AN APPROACH TO ASSESS THE RESILIENCY OF ELECTRIC POWER GRIDS

By

NAVIN KODANGE SHENOY

Bachelor of Engineering in Electrical Engineering
University of Mumbai
Mumbai, India
2006

Master of Science in Electrical Engineering
Oklahoma State University
Stillwater, Oklahoma
2010

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 2015

AN APPROACH TO ASSESS THE RESILIENCY OF ELECTRIC POWER GRIDS

Dissertation Approved:

Dr. R. Ramakumar

Dissertation Adviser

Dr. Qi Cheng

Dr. Nishantha Ekneligoda

Dr. Girish Chowdhary

.

ACKNOWLEDGEMENTS

I sincerely thank my advisor Dr. Ramakumar for giving me the opportunity to work on this research topic. His guidance, support and encouragement helped me in all times of my graduate studies, research and writing of this dissertation. I feel honored to have worked with him and he has taught me more than I could ever give him credit for.

I would like to thank Dr. Qi Cheng, Dr. Nishantha Ekneligoda and Dr. Girish Chowdhary for agreeing to serve on my committee and for extending their full support whenever I needed it.

I gratefully acknowledge the financial assistance provided by the Engineering Energy Laboratory and the PSO/Albrecht Naeter Professorship in the School of Electrical And Computer Engineering, Oklahoma State University in the form of research assistantship.

Finally, I dedicate this work to my family and friends with all love and gratitude and offer my sincere regards to all of those who supported me in any respect during the completion of this research.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: NAVIN KODANGE SHENOY

Date of Degree: DECEMBER, 2015

Title of Study: AN APPROACH TO ASSESS THE RESILIENCY OF ELECTRIC POWER GRIDS

Major Field: ELECTRICAL ENGINEERING

Abstract: Today’s electric power grids face serious challenges due to overstressed networks, need to assimilate variable generation, strict environmental regulations and widespread weather-caused outages. There is an urgent need to improve grid resilience and system security more than ever before. Amidst such challenges, the best approach would be to focus primarily on the grid intelligence rather than implementing redundant preventive measures. The foundation of any intelligent operational strategy would be on the ability of the grid to assess its current dynamic state instantaneously. Traditional forms of real-time power system security assessment consist mainly of methods based on power flow analyses. Such methods do not consider the dynamics inherent in the system and hence, are static in nature. However, in order to capture the nonlinear dynamics present in the system, it is necessary to carry out time-domain simulations (TDS) that are computationally too involved to be performed in real-time. Machine learning (ML) techniques have the capability to organize data gathered from such simulations and thereby extract useful information in order to better assess the system security instantaneously. This dissertation presents a framework that would enable implementation of machine learning techniques for real-time assessment of grid resilience. An IEEE 14-bus test system is used in this work for simulation purposes. Firstly, a set of multiple steady-state operating points is generated by performing a SSA on the base case of the power system. Secondly, a TDS is performed on each operating point to assess the grid resilience against a specific disturbance, thus generating a database for this work. This work highlights the importance and need for selecting a few operating points as “landmarks” in the operational space under consideration for prediction of power system security. Further, a few heuristics are developed so as to rank all the operating points of the system. The proposed ranking methodologies are used to select the best landmarks in order to improve prediction accuracy on the original database, thereby enhancing the ability to assess grid resilience instantaneously.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

In the wake of new vulnerabilities such as those arising from severe weather events and cyber-attacks, current electric grids can no longer be allowed to operate as they did in the past. It is becoming increasingly difficult to analyze different combinations of contingencies under changing scenarios. Grid resilience and improved situational awareness will form the basis of future electric grids in order to tackle these new challenges.

A resilient grid would have the ability to adapt to changing conditions and recover rapidly from incidents such as natural disasters, deliberate attacks or any other types of system failures. The most cost effective way to meet such stringent requirements is through intelligent operation of the grid by employing data driven models that are both reactive and predictive in nature. The key attribute involved here is the ability to assess the current state of the power system in real-time in terms of its security. Power system security is defined as its ability to survive imminent disturbances (contingencies) without interruption of customer service. A more secure grid would be more resilient to all kinds of potential disruptions. Historically, it has been recognized that for a power system to be secure, it must be stable against all types of disturbances [1,2]. Stability refers to the ability of a power system, for a given initial operating condition, to maintain intact operation and equilibrium following a physical disturbance. Hence, stability analysis is an integral component that can facilitate the assessment of power system security and thus, its resiliency.

Security in terms of operational requirements implies that following a sudden disturbance, power system would be secure if and only if:

1) it could survive the transient swings and reach an acceptable steady state condition, and

2) there are no limit violations in the new steady state condition.

The first requirement can be met by carrying out time-domain simulations in order to investigate the instability phenomena such as loss of synchronism or voltage collapse in the post-contingency transient phase. The second requirement is met by using power-flow based methods in order to assess the new steady state condition for voltage and current limit violations.

Time-domain simulations (TDS) are computationally involved and too complex to be performed in real-time. Therefore, for many years in the past, the electric utility industry's framework for real-time security assessment mainly consisted of solution methods that would meet only the second requirement stated earlier. Such a type of real-time security analysis is prevalent even today and is commonly referred to as "Static Security Assessment (SSA)". On the other hand, a "Dynamic Security Assessment (DSA)" procedure would strive to meet both the requirements (as stated earlier) in real-time in order to assess power system security.

Different forms of DSA practices have existed in North America since the late 1980s [3]. Modern DSA implementations are able to complete a computation cycle within 5-20 minutes after a real-time snapshot (base case) of the system is available [4]. Real-time snapshots are provided by existing SCADA-based state estimators every few seconds or minutes depending on the size of the system [5-9]. Thus, these modern DSA implementations can be termed as "near real-time" and not "real-time". However, the latest PMU-based data collection technology can provide much better snapshots wherein the measurements are transmitted to the main control center at rates as fast as 60 samples/second [10]. Thus, DSA implementations of the future will be required to handle large amounts of data and complete the computation cycles much faster in order to assess the system security in true “real-time”. Mathematically, such an instantaneous assessment would be possible only if grid resilience against any contingency can be expressed as a function of the state estimator output. In other words, input to the data-driven models must consist of only steady-state (static) quantities namely bus voltages and bus angles derived from power-flow based methods.

Moreover, with consumers demanding better power quality and reliability, our systems are required to be more adaptive and secure than ever before. In order to come to terms with new challenges, nations around the world are developing increased interest in modernization of their current electric grids. The ongoing efforts taken in this direction has led to the emergence of “Smart Grid” paradigm throughout the world. The main objective of the smart grid concept is to provide safe, secure and reliable power to today’s modern digital societies. Some of its features include the following:

- Self-healing (ability to quickly detect, respond and restore the system from perturbations)
- Grid and consumer interaction (to provide an interface in order to incorporate consumer behavior and equipment in to the current grid)
- Two-way power flow
- To facilitate the integration of new and intermittent renewable sources of energy and storage options
- Tolerant of terrorist and cyber attacks
- To optimize asset management and operational efficiency
- Use of advanced communications, internet and IT solutions for real-time monitoring
- Optimize grid operations in order to reduce costs of maintenance and system planning

Again, the foundation of such a concept would lay on the ability of the grid to assess its current operating state instantaneously. Today, the most promising approach to achieve such instantaneous assessment for large power systems is to harness the power of data-driven models through application of machine learning techniques.

Machine learning (ML) techniques have the ability to assimilate and reason with knowledge the way human brain does. Such techniques are primarily driven by data that could be in the form of various power system parameters such as [11-13]: voltage, current, power, frequency, power angles etc. ML techniques can capture the nonlinear dynamics of power systems by extracting useful information from such data. DSA tools employing such ML techniques will have the ability to determine stability limits in real-time. Such sophisticated tools will be able to analyze the current and future dynamics of power systems without carrying out extensive time-domain simulations. Additionally, these tools would also benefit the system operators by providing them with real-time information on trends in system security, thereby facilitating faster decision-making during crucial times. Also, as the entry of renewable energy systems further increases grid complexity, it is possible to extend the proposed work in order to accommodate online training, thereby resulting in a smart tool that can very effectively assess the system security in real-time.

Based on power systems analysis fundamentals, this dissertation develops a framework that would enable implementation of machine learning techniques for real-time assessment of grid resilience. An IEEE 14-bus test system is used in this work for simulation purposes. Firstly, a set of multiple steady-state operating points is generated by performing a SSA on the base case of the power system. Secondly, a TDS is performed on each operating point to assess the grid resilience against a specific disturbance, thus generating a database for this work. This work highlights the importance and need for selecting a few operating points as “landmarks” in the operational space under consideration for prediction of power system security. Further, a few heuristics are developed so as to rank all the operating points of the system. The proposed ranking methodologies are used to select the best landmarks in order to improve prediction accuracy on the original database, thereby enhancing the ability to assess grid resilience instantaneously.

CHAPTER II

STATIC SECURITY ASSESSMENT (SSA)

2.1 SSA Algorithm

Static security assessment (SSA) provides a mathematical framework to compute stability limits for individual buses and lines based on power flow based methods. Such an assessment forms an integral part of the framework mentioned in this report and is conducted before carrying out time-domain simulations. SSA involves checking for voltage stability criteria at each bus, i.e. if steady-state voltages at all buses are within limits (in per-unit). Power-voltage (PV) curves are plotted for each bus by systematically loading the base case of the power system under consideration. This is achieved by means of an algorithm called as “Continuation Power Flow (CPF)” [14].

CPF is a “case worsening” procedure where the power system is loaded in steps as follows:

$$\begin{aligned} P_G &= \lambda\, P_{G0} \\ P_L &= \lambda P_{L0} \\ Q_L &= \lambda Q_{L0} \end{aligned} \tag{1}$$

where P_{G0}, P_{L0}, Q_{L0} are the base case generator and load powers (in per-unit) and λ is the loading parameter (in per-unit). CPF facilitates plotting of voltage curves as a function of loading parameter λ, for each bus.

2.2 A Set of Operating Points

As stated earlier, such a framework can be used to generate a dataset consisting of multiple steady-state operating points. For an *n*-bus system, every such operating point can be represented by a feature vector *x* of dimension *2n* consisting of *n* bus voltages and *n* bus angles as features. A set *S* containing such objects is given by,

$$\{x \in \Re^{2n} : x = [V_1,......V_n, \delta_1,......\delta_n]^T\} \tag{2}$$

where V_i's are bus voltages (in per unit) and δ_i's are bus angles (in degrees).

Such a set of *S* can be generated by systematically loading the base case of the power system under consideration. In order to cover the broader spectrum of power system operation, additional scenarios in the form of transmission line outages can be considered, thus adding more number of objects of the form *x* to the set *S* as given in equation (2).

A CPF routine can be performed for each line outage *i* of the power system along with the base case (*i*=0). Thus, a maximum loading parameter λ_{maxi} can be calculated for each *i*, taking voltage stability criteria into account. For SSA, voltage stability criteria can be stated in the form of maximum and minimum values of permissible voltage levels at all buses in the system. The set *S* represented by equation (2) is generated only for values of λ given by,

$$1 \le \lambda \le \lambda_{\max i}, \forall i \tag{3}$$

In equation (3), the loading at λ=1 represents the base case loading of the power system under consideration and hence, this signifies that set *S* is generated by successively loading the base case until there is violation of the voltage stability criteria at any of the buses in the system. It has to be noted that these λ_{maxi} values account for only steady-state voltage violations and hence, do not provide any information about dynamic system security. In order to account for dynamic stability, time-domain simulations are performed for each operating point, as described in the next chapter. **Figure 1** shows flowchart for implementing static security assessment (SSA) on any power system in order to generate a set *S* containing objects of the form *x* with each representing a steady-state operating point of the system.

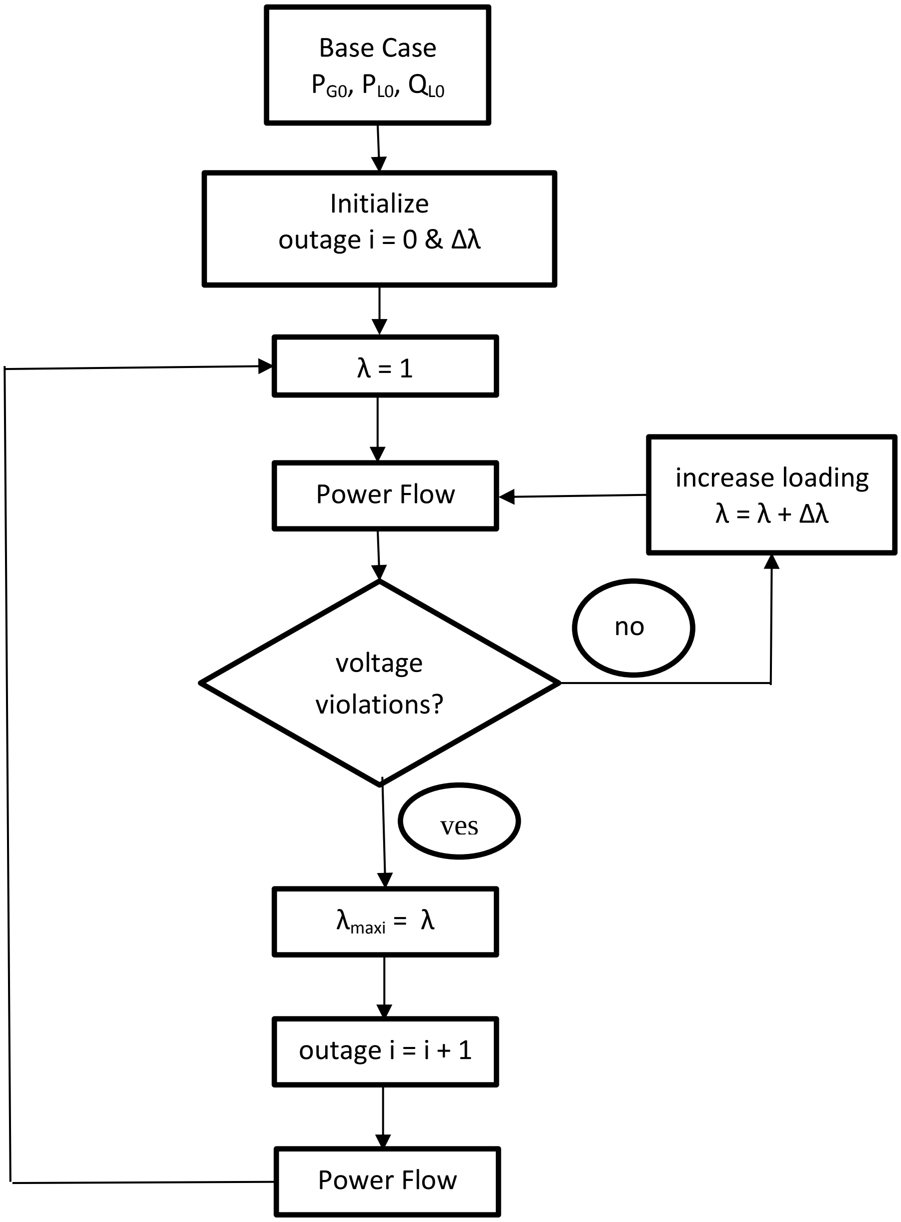


Figure 1: Flowchart for SSA implementation

2.3 Test System: IEEE 14-bus system

The IEEE 14-bus test system as shown in **Figure 2** is used for simulation purposes [15]. It has 14 buses, 5 generators and 11 loads. Generators are represented by machine models along with automatic voltage regulators and turbine governors. The voltage stability criteria for each bus is considered as follows,

Maximum permissible voltage: V_{max} = 1.2 per-unit

Minimum permissible voltage: V_{min} = 0.8 per-unit

PSAT toolbox for Matlab is used to carry out all the power system routines on the test system [16].

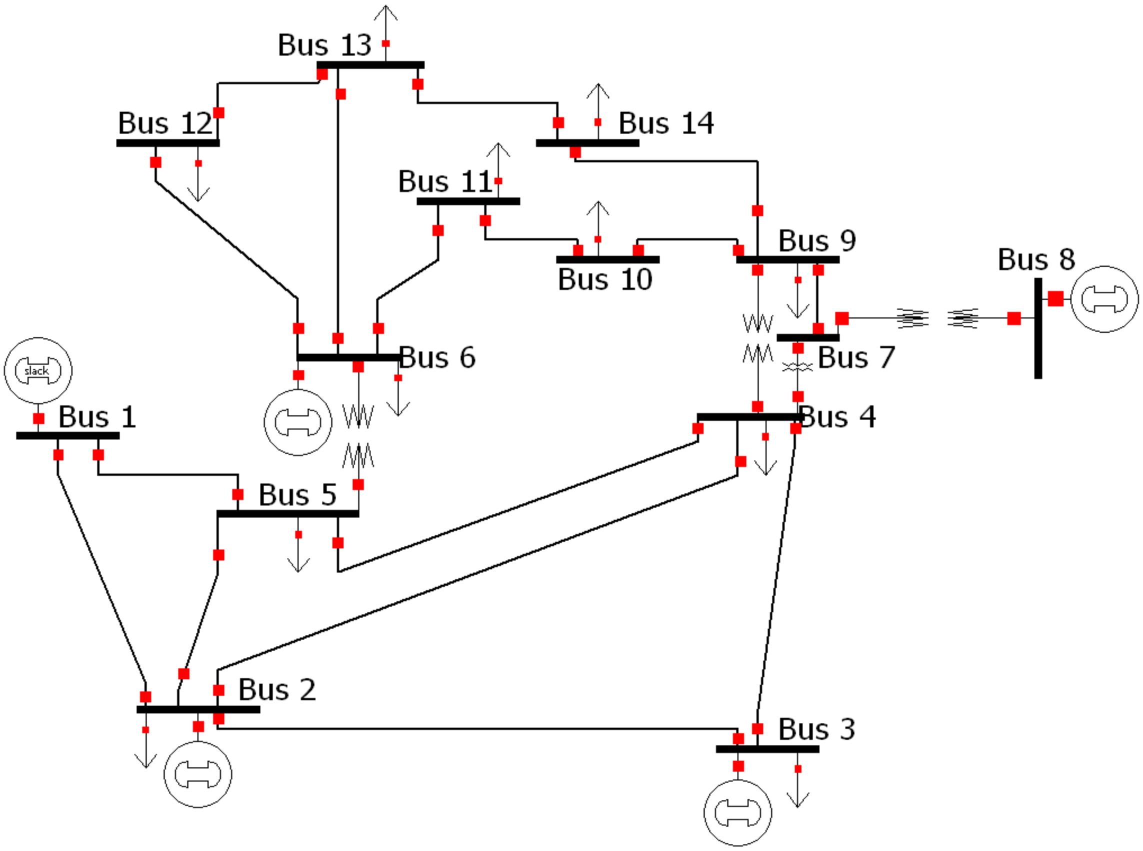


Figure 2: IEEE 14-bus Test System

2.4 SSA on the Test System

SSA is performed on the IEEE 14-bus test system for the voltage stability criteria as stated earlier: V_{max} = 1.2pu and V_{min} = 0.8pu at each bus. The CPF routine from the PSAT toolbox is simulated for each line outage *i* of the test system along with the base case (*i*=0). For each scenario *i*, the CPF routine generates a power flow solution at every steady-state operating point of that scenario, starting from zero loading (λ=0) until there is violation of the voltage stability criteria at any of the buses in the system, thus generating the value for λ_{maxi}. **Figures 3a and 3b** show V-λ curves for the base case and a particular line outage respectively. **Table 1** shows the values of maximum loading parameter λ_{maxi} for each *i*, taking voltage stability criteria into account.

