. P. Yadav, and R. Kavesseri, "Reliability-based placement of phasor measurement units in power systems," *J. Risk and Reliability,* vol. 226, no. 1, pp. 109–117, Feb. 2012.
[90] L. Hyafil and R. L. Rivest, "Constructing optimal binary decision trees is NP-complete," *Information Processing Letters,* vol. 5, no. 1, pp. 15–17, May 1976.
[91] F. Aminifar, S. Bagheri-Shouraki, M. Fotuhi-Firuzabad, and M. Shahidehpour, "Reliability modeling of PMUs using fuzzy sets," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2384–2391, Oct. 2010.

A1. Appendix 1: Regression Tree Growing and Splitting

Suppose a knowledge base K consisting of N sample cases (x_1, y_1), (x_2, y_2), …, (x_N, y_N) is used to construct a RT.

Using the *Least Squares Regression*:

$$R(d) = \frac{1}{N}\sum_n \left(y_n - d(x_n)\right)^2$$

The value of $y(t)$ that minimizes $R(d)$ is the average of y_n for all cases (x_n, y_n) falling into node t, that is:

$$\bar{y}(t) = \frac{1}{N(t)}\sum_{x_n \in t} y_n$$

Given the set of candidate splits S, for any $s \in S$ that splits node t into t_L and t_R, let

$$\Delta R(s, t) = R(t) - R(t_L) - R(t_R)$$

The best split s^* of node t is that split in S which decreases $R(t)$ the most:

$$\Delta R(s^*, t) = \max_{s \in S} \Delta R(s, t)$$

A RT with T_{max} nodes is built by iteratively splitting nodes so as to maximize the decrease in $R(T)$. Splitting stops when for every $t \in T_{max}$, $N(t) \le N_{min}$. $N(t)$ is the number of samples falling into node t and N_{min} is a pre-defined threshold.

A1.1. RT Pruning and Testing

For any subtree $T \le T_{max}$, let us define its complexity as $\tilde{T}$, the number of terminal nodes in T. Then its cost-complexity measure $R_\alpha(T)$ is:

$$R_\alpha(T) = R(T) + \alpha\left|\tilde{T}\right|$$

where $\alpha \ge 0$ and is called the complexity penalty.

For each value of α, find the subtree $T(\alpha) \le T_{max}$ such that the cost-complexity $R_\alpha(T)$ is minimized:

$$R_\alpha(T(\alpha)) = \min_{T \le T_{\max}} R_\alpha(T)$$

The result is a decreasing sequence of pruned trees, with an increasing sequence of α values:

$$T_1 > T_2 > T_3 > \ldots > \{t_1\}$$

$$0 = \alpha_1 < \alpha_2 < \alpha_3 < \ldots$$

where $T_1 \le T_{max}$, t_1 is the tree contains the root node only.

To select the right sized tree from the sequence $\{T_1, T_2, \ldots\}$, a proportion of N is randomly selected and used as test samples TS. The cost of subtree T_k is:

$$R^{TS}(T_k) = \frac{1}{N_2} \sum_{(x_n, y_n) \in TS} \left(y_n - d_k(x_n)\right)^2$$

Another test method is the *V-fold cross-validation* (*CV*). Dividing N in V subsets $\{N_1, N_1, \ldots, N_V\}$, let:

$$R^{CV}(T_k) = \frac{1}{N} \sum_{V=1}^{V} \sum_{(x_n, y_n) \in N_V} \left(y_n - d_k(x_n)\right)^2$$

The relative error $RE^{CV}(T_k)$ of subtree T_k is given by:

$$RE^{CV}(T_k) = R^{CV}(T_k) / R(\bar{y})$$

A1.2. Selection of the Best Pruned Tree

The *Standard Error* (SE) estimate is used to select the best pruned subtree commensurate with accuracy.

Take the cross-validation testing for example, the subtree with T_k nodes is selected as the best pruned tree if:

$$R^{CV}(T_k) \le R^{CV}(T_{k0}) + SE$$

where

$$R^{CV}(T_{k0}) = \min_k R^{CV}(T_k)$$