Now, a set *S* as represented by equation (2) is generated for only those values of λ as given by equation (3). For the 14-bus test system, each object *x* in the set *S* is a 28-dimensional vector consisting of 14 bus voltages and 14 bus angles as features. Such an object *x* can be represented as follows,

$$x = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ \vdots \\ V_{14} \\ \delta_1 \\ \delta_2 \\ \vdots \\ \vdots \\ \delta_{14} \end{bmatrix} \tag{4}$$

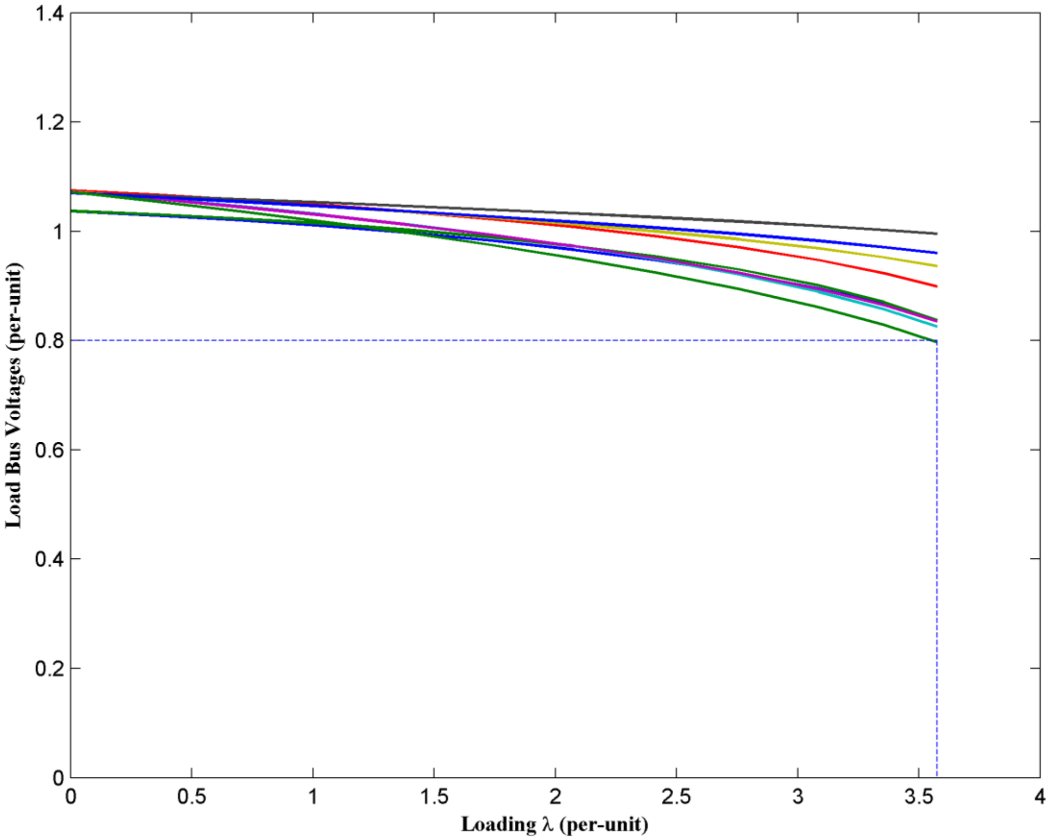


Figure 3a: V-λ curves of PQ buses for the base case, bus 14 voltage reaches V_{min} at λ = 3.5748

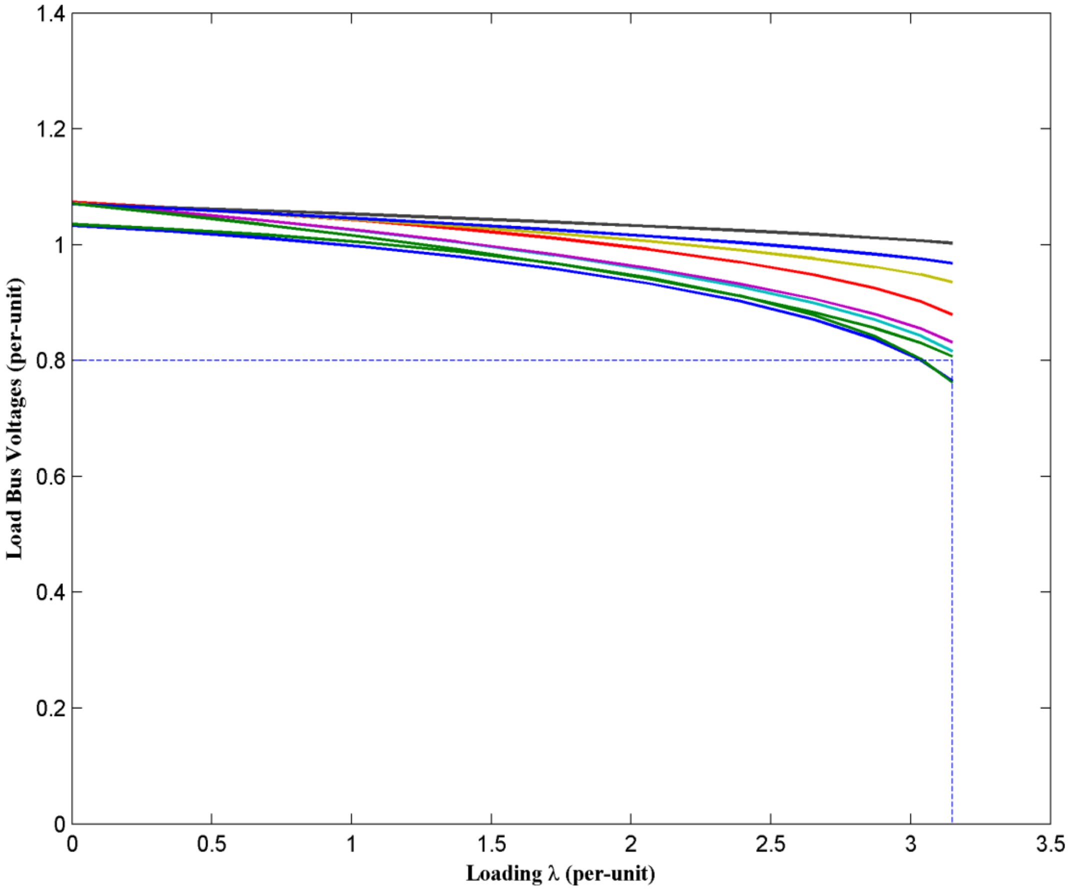


Figure 3b: V-λ curves of PQ buses for outage of line#16 (bus 2 to bus 4), bus 5 voltage reaches V_{min} at λ = 3.1487

Table 1: Maximum Loading Parameters λ_{max} for base case and line outages

Line outage	**λ_{maxi} (per-unit)**
Base case	3.5748
1	3.2587
2	3.5436
3	3.5715
4	2.7225
5	2.7464
6	3.0640
7	3.5626
8	3.0592
9	2.7461
10	2.3900
11	1.3306
12	2.2194
13	3.5660
14	3.3899
15	3.4707
16	3.1487
17	3.5150
18	2.2428
19	3.3218
20	3.0248

CHAPTER III

DYNAMIC SECURITY ASSESSMENT (DSA)

3.1 A Survey on DSA of Power Systems

As mentioned earlier, traditional implementations of security assessment rely only on steady-state methods whereas a DSA implementation would assess both: steady-state and dynamic security of the power system. The following subsections present a discussion on basic DSA architecture and its various other aspects such as: typical DSA workflow, solution methods, security indices and examples of DSA implementations around the world.

3.1.1 DSA Architecture

DSA architecture consists of various modules such as measurements, state estimator, computation, outputs and controls. Measurements from the traditional SCADA systems or the latest PMU-based systems form the main DSA input. The state estimator makes use of the complete system model along with external equivalents in order to generate system snapshots. These snapshots are presented to the computation module that carries out all the computations required for security assessment. The computation module consists of solution methods that evaluate both the steady-state and dynamic performance of the power system under consideration. Outputs are provided in the form of various displays and visualizations in the control room for the attention of the operators. Outputs are also integrated with remedial action schemes thereby invoking closed-loop automatic controls. **Figure 4** shows the block diagram of DSA architecture.

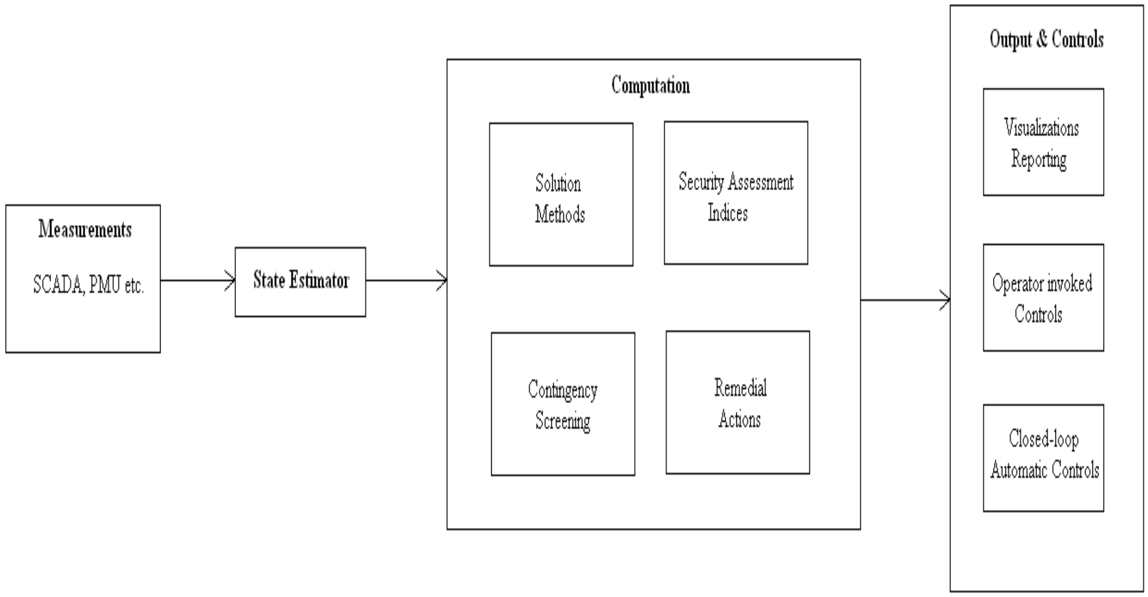


Figure 4: DSA Block Diagram

3.1.2 Typical DSA Workflow

The snapshot presented to the computation module forms the starting point of any typical DSA workflow process as follows,

- Apply the list of contingencies/scenarios to the current snapshot
- Evaluate steady-state and dynamic security assessment using available solution methods
- Case worsening: Systematically load the given snapshot by using an algorithm such as continuation power flow in order to generate more snapshots
- Repeat analyses for all individual snapshots
- Advanced forms of DSA may even predict future snapshots and repeat the above procedure for each predicted snapshot
- Perform post-disturbance event analysis for each contingency
- Generate stability indices to evaluate different contingencies/scenarios and to evaluate the security status of the snapshot under consideration

- These indices can be converted into various forms of displays and visualizations.

3.1.3 DSA Solution Methods

The basic power system solution methods are as follows,

- Steady-State Power flow: This involves solving the basic algebraic power flow equations of the system. It includes employing steady state limits on power flows, voltages and currents. Power flow analysis can be performed for one (N-1) or more (N-k) outages of power system components.
- Transient Stability: This involves solving the differential algebraic equations following a large disturbance such as faults on transmission system, sudden trips of generators, loads etc. Such an analysis involves performing time-domain simulations that generate dynamic responses of all the system state variables.
- Small-Signal Stability: This involves study of system oscillations following small disturbances such as small variations in generation and loads. Such an analysis checks if all the system modes are sufficiently damped by using the system state matrix derived after linearizing the differential algebraic equations of the system.
- Frequency Stability: This involves monitoring the ability of the system to maintain its frequencies within safe limits following any disturbance. Such an analysis can be facilitated by performing time-domain simulations.

Reference [2] provides a brief summary on solution methods available in the power systems domain and also includes extensions to the basic solution methods mentioned above. In addition to this, [17] describes Single Machine Equivalent (SIME) method that transforms multi-machine system into a suitable one-machine infinite bus equivalent and [18] gives description about the direct Transient Energy Function (TEF) method.

Different types of security applications can be deployed by using such solution methods in order to improve situational awareness in the power grid. For example, a DSA can be implemented to handle any of the following problems,

- Abnormal angles
- Abnormal voltages
- Dynamic oscillations and damping
- Line overloads
- Loss of synchronism
- Voltage collapse
- Frequency rise/decline
- Post-disturbance violations

3.1.4 Indices for DSA

The computation module of DSA may carry out measurement based analyses as well as the ones that are simulation based or may use a hybrid approach combining the benefits of both. The output results from DSA must be properly presented to the operators through displays and visualizations in the control room in order to highlight the security status of the power system. These displays may range from simple P-V curves or time-plots of state variables to security indices designed specifically to understand the phenomena of interest. Such indices measure the security level of the current operating point and provide deeper insight into the steady-state and dynamic violations that occur in the system. Moreover, security indices help in reducing excessive amount of information generated by any of the solution methods or DSA applications, so as to avoid information overload to the operators. As mentioned earlier, these indices would clearly identify system security concerns related to thermal overloading, steady-state stability, voltage stability, transient stability and frequency stability.

References [19,20] define indices that represent overload on every branch and voltage violations at every bus in the system. The motivation and justification for using composite indices is presented in [21,22]. References [23-27] propose different types of indices that are based on deviations of rotor angles, frequencies, voltages and power flows. Risk-based indices that depend on probability of occurrence and impact of the contingency are defined in [28]. References [29,30] state indices that are based on coherency of generator angles and variations of transient energy during the contingency. Indices for transient classifiers and filters that employ transient energy function method are mentioned in [31-33]. The fluctuation analysis technique defined in [34] uses a frequency-based index that is calculated for every window of samples from PMU measurements. Reference [35] proposes an index that is based on primary frequency response of the system and [36] proposes an index for assessment of voltage unbalance. Reference [37] provides an exhaustive list of dynamic security indices.

3.1.5 DSA Implementations around the World

DSA practices have been followed in North America since the late 1980s [38]. The DSA tool implemented at PJM completes a transient stability assessment cycle within 15 minutes by applying 3000 contingencies on the current system snapshot provided by its EMS [39]. Southern Company Services (USA) has implemented a transient security assessment tool that roughly processes 250 contingencies in about 30 minutes [40]. In this case, the real-time snapshots are created by Southern's EMS on an hourly basis. ERCOT's DSA implementation consists of Voltage Security Assessment Tool (VSAT) and a Transient Security Assessment Tool (TSAT) [41]. The wide-area monitoring system at ISO New England consists of System Disturbance Management (SDM) module and Oscillatory Stability Management (OSM) module [42]. SDM identifies, locates and characterizes significant events in the system whereas OSM characterizes electromechanical oscillations.

BC Hydro's online transient stability tool not only assesses the system dynamic performance but also determines the margin of stability and its sensitivity to key variables of the system [43]. Reference [44] presents six security analysis methods that are available in the DSA implementation of Brazilian system operator – ONS. Sensitivity analysis is used in the ONS DSA in order to rank various voltage and power flow controls in the system. The DSA module presented in [45] has been tested on several provincial power grids in China. Its main features are contingency screening and preventive control algorithms based on TEF methods. Hydro-Quebec's DSA tool provides dynamic security limits every 5 seconds [46]. Reference [47] elaborates the methodology adopted by BC Transmission Corporation to determine 2-dimensional security margins for its northern and eastern generation areas through DSA implementation. The European PEGASE Project is aimed at implementing an online DSA for the Pan European Transmission Network through advancements in state estimation, optimization algorithms, time-domain simulations and power system component modeling [48]. The DSA implementation at National Control Centre in Ireland is designed for handling high amounts of wind generation in the system [49]. It calculates secure wind levels and ramping requirements for real-time operating points. All these DSA examples simply highlight the importance of such implementations in ensuring optimal security conditions on a continual basis for any electric power grid.

3.2 Database Generation

The goal of a DSA is to classify different cases based on their dynamic security severity. Dynamic security depends on the time responses of various system variables for the contingency under consideration. As mentioned earlier, it is not possible to perform computationally intensive time-domain simulations in real-time. Nonetheless, machine learning techniques have the ability to extract information from offline time-domain simulations. Subsequently, such useful information can be used to predict dynamic system security for new configurations in order to avoid lengthy time-domain simulations. To implement such an application, detailed time-domain simulations are required to be conducted for different operating points. Thus, a database, on which ML techniques can operate, needs to be generated in offline mode.

The database is generated in the form of a feature matrix X and an output vector y. Each row of the feature matrix X represents a steady state operating point in the form of object $x \in \mathfrak{R}^{2\mathrm{n}}$ from set *S* as defined in equation (2). Matrix X contains total number of '*m*' such objects and hence, its size is (*m x 2n*). A time-domain simulation for a specific contingency is performed on each of these *m* objects. These simulations are tagged as 'stable' or 'unstable' depending on the time responses of system variables. As mentioned earlier, different stability indices can be generated from such time-domain simulations which typically represent instability phenomenon such as loss of synchronism, voltage collapse or frequency excursions. Power system quantities like dynamic voltages, frequencies and generator rotor angles provide access to such phenomenon. A few examples of stability indices are as follows: maximum angle separations between generators, duration of frequency deviation, time to instability and transient voltage violations.

Output vector y is a binary column vector with *m* rows wherein each row represents whether the corresponding TDS is stable(1) or unstable(0). As shown in **Figure 5**, each row of matrix X represents a steady state operating point in the form of object *x* and each column represents a feature of that object. As highlighted in equation (2), the *n* bus voltages and *n* bus angles form the features for any *n*-bus system.

Essentially, DSA is a mapping between each object *x* and its resiliency against the contingency under consideration, expressed by function *f* such that,

$$f(x) = \begin{cases} 1, & if \;\; TDS \;\; is \;\; stable \\ 0, & if \; TDS \;\; is \;\; unstable \end{cases} \tag{5}$$

The next chapter describes the application of machine learning techniques in order to arrive at this unknown function *f*.

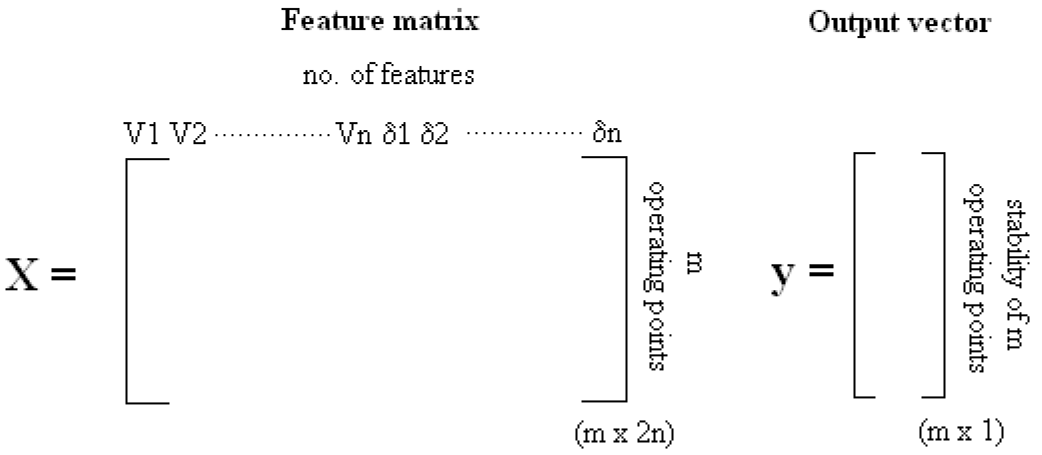


Figure 5: Matrix X and vector y

3.3 DSA on the Test System

For the IEEE 14-bus test system considered in this report, a feature matrix X is formed consisting of objects *x* from set *S* as generated in the previous chapter. Each row of matrix X is a 28-dimensional vector, as given by equation (4), representing 14 bus voltages and 14 bus angles as features of the test system. A TDS for a load disturbance of 0.2 per-unit (increase) is performed for every steady state operating point as represented by each row of matrix X. Stability is decided based on the average values of voltage violations over the entire simulation period (V_{max} = 1.2pu and V_{min} = 0.8pu). Accordingly, an output vector y is formed wherein a stable TDS is represented by a bit ‘1’ and an unstable TDS by bit ‘0’. **Figure 6** shows the flowchart for DSA implementation.

Figures 7a, 7b and 7c show the dynamics of generator frequencies, relative generator angles and bus voltages for a stable case. **Figures 8a, 8b and 8c** show the dynamics of generator frequencies, relative generator angles and bus voltages for an unstable case. From **figure 7a, 7b, 8a and 8b**, it can be observed that generators do not loose synchronism for both stable and unstable cases. However, the voltage stability criteria is being violated for the unstable case (**figure 8c**) when averaged over the entire simulation period. Thus, it can be concluded that voltage violations occur before the loss of grid synchronism for the 14-bus test system under consideration. Similar behavior is observed when a TDS (as specified earlier) is performed on any of the steady-state operating points contained in matrix X. Therefore, for the purpose of this work, stability of each operating point is decided based on the violation of the voltage stability criteria.

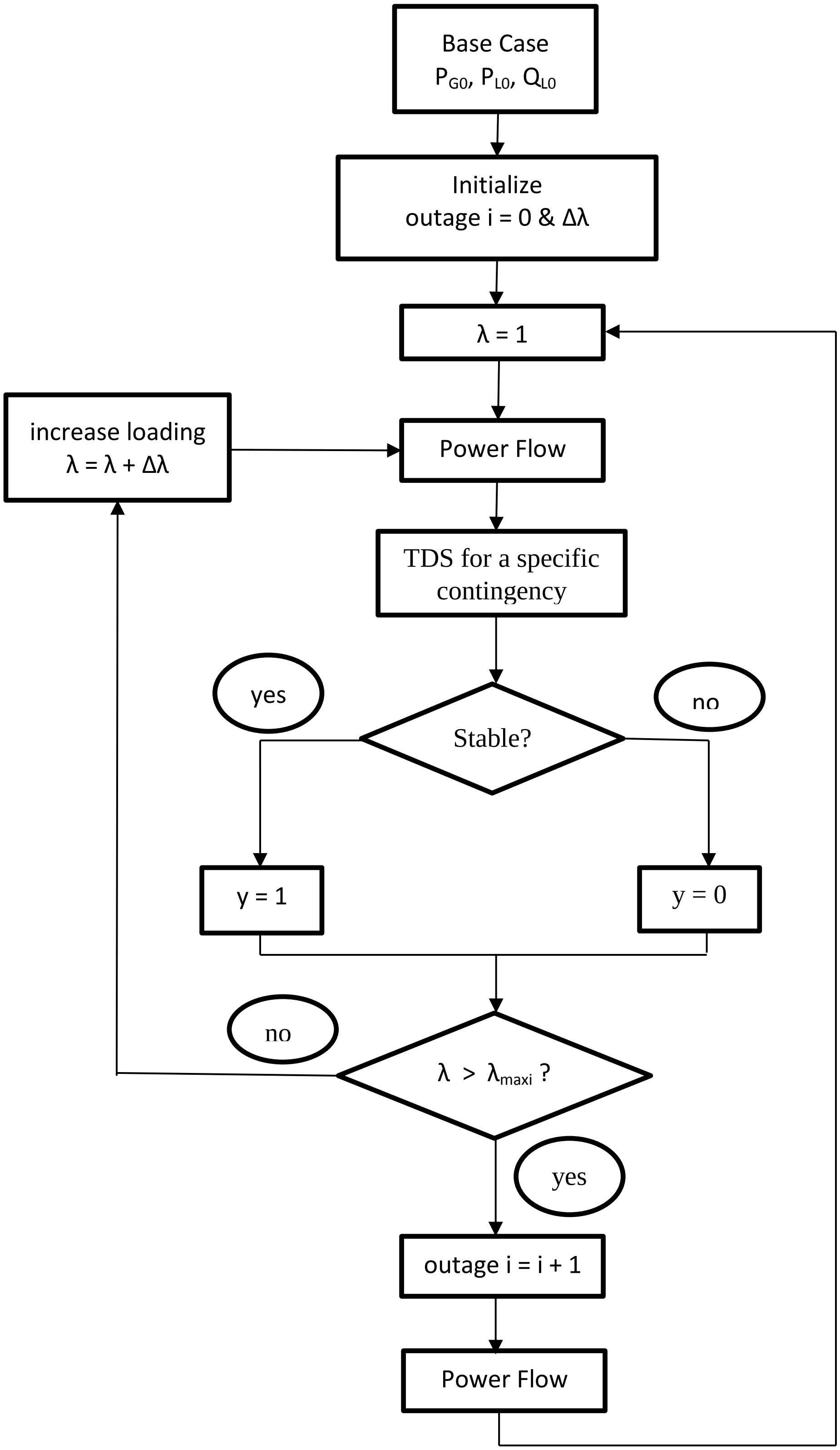
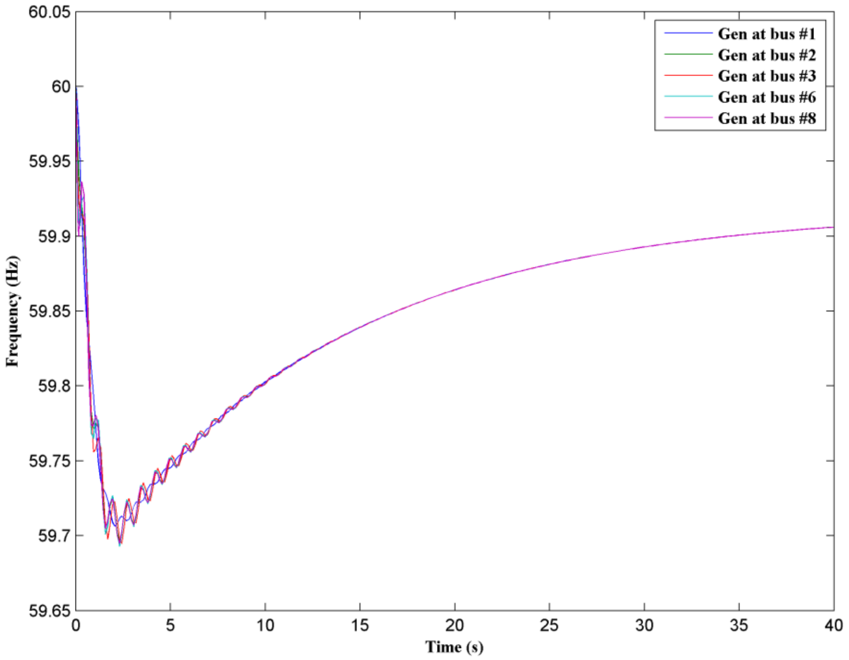
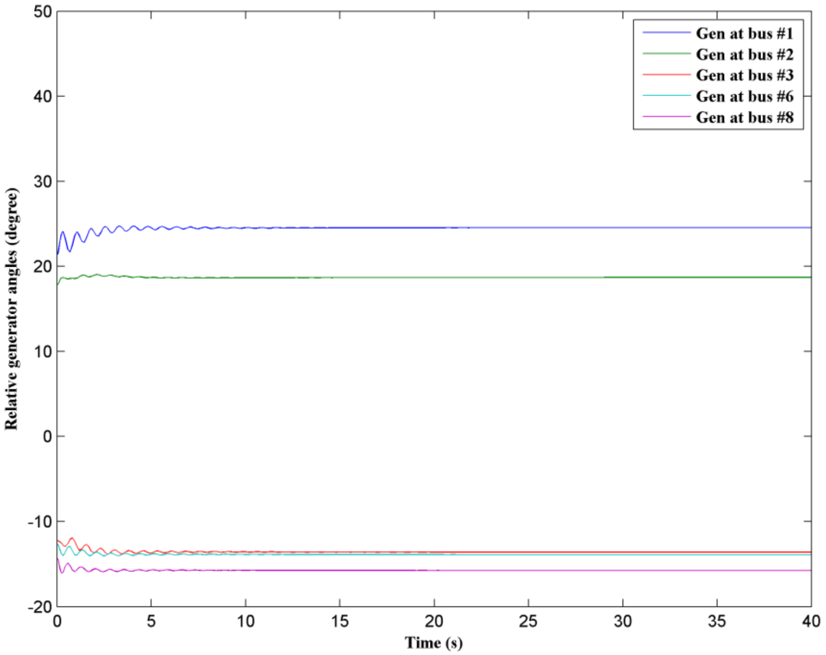


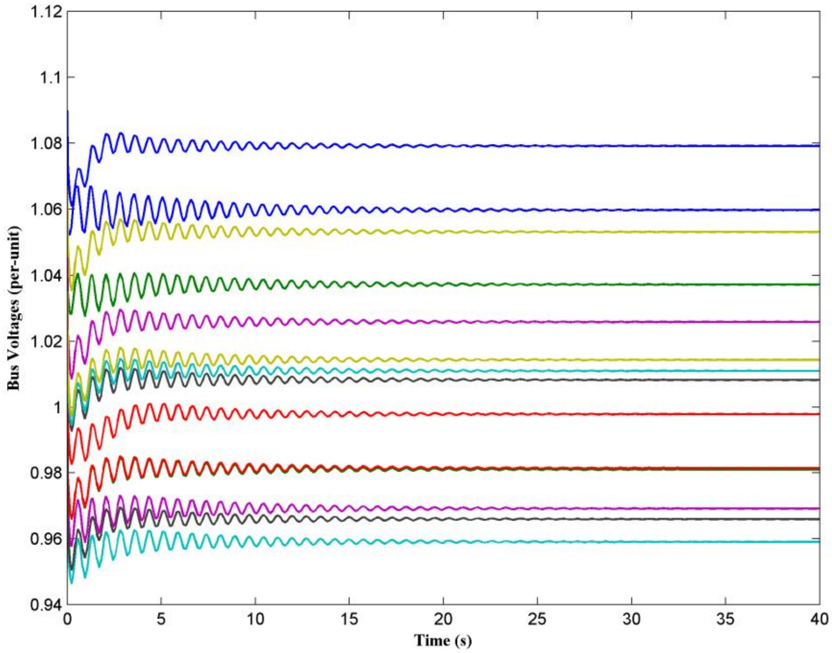
Figure 6: Flowchart for DSA implementation



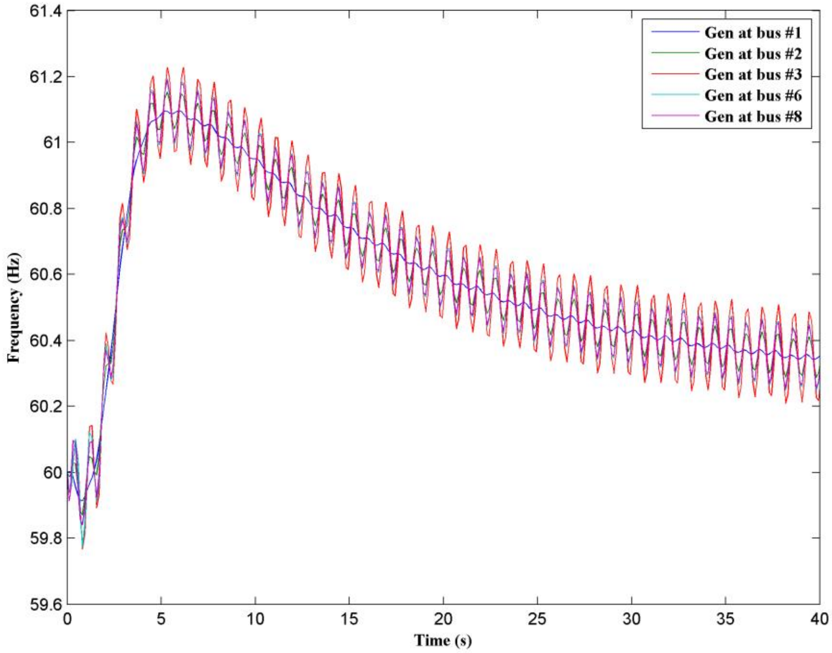
Figures 7a: Generator frequencies (stable case)



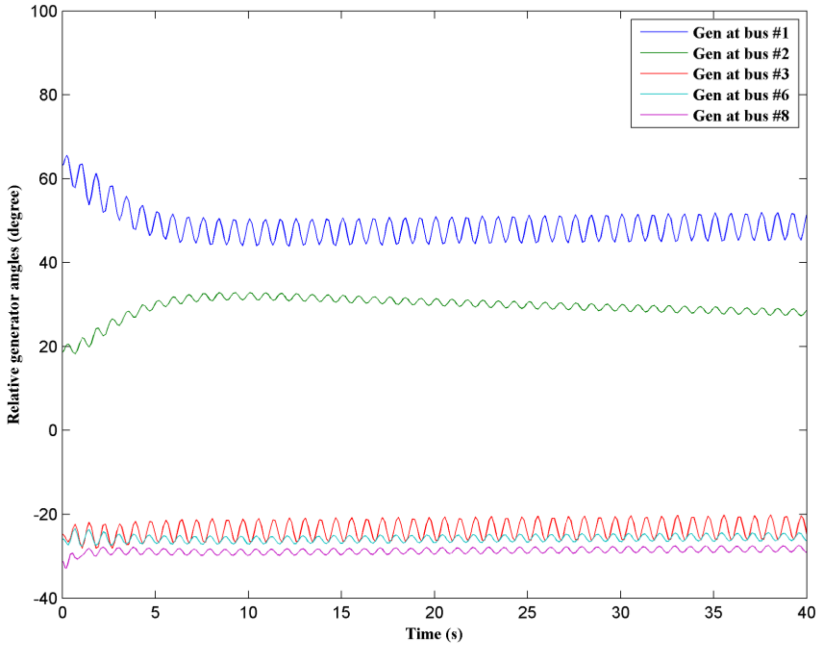
Figures 7b: Relative Generator Angles (stable case)



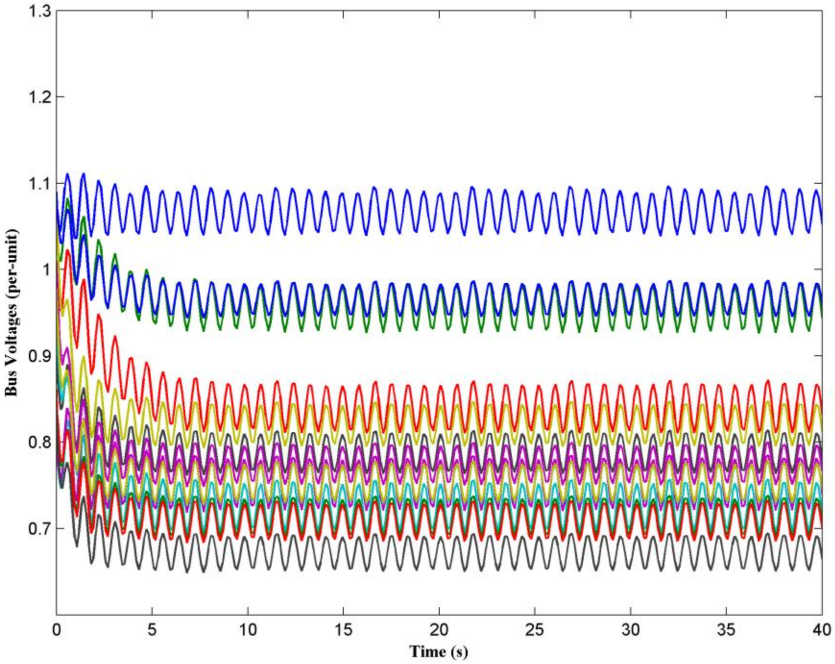
Figures 7c: Bus Voltages (stable case)



Figures 8a: Generator frequencies (unstable case)



Figures 8b: Relative Generator Angles (unstable case)



Figures 8c: Bus Voltages (unstable case)

CHAPTER IV

APPLICATION OF MACHINE LEARNING TECHNIQUES

4.1 Introduction to Machine Learning (ML)

A database forms an essential part of any machine learning (ML) framework, as shown in **Figure 9**. Generally, database contains pre-processed data rather than raw data, as a collection of objects. An object is described by a certain number of features or attributes providing some information. In this work, database is formed in the form of matrix X and vector y as described earlier. The output information for each object in matrix X is contained in vector y. Depending on the output information given by any ML algorithm, a problem can be of two types as follows [50]:

- Classification problem: to predict discrete valued output
- Regression problem: to predict continuous valued output

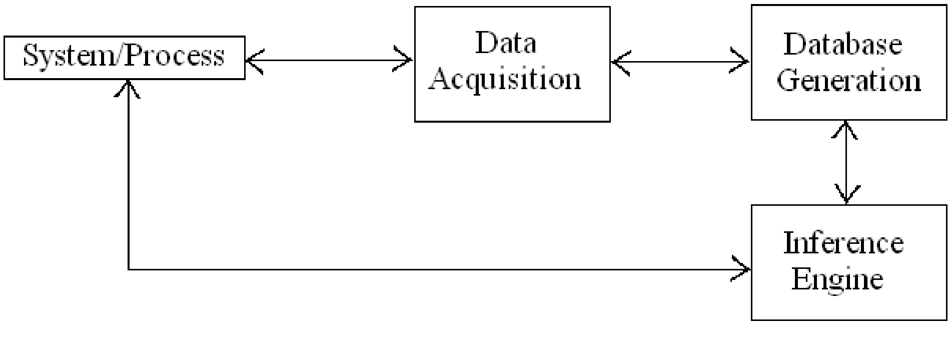


Figure 9: Machine Learning Framework

As shown in **Figure 10**, a classification problem involves determining the class of each object into 2 or more classes by learning the decision boundary. In this figure, each object is a 2-dimensional vector with x_1 and x_2 as its features; and output y of each object represents the class to which it belongs (either class ‘1’ or class ‘0’ in this figure). Further, the class of any new object appearing in the future time can be found out instantaneously once the algorithm learns the decision boundary from the training data.

In a regression problem, the algorithm predicts a continuous numerical value for the object under consideration. This is illustrated in **Figure 11** which shows a 1-dimensional object along x-axis and its output along y-axis. In this figure, each object is described by only one feature x_1 and as in case of classification problem, the output of any new object appearing in the future time can be easily found once the algorithm fits the training data to the curve as shown.

Further, MI techniques can be employed to learn the input-output relation in two ways, namely, supervised and unsupervised learning. In supervised learning, algorithm is trained using a training set of data consisting of known input-output value pairs. The algorithm then models the input-output relationship which can be used to explain the observed pairs or to predict the output value for any new unseen input. In other words, algorithm maps the relationship between the matrix X and vector y. On the other hand, unsupervised learning methods are not oriented towards any prediction task. Rather, these methods try to cluster similar objects automatically without any output information or they analyze similarity among attributes that describe an object. Hence, for unsupervised methods, the database consists of only matrix X and not vector y.

For the purpose of work presented in this report and as mentioned in Chapter III, the stability or output of each object in matrix X is tagged as either stable ‘1’ or unstable ‘0’ with respect to the TDS of a specific disturbance or contingency. Such a problem can be categorized as a “classification” problem for the purpose of implementing the ML framework.

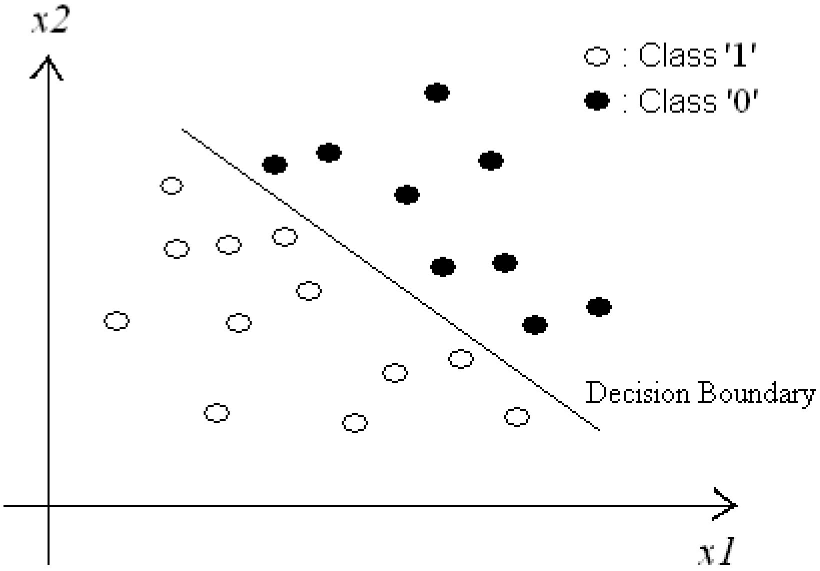


Figure 10: Classification Example for Machine Learning

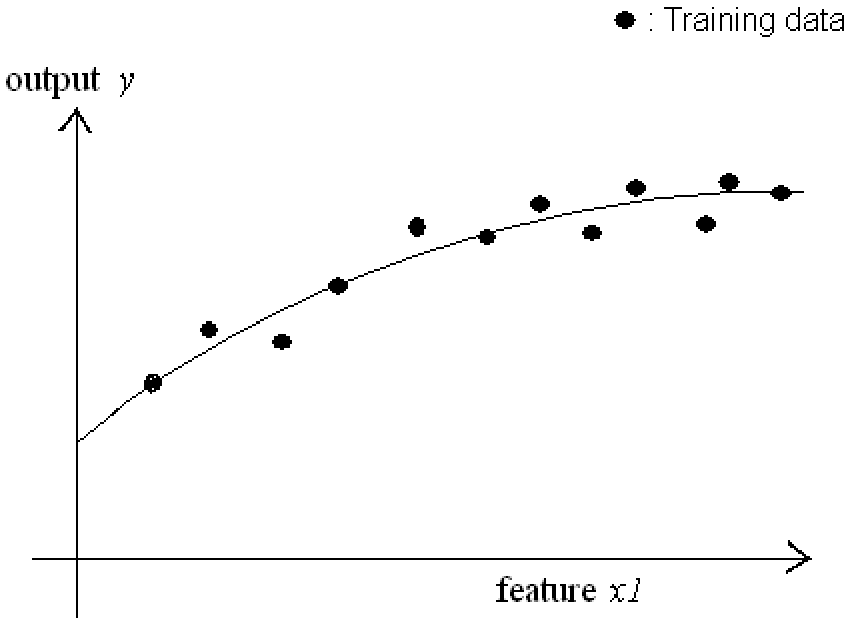


Figure 11: Regression Example for Machine Learning

4.2 Application of ML Techniques to DSA

The first stage of any machine application is to generate a sufficient database that covers a wide range of operating scenarios of the power system under consideration. As mentioned earlier, database is generated in the form of feature matrix ‘X’ and output vector ‘y’. Each row of matrix ‘X’ is a collection of variables that represents an operating point of the power system. This can be a solved power flow case or an output from the state estimator. And each row of output vector ‘y’ represents the security index in form of real-valued scalar or a categorical variable. The process of database generation is critical to the success of ML implementation in order to analyze the dynamic phenomena of interest. This section provides a brief survey on application of ML techniques for DSA of power systems.

Reference [51] explores the application of neural networks (NN) to screen and rank contingencies based on a dynamic index generated from variant of TEF method. A statistical pattern recognition approach for classification of transient simulations in to stable and unstable cases is presented in [52] whereas [53] presents a probabilistic approach to transient stability using Monte Carlo simulations. References [54,55] employ Decision Trees (DT) based approach for security assessment and preventive control. Hierarchical clustering is used in [56] to group generators that show similar rotor angle behavior for a large number of contingencies. This work can be applied to label the training data required for online prediction of power system dynamic signature. Reference [57] makes comparison of Kernel Regression Trees with DTs and NNs implementations of DSA. A Case-Based Reasoning (CBR) methodology for assessment of real-time dynamic security is presented in [58]. A much more computationally efficient NN algorithm called Extreme Learning Machine (ELM) is described in [59-61]. Reference [62] presents an application of various ML methods for detection of cyber-attacks on the power system. A fuzzy-logic based classification method is developed in [63] in order to predict the security index of any given operating point of the power system. Further, a range of supervised ML methods like NNs, Support Vector Machines (SVM), DTs, AdaBoost and unsupervised methods like self-organizing maps are described in [64-71].

As can be seen from the above literature survey, the starting point for most DSA implementations is the current system snapshot as provided by the state estimator. The next step involves carrying out simulations for many contingency scenarios by employing different solution methods, thereby generating a database. However, as mentioned earlier, the latest PMU-based data collection technology can provide much better snapshots and the actual states of the power system to the main control center in real-time. Thus, DSA implementations of the future will be required to handle such large amounts of data and complete the computation cycles much faster in order to assess the system security in "actual" real-time. Under these circumstances, sparsity techniques that enable reduction of data burden and information overload will gain importance. The ML framework presented in this chapter is an attempt along similar lines.

4.3 Implementation of ML Framework

Machine learning techniques can be applied to the database as generated in the previous section in the form of feature matrix X (size *m x 2n*) and output vector y (size *m x 1*). Each row *i* of matrix X is in the form of object $x \in \mathfrak{R}^{2n}$ from set *S* as defined in equation (2) and is referred to as the i^th^ training example: $x^{(i)}$. Similarly, the i^th^ row from vector y represents the output of the i^th^ training example and is represented by a bit $y^{(i)}$ (either 0 or 1). Therefore, we have,

$x^{(i)}$ = i^th^ training example

$y^{(i)}$ = output (stability) of the i^th^ training example (either 0 or 1)

For '*2n*' features and '*m*' training examples, the mathematical representation of matrix X and vector y can be now given as follows,

$$X = \begin{bmatrix} ----x^{(1)^T} ---- \\ ----x^{(2)^T} ---- \\ \vdots \\ \vdots \\ ----x^{(m)^T} ---- \end{bmatrix}_{mx2n} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ \vdots \\ y^{(m)} \end{bmatrix}_{mx1} \tag{6}$$

For a *n*-bus power system, training example $x^{(i)}$ can be represented by *2n*-dimensional object *x* as defined in equation (2). The more general form of object *x* for any *2n* features can be given as follows,

$$x = [x_1, x_2, ..., x_{2n}]^T \tag{7}$$

Next, a prediction/hypothesis function *h* in terms of parameter vector θ (column vector) of size *2n* is proposed as follows,

$$h_\theta(x) = g(\theta^T x) \tag{8}$$

where *x* is any training example vector as given in equation (7) and *g* depends on the machine learning algorithm being employed.

Parameter vector θ is given as follows,

$$\theta = [\theta_1, \theta_2, ..., \theta_{2n}]^T \tag{9}$$

This leads to,

$$\theta^T x \ = \ \theta_1 x_1 + \theta_2 x_2 + ... + \theta_{2n} x_{2n} \tag{10}$$

In ML applications, bias terms x_0 (=1) and θ_0 are often added to the object vector *x* (equation 7) and parameter vector θ (equation 9) respectively. Hence, a term θ_0 x_0 gets added to the RHS of equation (10).

The cost function *J* for machine learning algorithms is generally of the form [72],

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]^2 \tag{11}$$

The above cost function is the mean of the sum of squared errors in predicting the outputs of m training examples. Such a cost function can be minimized by using analytical method or batch gradient descent method. The batch gradient descent method updates any randomly initialized vector θ in an iterative manner as follows,

for *j* = 0,2,………, 2n

iterate {

$$\begin{gathered} \theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \\ where, \\ \frac{\partial}{\partial \theta_j} J(\theta) = \sum_{i=1}^{m}[(h_\theta(x^{(i)}) - y^{(i)})^2 \;\; x_j{}^{(i)}] \end{gathered} \tag{12}$$

}

The optimal parameter vector θ thus derived can be used for predicting the stability of future cases in real-time. In this work, a robust MATLAB routine “fminunc” [73] is used for solving the unconstrained minimization problem represented by equations (11) and (12).

The problem presented in this report is to classify a TDS as stable(1) or unstable(0). For such classification problems, logistic regression can be used, in which case functions *g* and *J* are given as follows [74],

$$g(z) = \frac{1}{1+e^{-z}} \tag{13}$$

and

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))] \tag{14}$$

The function *g(z)* given in equation (13) is a sigmoid function and its value lies between 0 and 1 as shown in **Figure 12**. For classification purposes, TDS cases for which *g(z)* is greater than 0.5 can be considered as stable and the rest as unstable. At this point, it should be noted that function *h* given in equation (8) approximates the unknown function *f* of equation (5) given in the previous chapter, when the parameter θ is optimal.

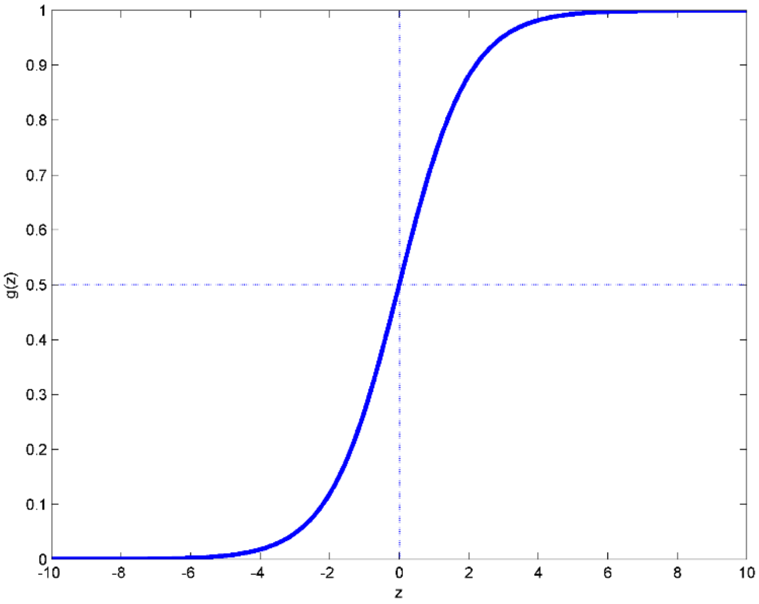


Figure 12: Sigmoid function *g*

The approximated function f_{apprx} can be given by,

$$f_{apprx}(x) = \begin{cases} 1, & if \ \ h_{\theta}(x) \geq 0.5 \\ 0, & if \ \ h_{\theta}(x) < 0.5 \end{cases} \tag{15}$$

In order to test the algorithm, the 14-bus dataset represented by matrix X and vector y (as generated in the previous section) can be divided into a training set (75%) and a test set (25%), which is a normal practice in ML domain. We may also delete the constant feature columns from matrix X such as those containing PV bus voltages and reference angles, since such constant feature values do not add any valuable information. Therefore, an original matrix X with 22 columns (features) is used in this report for training the ML algorithms. **Figures 13a and 13b** show the learning curves for the training and test sets respectively. Learning curves are plotted by varying the number of examples *m* in the training set. As highlighted in these figures, the average prediction errors on the training and test sets are calculated as 1.487 % and 2.674 % respectively. For *m* objects, prediction error is the percentage of examples that are classified incorrectly by the function f_{apprx} given in equation (15) and it is calculated as follows,

$$\% \ Error = \frac{100}{m} \sum_{i=1}^{m} err(f_{apprx}(x^{(i)}), y^{(i)})$$

$$where$$

$$err(f_{apprx}(x^{(i)}), y^{(i)}) = \begin{cases} 1, \ \ if \ \ f_{apprx}(x^{(i)}) = 1 \,\&\, y^{(i)} = 0 \quad or \\ \qquad if \ \ f_{apprx}(x^{(i)}) = 0 \,\&\, y^{(i)} = 1 \\ 0, \ \ otherwise \end{cases} \tag{16}$$

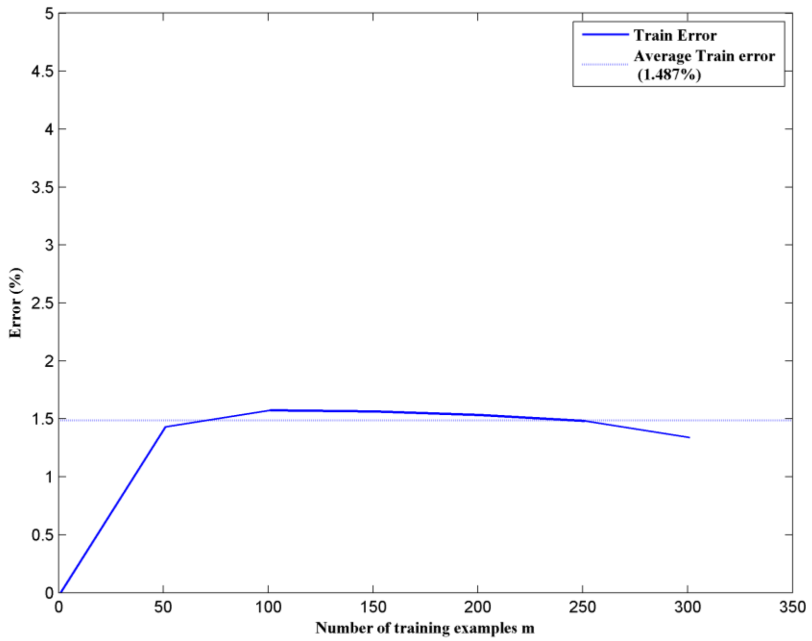


Figure 13a: Learning Curve (Training set)

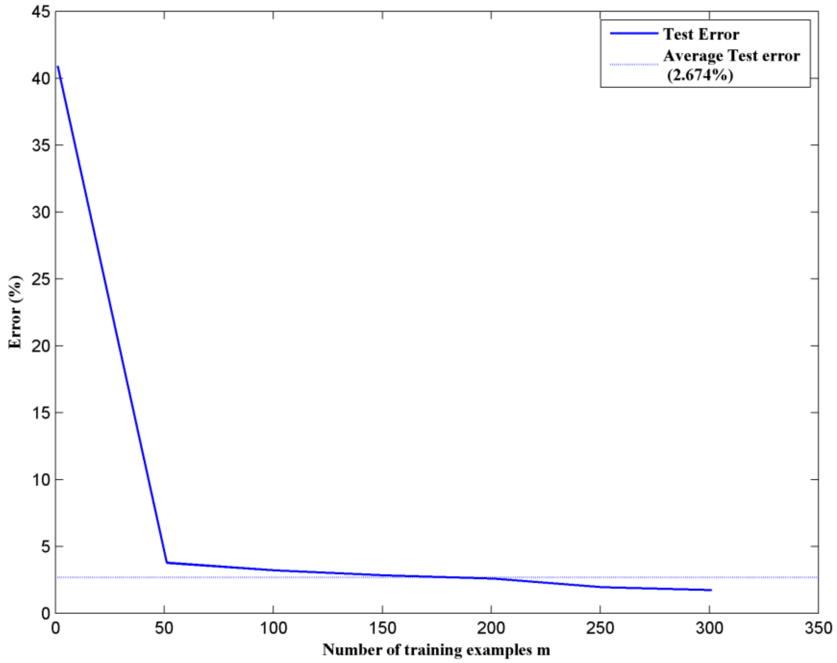


Figure 13b: Learning Curve (Test set)

Apart from logistic regression, this work also employs an unsupervised ML technique called k-means clustering [75]. The k-means algorithm forms clusters of objects in the given dataset as shown in **Figure 14**. In this figure, each object is a 2-dimensional vector with x_1 and x_2 as its features. With respect to the ML framework as described in earlier sections, k-means algorithm operates on just the feature matrix X to generate the clusters and the output vector y is not required. Hence, this algorithm is categorized as an unsupervised ML technique.

Essentially, given a dataset containing m′ objects in d-dimensional space, $\mathfrak{R}^d$ and an integer K, the k-means algorithm generates K centroids in $\mathfrak{R}^d$, so as to minimize the mean squared distance of each object from its nearest centroid. After random initialization of K centroids, such a minimization can be achieved by iterating over the following 2 steps:

- Cluster assignment step:

  In this step, each of the m′ objects is assigned the cluster that is nearest to it.

- Move centroid step:

  In this step, each centroid is moved to a new position given by the mean of all the objects assigned to the cluster represented by that centroid.

The k-means algorithm is used in the later part of this report in order to generate the “landmarks” in the operational space under consideration from the database of the form given by equation (6). The next section of this chapter introduces the concept of “landmarks” and “linear kernel”. Further, this report presents a strategy to select best landmarks in order to improve the prediction accuracy.

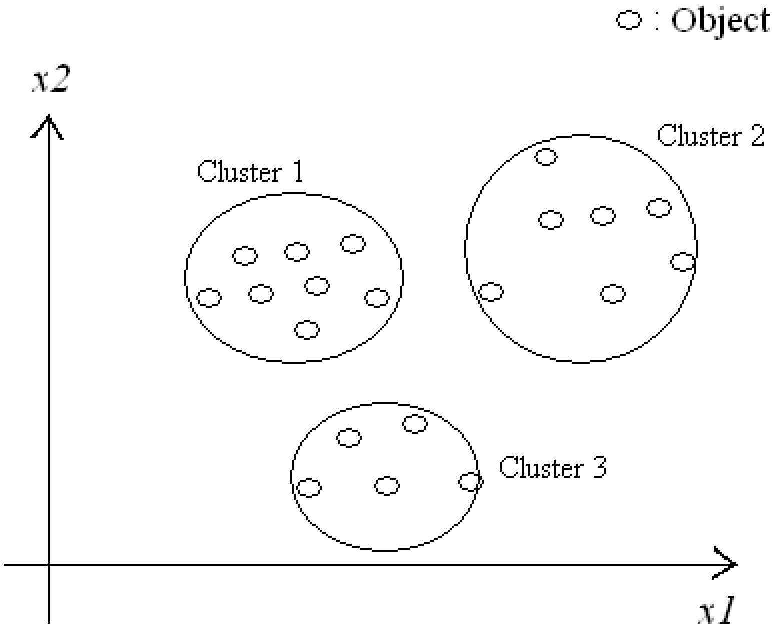


Figure 14: k-means clustering

4.4 Landmarks and Linear Kernel

The concept of selecting “landmarks” gains importance from the fact that a few training examples may contain the most relevant information about the inherent dynamics present in the database [76]. This section investigates the possibility of selecting such landmarks within the operational space under consideration in order to improve prediction accuracy without compromising computational efficiency. Essentially, these landmarks are *2n*-dimensional objects belonging to the same set *S* given by equation (2). Such a landmark *l* can be specified as follows,

$$l = [l_1, l_2, \ldots\ldots\ldots\ldots\ldots\ldots, l_{2n}]^T \tag{17}$$

In order to demonstrate the effectiveness of this concept, *L* number of landmarks are drawn at random from the rows of matrix X and then, every (training example, landmark) pair is compared using a linear kernel [77]. A linear kernel measures the similarity between any training example *x* (as specified in equation (7)) and landmark *l* using the dot product and is given by,

$$sim(x,l) = x^T l \ = x_1 l_1 + x_2 l_2 + \dots\dots\dots\dots\dots\dots . + x_{2n} l_{2n} \tag{18}$$

For *m* training examples in feature matrix X and *L* number of landmarks, similarity is calculated between each training example $x^{(i)}$ and landmark point $l^{(j)}$ such that $1\leq i \leq m$ and $1\leq j\leq L$. As a result, the original feature matrix X (size *m x 2n*) gets transformed into a new matrix X'$_{\text{random}}$ (size *m x L*) which can be now used for training and testing purposes. If a matrix X$_{\text{LMs1}}$ contains all the *L* landmarks of the form as specified in equation (17) along its rows, then such a matrix of size (*L x 2n*)is given by,

$$X_{LMs1} = \begin{bmatrix} ----l^{(1)^T} ---- \\ ----l^{(2)^T} ---- \\ \vdots \\ \vdots \\ ----l^{(L)^T} ---- \end{bmatrix}_{Lx2n} \tag{19}$$

For matrix X (size *m x 2n*) as specified in equation (6) and matrix X$_{\text{LMs1}}$ (size *L x 2n*), a new matrix X'$_{\text{random}}$ (size *m x L*), as mentioned earlier, can be obtained as follows,

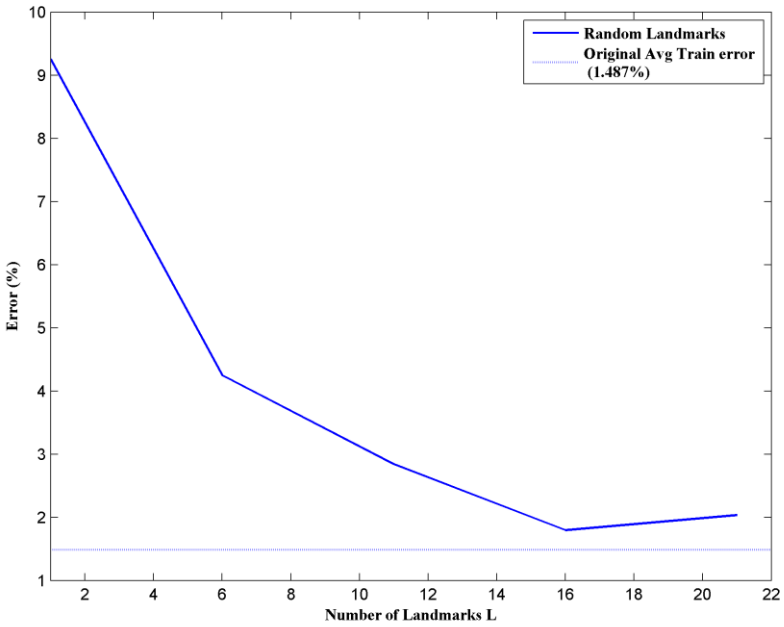
$$X^{'}_{random} = XX^{T}_{LMs1} \tag{20}$$

Computational efficiency is maintained by enforcing the following constraint,

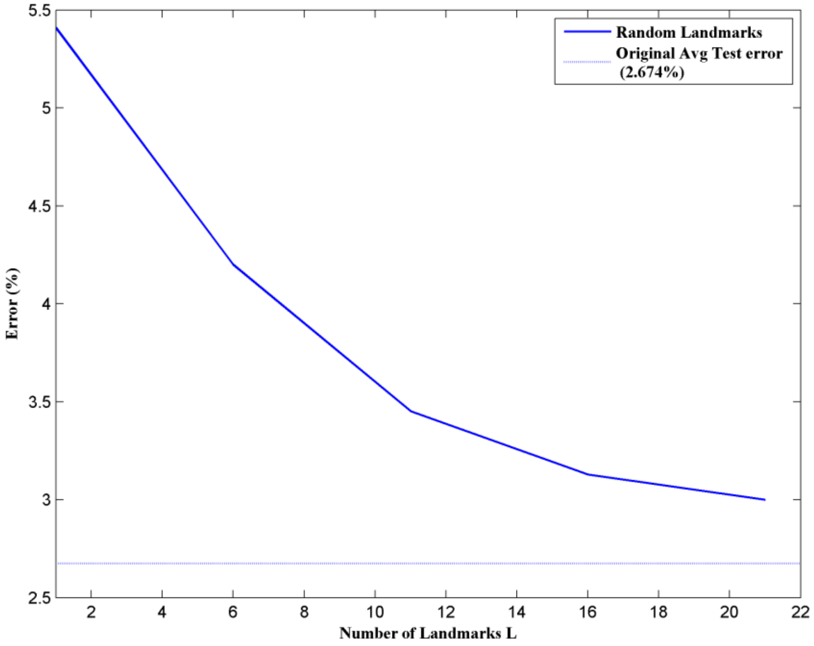
$$L \leq 2n \tag{21}$$

The above constraint states that the number of columns in the new matrix X'$_{\text{random}}$ should not be greater than that in the original feature matrix X. This ensures that the size of matrix X'$_{\text{random}}$ (in terms of computer memory requirements) is not greater than that of matrix X, thus maintaining the computational efficiency.

As shown in **Figures 15a and 15b**, prediction errors on the training and test sets decrease as the number of landmarks *L* increases. However, it should be noted that such a random selection of landmarks does not guarantee better performance when compared with the average training and test set errors calculated in the previous section as seen in these figures.



Figures 15a: %Error vs number of landmarks (training set)



Figures 15b: %Error vs number of landmarks (test set)

4.5 Selection of Landmarks using K-means Algorithm

Choosing the most appropriate set of "landmarks" for a given dataset is not an easy task. In this section, k-means algorithm is used to derive better landmarks as compared to the random ones selected in the previous section. Using k-means algorithm as described in Section 4.3 of this chapter, centroids can be calculated for any feature matrix X. A total number of *L* such centroids are generated from original matrix X for use as landmark points and then, using linear kernel a new matrix $X'_{centroids}$ is formed like in the previous section. If these L centroids are contained in the matrix X_{LMs2} with each of the form as specified in equation (17), then the matrix X_{LMs2} will be similar to the matrix X_{LMs1} as defined in equation (19). Matrix $X'_{centroids}$ can be now given as follows,

$$X'_{centroids} = XX^{T}_{LMs2} \tag{22}$$

In an attempt to find the better landmarks, the original matrix X is divided into 2 matrices X_{stable} and $X_{unstable}$ consisting of only stable and unstable cases respectively. If matrix X contains m_1 stable cases and m_0 unstable cases such that $m_1 + m_0 = m$, then the size of matrix X_{stable} is (m_1 *x* $2n$) and that of matrix $X_{unstable}$ is (m_0 *x* $2n$). Using k-means, a total number of *L* centroids are generated for each of these two matrices separately and again using linear kernel, two new matrices X'_{stable} and $X'_{unstable}$ are formed. If matrix X_{LMs3} contains *L* centroids generated from matrix X_{stable} and matrix X_{LMs4} contains *L* centroids generated from matrix $X_{unstable}$, then both matrices X_{LMs3} and X_{LMs4} will be of the size (*L x 2n*) as defined in equation (19). Matrices X'_{stable} and $X'_{unstable}$ can be now derived as follows,

$$\begin{aligned} X'_{stable} &= XX^{T}_{LMs3} \\ X'_{unstable} &= XX^{T}_{LMs4} \end{aligned} \tag{23}$$

Note that all of the matrices X’$_{\text{random}}$, X’$_{\text{centroids}}$, X’$_{\text{stable}}$ and X’$_{\text{unstable}}$ have the same size: (*m x L*). Again, while selecting total number of landmark points *L*, the constraint given by equation (21) should be enforced so as maintain the computational efficiency.

The strategy for selecting landmarks using k-means can be summarized as follows,

- Select *L* random examples from original matrix X as landmarks and generate X’$_{\text{random}}$
- Select *L* centroids from original matrix X as landmarks and generate X’$_{\text{centroids}}$
- Select *L* centroids from X$_{\text{stable}}$ as landmarks and generate X’$_{\text{stable}}$
- Select *L* centroids from X$_{\text{unstable}}$ as landmarks and generate X’$_{\text{unstable}}$
- Plot learning curves using X’$_{\text{random}}$, X’$_{\text{centroids}}$, X’$_{\text{stable}}$, X’$_{\text{unstable}}$
- Compare the training and test set errors and select the best *L* landmarks

Figure 16 shows the flowchart representation for implementing the above strategy.

For simulation purposes, the original matrix X, as generated from IEEE 14-bus test system in Section 2 of this chapter, is used to form the matrices X’$_{\text{random}}$, X’$_{\text{centroids}}$, X’$_{\text{stable}}$ and X’$_{\text{unstable}}$. **Figures 17a and 17b** show learning curves for the training and test sets generated from each of the these matrices with *L*=22 landmarks. From these figures we can conclude that centroids selected from the unstable cases are the best landmarks for this dataset. Moreover, it has to be noted that computational efficiency is not compromised since the total number of landmarks used here (*L*=22) is not greater than the total number of columns in the original matrix X (=22). **Figures 18a and 18b** again compare the learning with increasing number of landmarks for the case of random landmarks against landmarks selected as centroids from only unstable cases. **Figures 19a and 19b** plot the learning curves for the original matrix X (without any landmarks) and X’$_{\text{unstable}}$ (best landmarks). These plots confirm that when best landmarks are employed, prediction accuracy improves on both: the training set and the test set.

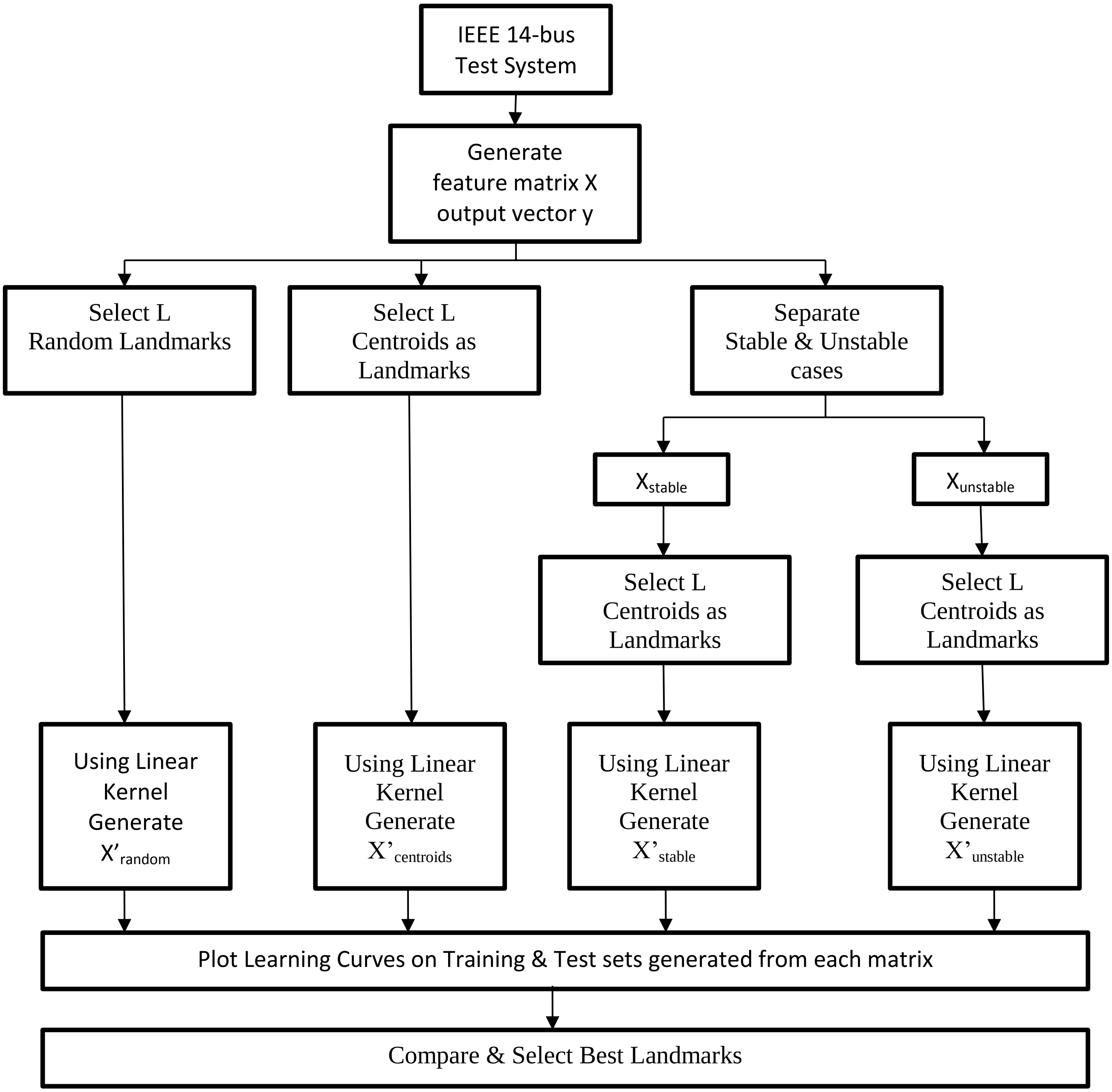


Figure 16: Selection of Landmarks using k-means

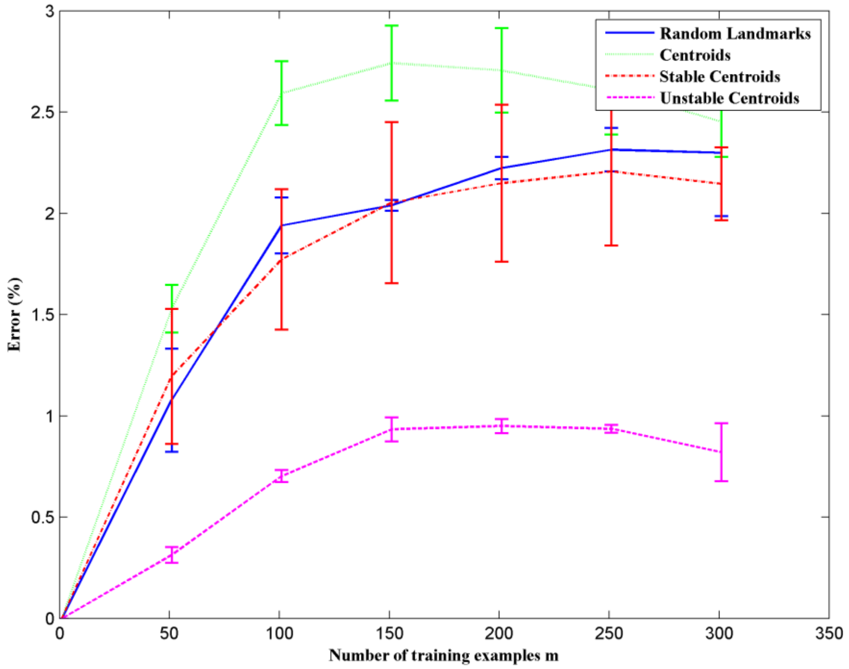


Figure 17a: Learning curves (training set): mean ± SEM (standard error of the mean)

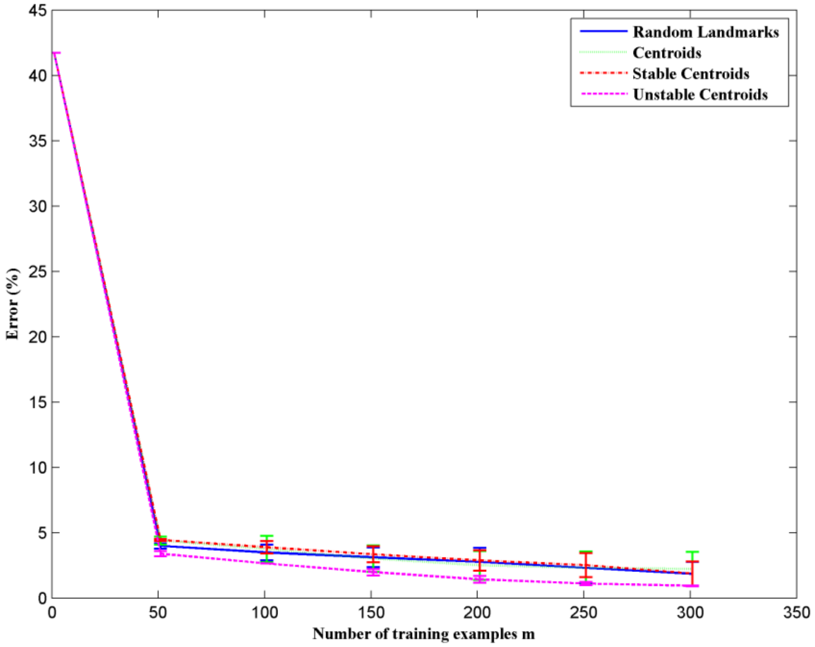


Figure 17b: Learning curves (test set): mean ± SEM (standard error of the mean)

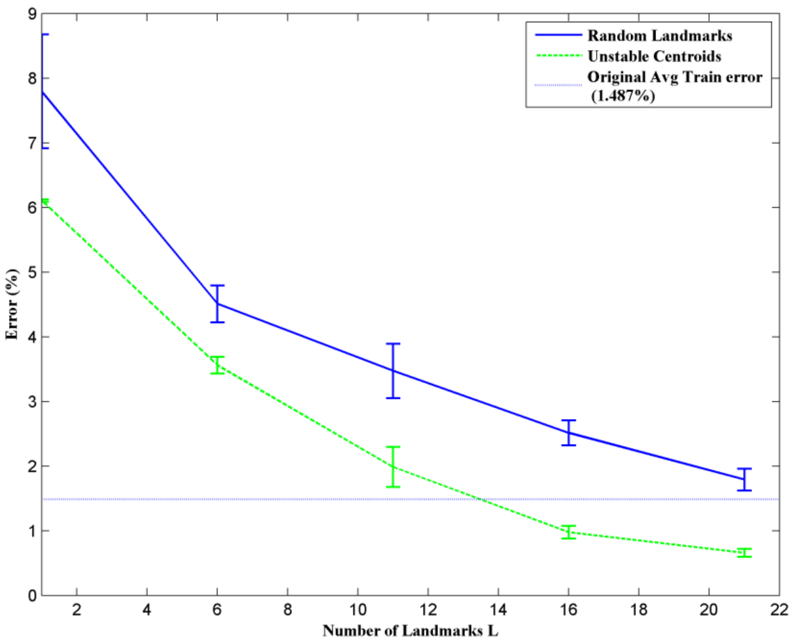


Figure 18a: %Error vs num. of landmarks (training set): mean ± SEM

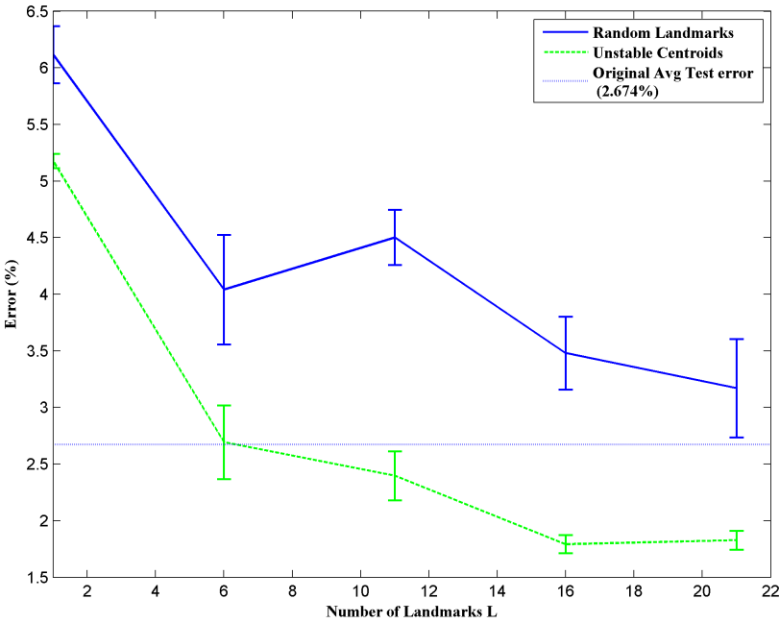


Figure 18b: %Error vs num. of landmarks (test set): mean ± SEM

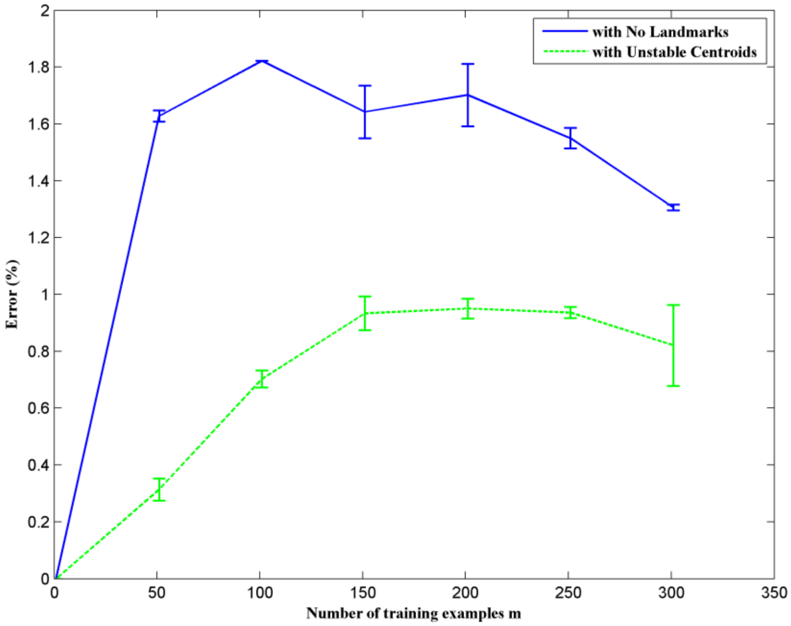


Figure 19a: Learning curves (training set): mean ± SEM

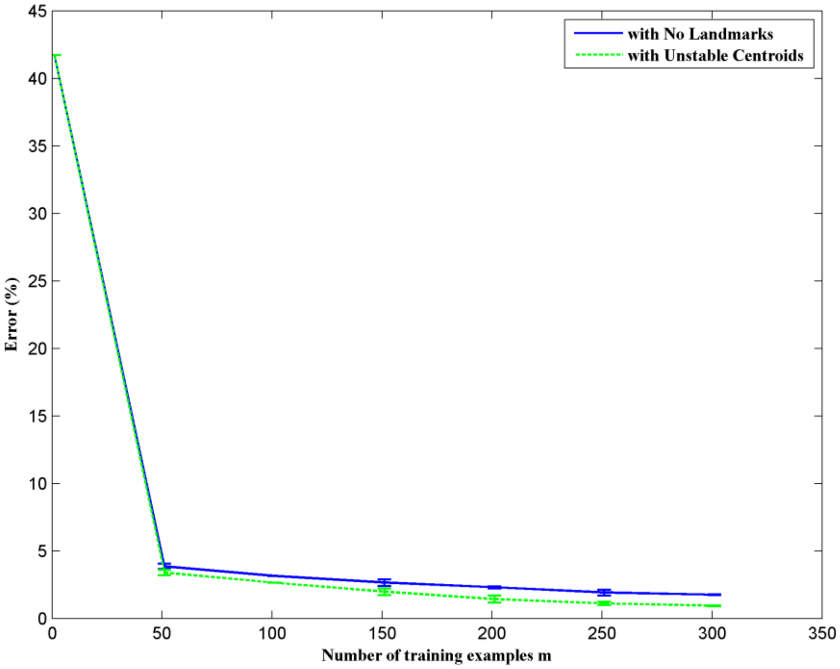


Figure 19b: Learning curves (test set): mean ± SEM

CHAPTER V

SELECTION OF BEST LANDMARKS AND RESULTS

Chapter IV highlights the concept of employing "landmarks" in order to reduce prediction errors on a database given in the form of matrix X and vector y as in equation (6). The practical value of this concept is demonstrated by selecting random landmarks from matrix X in order to show the reduction in prediction errors. A second approach is to employ centroids generated from full matrix X as landmarks by using k-means algorithm. However, such an approach does not use information contained in vector y of the database given by equation (6). For the case of 14-bus test system used in this work, best landmarks are found by generating centroids from only unstable operating points of matrix X, thereby using the information contained in vector y. This chapter presents ranking methodologies in order to select such best landmarks so as to decrease prediction errors on both, training and test sets of the database.

5.1 Based on Fisher's Vector

The Fisher's vector is derived from a technique known as Fisher's linear discriminant analysis (FLDA) [77,78]. FLDA is a dimensionality reduction technique that generates low-dimensional representation of the original data which is optimal for the classification problem under consideration. Hence, such a technique uses information contained in both, matrix X and vector y of the database given by equation (6).

The classification problem considered in this work consists of two classes: "stable" (bit 1) and "unstable" (bit 0) as previously mentioned. For such a two-class case, it is possible to generate an optimal vector *w* which is called as Fisher's vector using FLDA technique. This vector belongs to the same operating space as the database under consideration and maximum class separation is achieved when original data is projected onto it. Therefore, if matrix X consists of operating points of the form $x \in \mathbb{R}^{2n}$ given by equation (7), vector *w* also belongs to the same *2n*-dimensional space i.e. $w \in \mathbb{R}^{2n}$. The one dimensional representation *z* of a vector *x* is derived by projecting it on to the Fisher's vector *w* as follows,

$$z = w^T x \tag{24}$$

Fisher's vector *w* is given by,

$$w = S_w^{-1}(\mu_1 - \mu_0) \tag{25}$$

where S_w = within-class scatter matrix

μ_1, μ_0 = means of stable and unstable classes respectively

The within-class scatter matrix S_w is given by,

$$S_w = \sum_{i:y^{(i)}=1} (x_i - \mu_1)(x_i - \mu_1)^T + \sum_{i:y^{(i)}=0} (x_i - \mu_0)(x_i - \mu_0)^T \tag{26}$$

For m_1 stable and m_0 unstable operating points in matrix X, class means are given by,

$$\mu_1 = \frac{1}{m_1} \sum_{i:y^{(i)}=1} x_i \ ; \quad \mu_0 = \frac{1}{m_0} \sum_{i:y^{(i)}=0} x_i \tag{27}$$

For ranking purposes, similarity between each operating point $x \in \mathbb{R}^{2n}$ and Fisher's vector $w \in \mathbb{R}^{2n}$ is computed using,

$$SIM(x) = w^T x \tag{28}$$

The operating points with high SIM values are the ones that are most similar to the Fisher's vector. Moreover, Fisher's vector represents a direction that spans high density regions in the *2n*-dimensional space under consideration. However, operating points in the low-density regions are the ones that are most useful for prediction [77]. Therefore, the operating points with the lowest SIM values are used for the purpose of generating landmarks.

This ranking methodology is tested on the original 14-bus database generated earlier in the form of matrix X and vector y. The operating points in matrix X are ranked in ascending order of their SIM values with the best ones having lowest SIM values. A total of *L* = 22 centroids are generated from only top 30% of the operating points in matrix X for use as "landmarks". The performance of these landmarks is compared with that of same number of random landmarks and same number of centroids generated from full matrix X for use as landmarks. **Figures 20a and 20b** confirm that the proposed ranking methodology gives the best performance on both, training and test sets in comparison to the other two approaches. Here, computational efficiency is not compromised since *L* is not greater than the total number of columns in the original matrix X (=22 features). **Figures 21a and 21b** compare the effect of increasing number of landmarks on the prediction errors for the case of random landmarks and the case of landmarks selected through the proposed ranking methodology. **Figures 22a and 22b** further prove that landmarks selected through this ranking methodology perform better in comparison to the case that does not employ the concept of "landmarks".

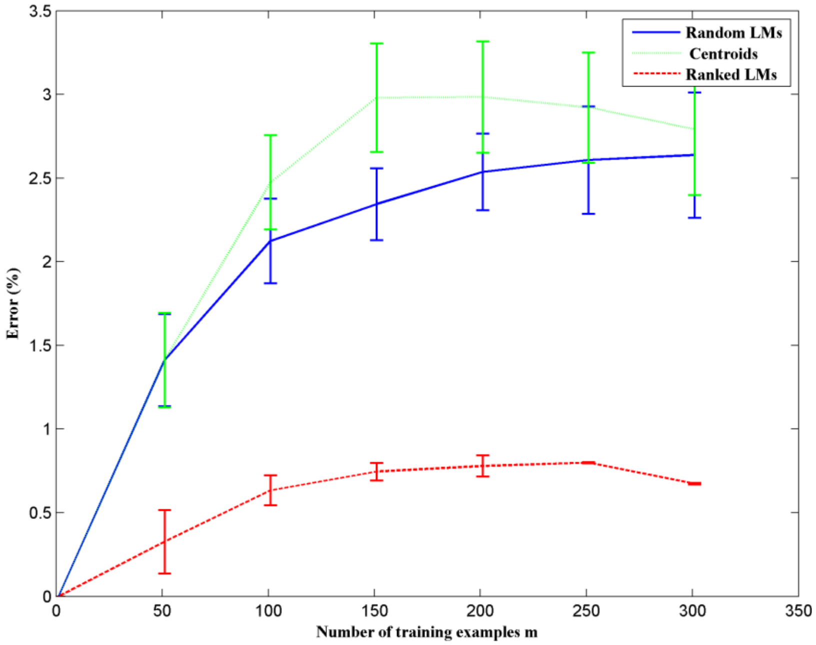


Figure 20a: Learning curves (training set): mean ± SEM (standard error of the mean)

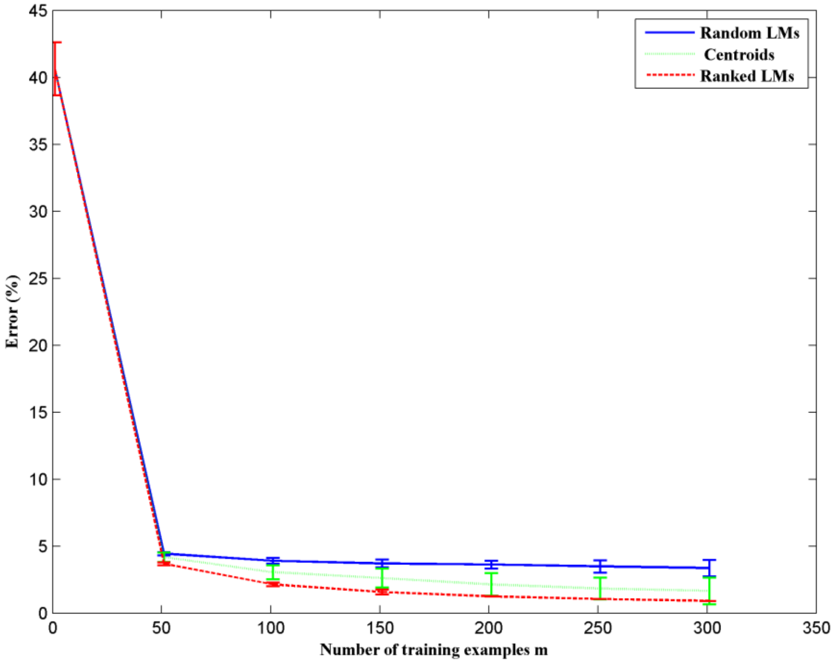


Figure 20b: Learning curves (test set): mean ± SEM (standard error of the mean)

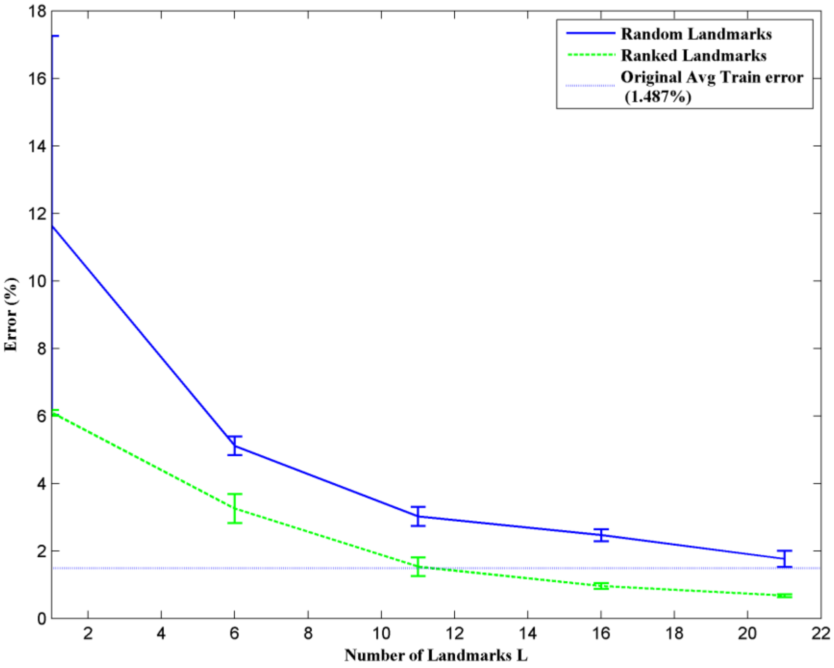


Figure 21a: %Error vs num. of landmarks (training set): mean ± SEM

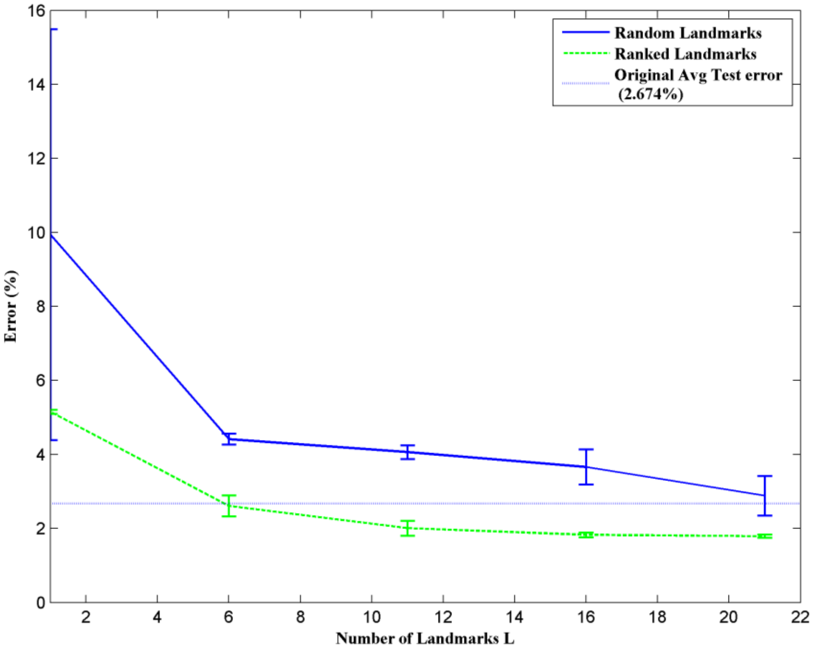


Figure 21b: %Error vs num. of landmarks (test set): mean ± SEM

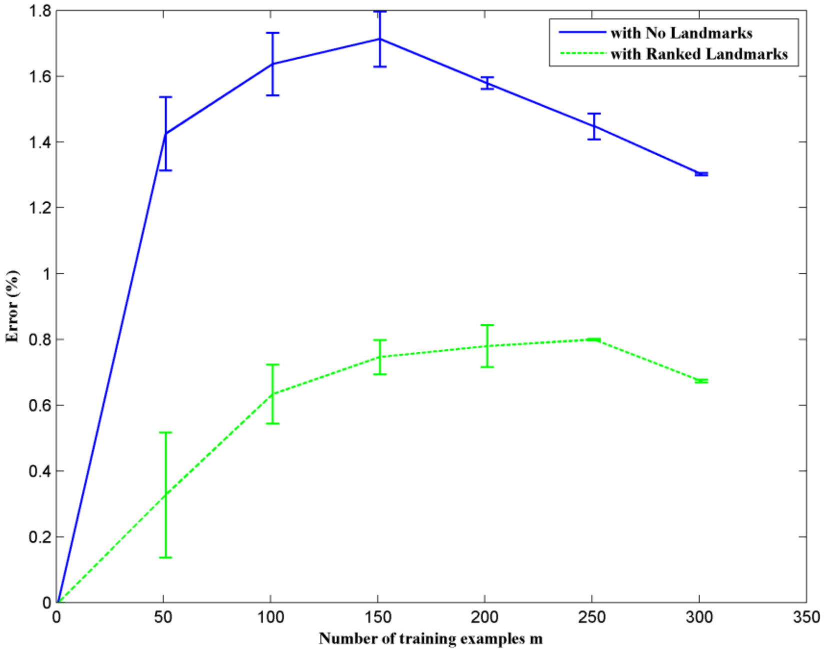


Figure 22a: Learning curves (training set): mean ± SEM

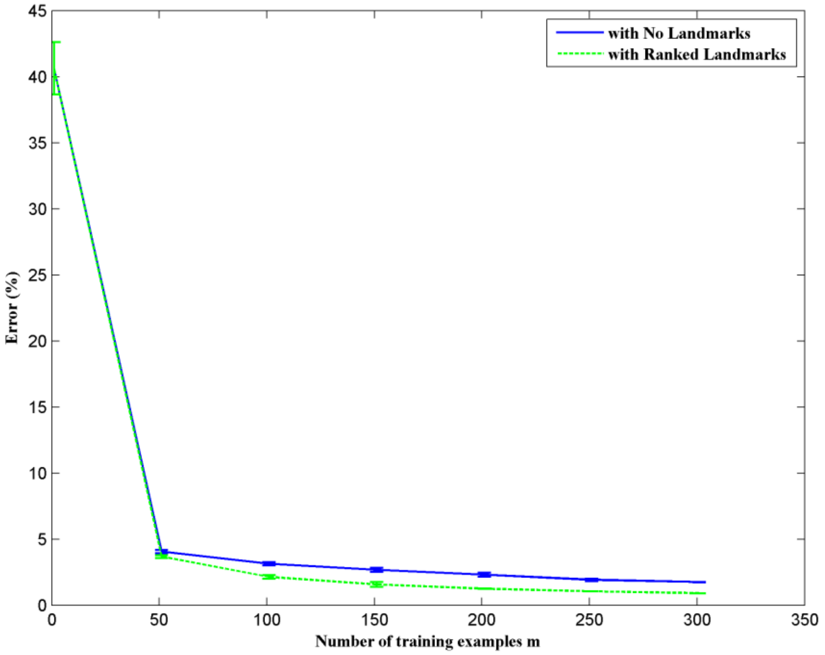


Figure 22b: Learning curves (test set): mean ± SEM

5.2 Based on Class Separation Metric

This section presents a ranking methodology based on Fisher criterion that is used in Fisher's linear discriminant analysis (FLDA) [77,78]. The proposed methodology is used to rank all the operating points contained in matrix X for the purpose of selecting better landmarks as compared to random landmarks or landmarks generated in the form of centroids over full matrix X. The basic idea is to consider each operating point in matrix X as a "landmark", one at a time and then project the entire matrix X on to this operating point. The resulting one-dimensional representation of the entire data is then used to compute a metric for that operating point.

As given in equation (6), size of the original matrix X is (*m x 2n*) where *m* is the total number of operating points of the form $x \in \mathbb{R}^{2n}$ given by equation (7). Matrix X is divided into 2 matrices X_{stable} and $X_{unstable}$ consisting of only stable and unstable cases respectively, thereby utilizing the information contained in vector y of the database given by equation (6). As mentioned in the previous chapter, if matrix X contains m_1 stable cases and m_0 unstable cases such that $m_1 + m_0 = m$, then the size of matrix X_{stable} is (m_1 *x 2n*) and that of matrix $X_{unstable}$ is (m_0 *x 2n*).

A one-dimensional projection of any operating point $x \in \mathbb{R}^{2n}$ on to the landmark $k \in \mathbb{R}^{2n}$ is given by,

$$z = k^T x \tag{29}$$

where *k* and *x* are both of the form given by equation (7).

Similarly, all operating points in matrices X_{stable} and $X_{unstable}$ are linearly projected on to the landmark *k*. The resulting one-dimensional representation of all operating points is derived in the form of vectors Z^k_{stable} (size m_1 *x 1*) and $Z^k_{unstable}$ (size m_0 *x 1*) as follows,

$$
\begin{aligned}
Z^{k}{}_{stable} &= X_{stable}\, k \\
Z^{k}{}_{unstable} &= X_{unstable}\, k
\end{aligned}
\tag{30}
$$

Let *(μ_{k1}, σ_{k1})* and *(μ_{k0}, σ_{k0})* be the maximum likelihood estimates of means and standard deviations computed from Z^{k}_{stable} and Z^{k}_{unstable} respectively. The measure of dissimilarity between stable and unstable operating points in the one-dimensional space represented by landmark *k* is given by a class separation metric (CSM) defined as follows,

$$
CSM(k) = \frac{(\mu_{k1} - \mu_{k0})^{2}}{\sigma_{k1}{}^{2} + \sigma_{k0}{}^{2}} \tag{31}
$$

The CSM defined in equation (31) is the Fisher criterion that is used in Fisher's linear discriminant analysis (FLDA). It is the ratio of the between-class variance to the within-class variance. Moreover, it is maximized when separation between the means of stable and unstable classes is maximized; and the variance within each class is minimized, thus reducing the overlap between the two class distributions. Therefore, higher the CSM value more is the separation between stable and unstable operating points in the one-dimensional space represented by landmark *k*.

For ranking purposes, each operating point $x \in \mathbb{R}^{2n}$ is considered as a landmark and a metric CSM(*x*) is computed using equation (31). The landmarks with high CSM values represent directions that span high density regions in the *2n*-dimensional space under consideration. As mentioned earlier, landmarks that lie in low density regions are the ones that are most useful for prediction [77]. Thus, the operating points with lowest CSM values are selected as the best landmarks.

The proposed ranking methodology is tested on the 14-bus database wherein the operating points are ranked in ascending order of their CSM values. The topmost L = 22 operating points in matrix

X are selected as the best landmarks. **Figures 23a and 23b** confirm that the proposed ranking methodology is better as compared to selecting random landmarks or selecting the centroids generated from matrix X as landmarks. As in the previous section, computational efficiency is not compromised since *L* is not greater than the total number of columns in the original matrix X (=22 features). **Figures 24a and 24b** compare the effect of increasing number of landmarks on the prediction errors for the case of random landmarks and the case of landmarks selected through the proposed ranking methodology. **Figures 25a and 25b** further prove that landmarks selected through this ranking methodology perform better in comparison to the case that does not employ the concept of “landmarks”.

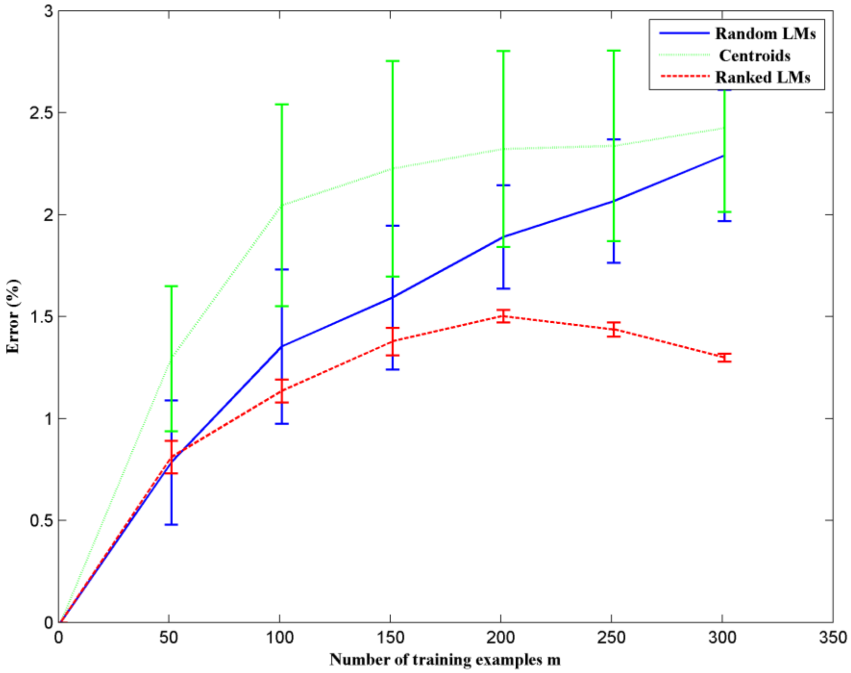


Figure 23a: Learning curves (training set): mean ± SEM (standard error of the mean)

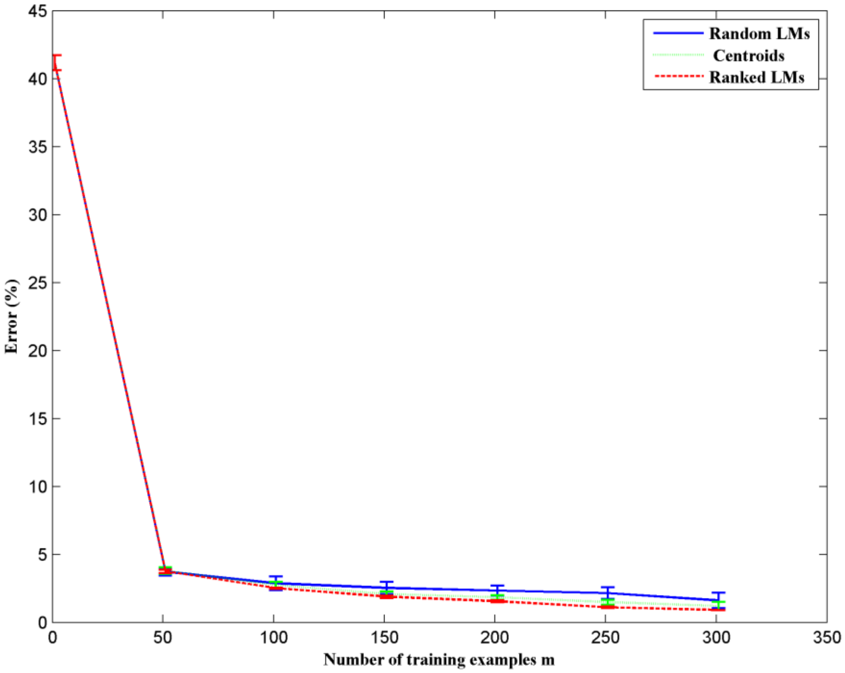


Figure 23b: Learning curves (test set): mean ± SEM (standard error of the mean)

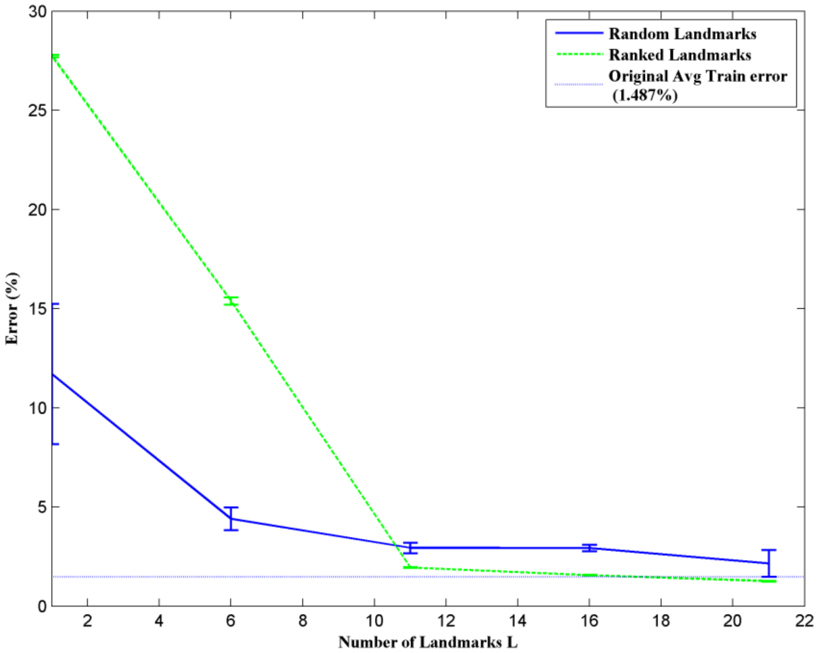


Figure 24a: %Error vs num. of landmarks (training set): mean ± SEM

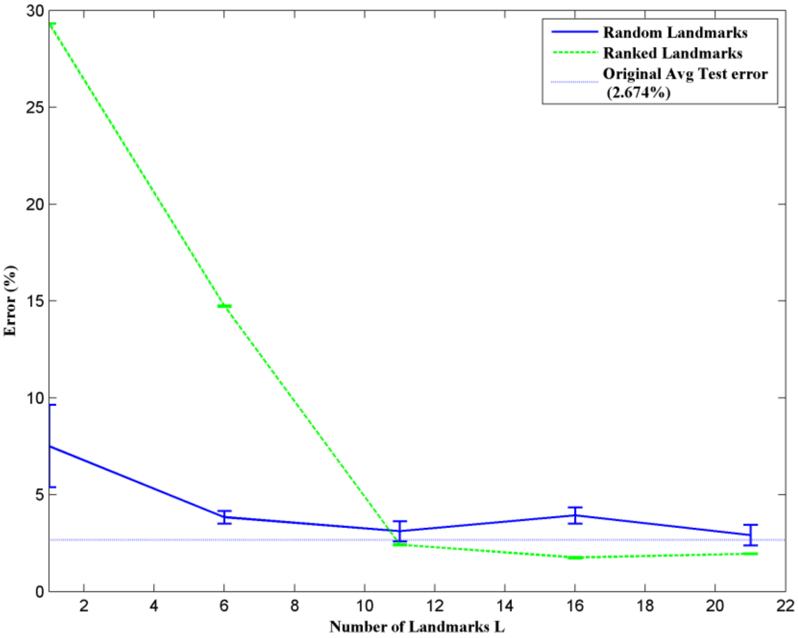


Figure 24b: %Error vs num. of landmarks (test set): mean ± SEM

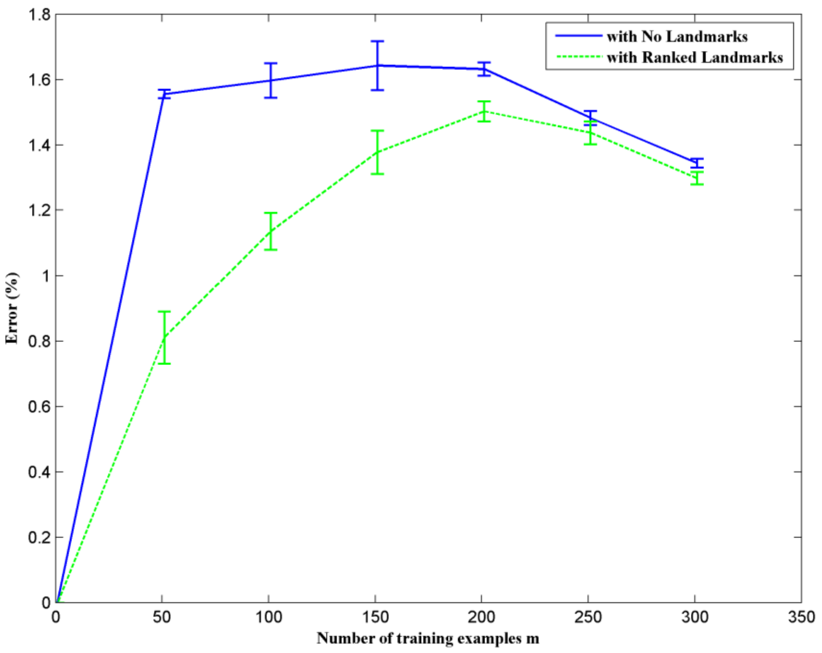


Figure 25a: Learning curves (training set): mean ± SEM

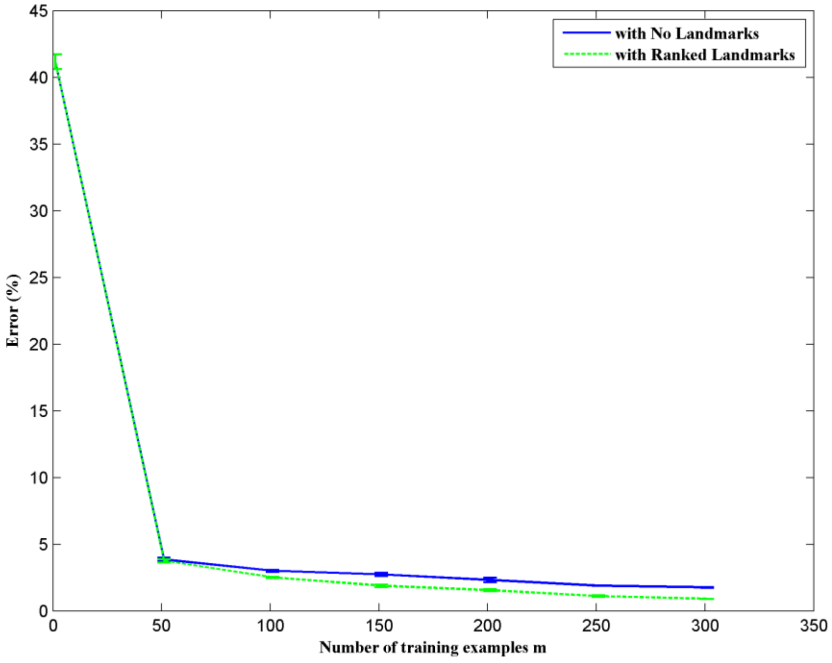


Figure 25b: Learning curves (test set): mean ± SEM

5.3 Based on Fisher Information Metric

This section presents a ranking methodology based on Fisher information distance, which is a measure of dissimilarity between two probability distribution functions [79]. It uses the same idea as described in the previous section, wherein the entire matrix X is projected on to each operating point considered as "landmark" one at a time and the resulting one-dimensional representation of the data is used to compute a metric for that operating point. Such a metric can be computed for every operating point in matrix X so as to rank them for the purpose of selecting best landmarks.

As mentioned earlier, X_{stable} (size *m_1 x 2n*) and $X_{unstable}$ (size *m_0 x 2n*) consisting of stable and unstable cases respectively such that m_1+ m_0 = *m* where *m* is the total number of operating points in matrix X (size *m x 2n*). For a landmark $k \in \mathbb{R}^{2n}$, all points in X_{stable} and $X_{unstable}$ are linearly projected on *k* using equation (29). The resulting one-dimensional representation of all operating points is derived in the form of vectors Z^k_{stable} (size *m_1 x 1*) and $Z^k_{unstable}$ (size *m_0 x 1*) by using equation (30). As defined in the previous section, *(μ_{k1}, σ_{k1})* and *(μ_{k0}, σ_{k0})* are the maximum likelihood estimates of means and standard deviations computed from Z^k_{stable} and $Z^k_{unstable}$ respectively. The measure of dissimilarity between stable and unstable operating points in the one-dimensional space represented by landmark *k* is given by Fisher Information (FI) metric as follows,

$$FI(k) = \sqrt{2}\ln\left(\frac{a_k + b_k + c_k}{4\sigma_{k1}\sigma_{k0}}\right) \tag{32}$$

where

$$a_k = (\mu_{k1} - \mu_{k0})^2$$

$$b_k = 2(\sigma_{k1}{}^2 + \sigma_{k0}{}^2)$$

$$c_k = \sqrt{(a_k + 2(\sigma_{k1} - \sigma_{k0})^2)(a_k + 2(\sigma_{k1} + \sigma_{k0})^2)}$$

For ranking purposes, each operating point $x \in \mathbb{R}^{2n}$ is considered as a landmark and a metric FI(x) is computed using equation (32). The landmarks with high FI values represent directions that span high density regions in the *2n*-dimensional space under consideration. As mentioned earlier, landmarks that lie in low density regions are the ones that are most useful for prediction [77]. Thus, the operating points with lowest FI values are selected as the best landmarks.

The proposed ranking methodology is tested on the 14-bus database wherein the operating points are ranked in ascending order of their FI values. The topmost L = 22 operating points in matrix X are selected as the best landmarks. **Figures 26a and 26b** confirm that the proposed ranking methodology is better as compared to selecting random landmarks or selecting the centroids generated from matrix X as landmarks. As in the previous section, computational efficiency is not compromised since L is not greater than the total number of columns in the original matrix X (=22 features). **Figures 27a and 27b** compare the effect of increasing number of landmarks on the prediction errors for the case of random landmarks and the case of landmarks selected through the proposed ranking methodology. **Figures 28a and 28b** further prove that landmarks selected through this ranking methodology perform better in comparison to the case that does not employ the concept of "landmarks".

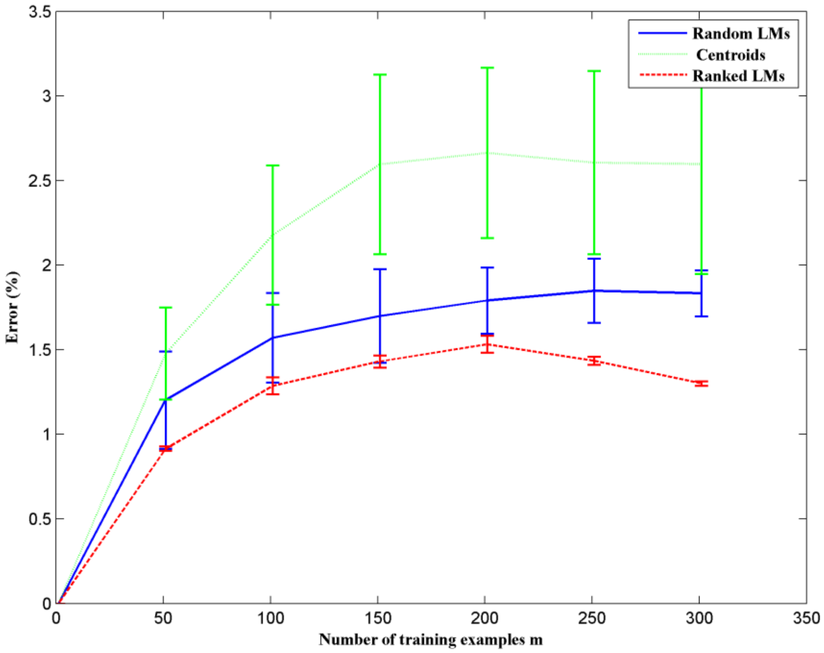


Figure 26a: Learning curves (training set): mean ± SEM (standard error of the mean)

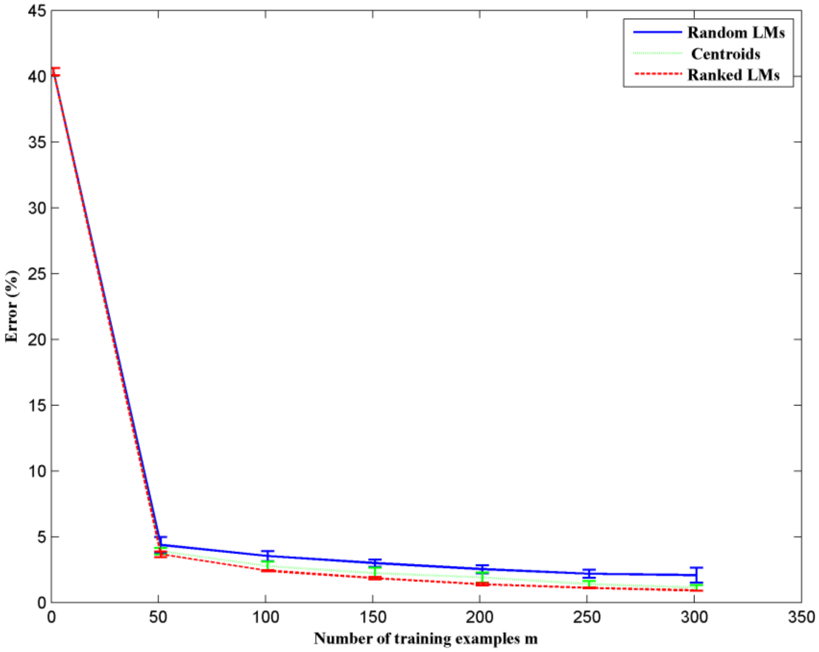


Figure 26b: Learning curves (test set): mean ± SEM (standard error of the mean)

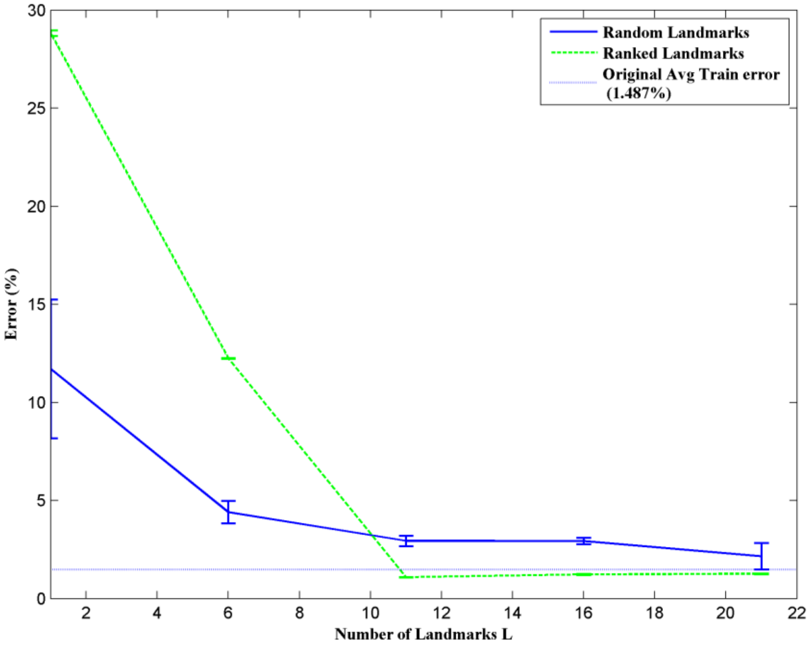


Figure 27a: %Error vs num. of landmarks (training set): mean ± SEM

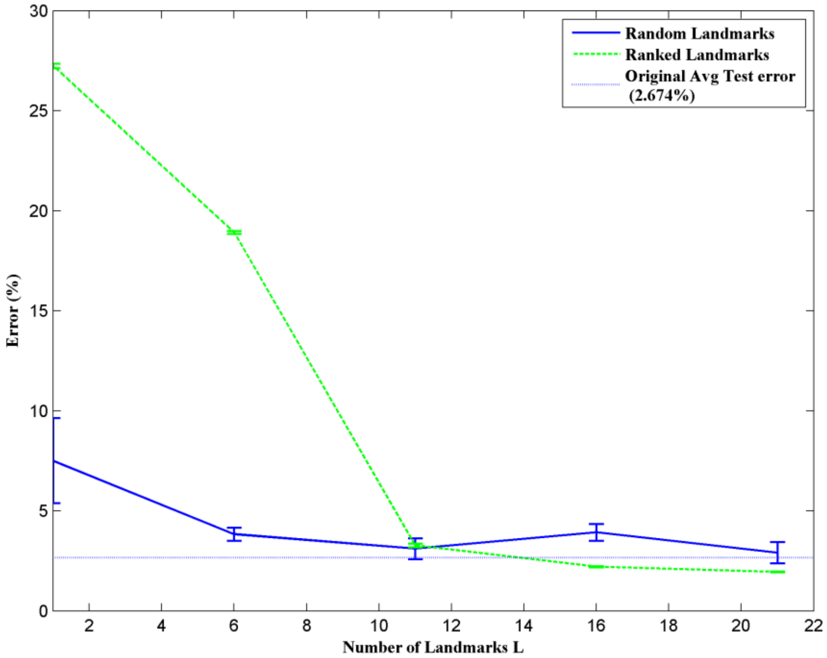


Figure 27b: %Error vs num. of landmarks (test set): mean ± SEM

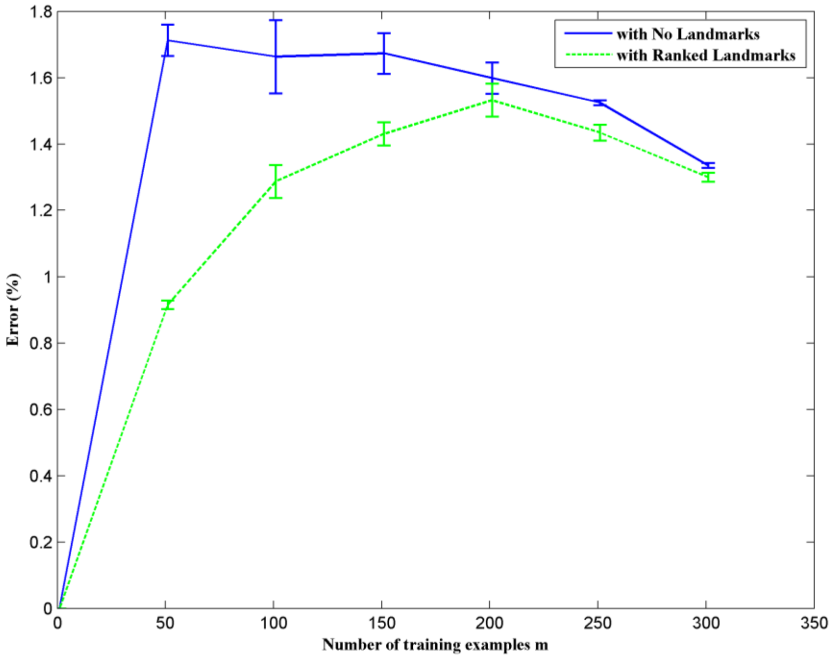


Figure 28a: Learning curves (training set): mean ± SEM

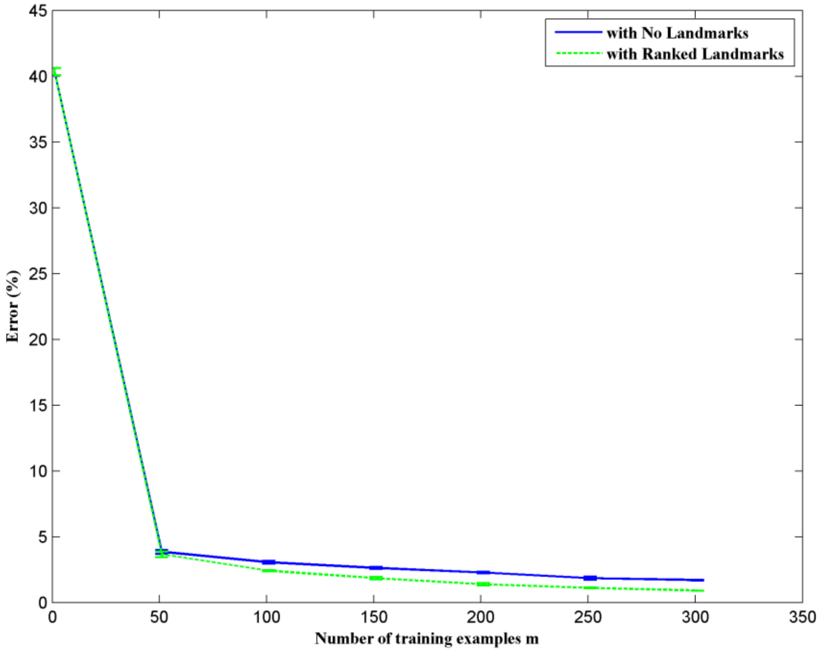


Figure 28b: Learning curves (test set): mean ± SEM

5.4 Based on Gaussian Mixture Model

This section presents a ranking methodology based on probabilistic approach known as Gaussian mixture model [77]. In this methodology, it is assumed that all the operating points of the form $x \in \mathbb{R}^{2n}$ in matrix X belong to a probability distribution model which is a mixture of Gaussians. The classification problem considered in this work consists of two classes: "stable" (bit 1) and "unstable" (bit 0) as previously mentioned. For such a two-class case, distribution model is assumed to be a mixture of two Gaussians; one for "stable" operating points and the other for "unstable" operating points.

For a class *K*, a multivariate Gaussian in *D*-dimensional space with mean μ_K and covariance matrix Σ_K is defined as follows,

$$N(x \mid \mu_K, \Sigma_K) = \frac{1}{(2\pi)^{D/2}\left|\Sigma_K\right|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu_K)^T \Sigma_K{}^{-1}(x-\mu_K)\right] \tag{33}$$

The probability distribution model is a mixture of two classes: stable ($K = 1$) and unstable ($K = 0$) and is given as follows,

$$p(x \mid \theta) = \pi_1 N(x \mid \mu_1, \Sigma_1) + \pi_0 N(x \mid \mu_0, \Sigma_0) \tag{34}$$

where μ_1 and Σ_1 = parameters of class "stable"

μ_0 and Σ_0 = parameters of class "unstable"

π_1 and π_0 = mixing weights for "stable" and "unstable" classes respectively

θ = list of model parameters = $(\pi_1, \mu_1, \Sigma_1, \pi_0, \mu_0, \Sigma_0)$

The two approaches that are used to compute the model parameters given by θ form the basis of the ranking methodology presented in this section. As already described in the previous chapter, the two approaches are called as “supervised” and “unsupervised” approaches. The supervised approach utilizes entire database i.e. matrix X and vector y as given by equation (6) whereas unsupervised approach utilizes only matrix X and not vector y. The ranking methodology for selecting best landmarks is formulated based on the two models thus derived.

The “supervised” approach fits a model using the concept of maximum likelihood estimation. The maximum likelihood estimates (MLE) of the model parameters given by θ are computed using matrix X and vector y. For this purpose, matrix X is divided into 2 matrices X_{stable} and X_{unstable} consisting of only stable and unstable cases respectively, thereby utilizing the information contained in vector y. As mentioned in the previous chapter, if matrix X (size *m x 2n*) contains m_1 stable cases and m_0 unstable cases such that $m_1 + m_0 = m$, then the size of matrix X_{stable} is (m_1 *x 2n*) and that of matrix X_{unstable} is (m_0 *x 2n*). Matrix X_{stable} is used to compute the MLEs of parameters (π_1, μ_1, Σ_1) and matrix X_{unstable} is used to compute the MLEs of parameters (π_0, μ_0, Σ_0). The MLEs are given as follows,

$$\pi_1 = \frac{m_1}{m} \quad ; \quad \pi_0 = \frac{m_0}{m} \tag{35}$$

$$\mu_1 = \frac{1}{m_1} \sum_{i:y^{(i)}=1} x_i \quad ; \quad \mu_0 = \frac{1}{m_0} \sum_{i:y^{(i)}=0} x_i^{\;T} \tag{36}$$

$$\Sigma_1 = \frac{1}{m_1} \sum_{i:y^{(i)}=1} (x_i - \mu_1)(x_i - \mu_1)^T \; ; \; \Sigma_0 = \frac{1}{m_0} \sum_{i:y^{(i)}=0} (x_i - \mu_0)(x_i - \mu_0)^T \tag{37}$$

The "unsupervised" approach fits a model using the concept of expectation-maximization (EM). For the model given in equation (34), EM algorithm computes its parameters by using only matrix X and not vector y. EM algorithm is an iterative improvement algorithm consisting of the following two steps after initialization,

1) E-step:

Given the parameter list *θ*, class posterior probabilities *p(y=1|x)* and *p(y=0|x)* are computed for each operating point *x* ∈ ℝ2n in matrix X. The operating points are assigned to the class *(y=1 or 0)* with maximum posterior probability as follows,

$$y^{(i)} = \arg\max_{K=0,1}\left(-\frac{1}{2}(x^{(i)}-\mu_K)^T\Sigma_K{}^{-1}(x^{(i)}-\mu_K)-\frac{1}{2}\log\left|\Sigma_K\right|+\log\pi_K\right) \qquad i=1,2......,m \tag{38}$$

Therefore, in this step, a vector y consisting of class labels is generated for the original matrix X thereby resulting in new class distributions.

2) M-step:

The parameter list *θ* is updated by computing MLEs of new class distributions using equations (35-37).

The convergence of EM algorithm depends on the initialization and hence, may lead to local maxima. Since full database in the form of matrix X and vector y is available for the purpose of this work, EM algorithm is initialized by using MLEs from the supervised approach. The model derived from EM algorithm can be termed as an "unsupervised model" since it does not use vector y, other than for initialization. In this work, an EM routine from the MATLAB package called PMTK is used for simulation purposes [80].

For ranking purposes, probabilities $p(x|\theta_{supervised})$ and $p(x|\theta_{unsupervised})$ are computed using equation (34) for each operating point $x \in \mathbb{R}^{2n}$ in matrix X, where $\theta_{supervised}$ and $\theta_{unsupervised}$ are the model parameters derived from supervised and unsupervised approach respectively. A metric that captures class information (CI) is formulated as follows,

$$CI(x) = \left| p(x \mid \theta_{\sup ervised}) - p(x \mid \theta_{un\sup ervised}) \right| \tag{39}$$

The operating points with high CI values are the ones that contain maximum class information and hence, lie in the high density regions of the *2n*-dimensional space under consideration. As mentioned earlier, landmarks that lie in low density regions are the ones that are most useful for prediction [77]. Thus, the operating points with lowest CI values are selected as the best landmarks.

This ranking methodology is tested on the original 14-bus database generated earlier in the form of matrix X and vector y. The operating points in matrix X are ranked in ascending order of their CI values with the best ones having lowest CI values. A total of L = 22 centroids are generated from only top 30% of the operating points in matrix X for use as “landmarks”. The performance of these landmarks is compared with that of same number of random landmarks and same number of centroids generated from full matrix X for use as landmarks. **Figures 29a and 29b** confirm that the proposed ranking methodology gives the best performance on both, training and test sets in comparison to the other two approaches. Here, computational efficiency is not compromised since L is not greater than the total number of columns in the original matrix X (=22 features). **Figures 30a and 30b** compare the effect of increasing number of landmarks on the prediction errors for the case of random landmarks and the case of landmarks selected through the proposed ranking methodology. **Figures 31a and 31b** further prove that landmarks selected through this ranking methodology perform better in comparison to the case that does not employ the concept of “landmarks”.

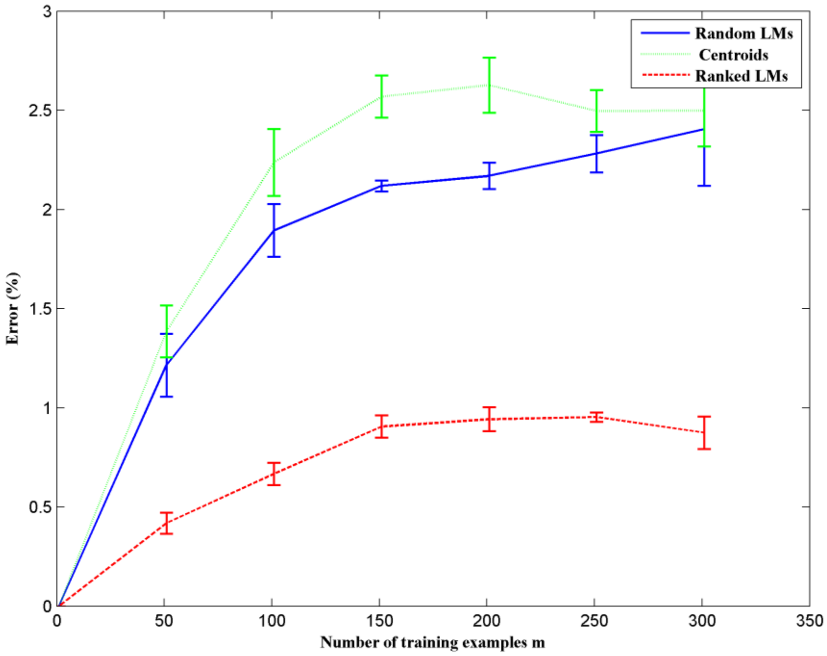


Figure 29a: Learning curves (training set): mean ± SEM (standard error of the mean)

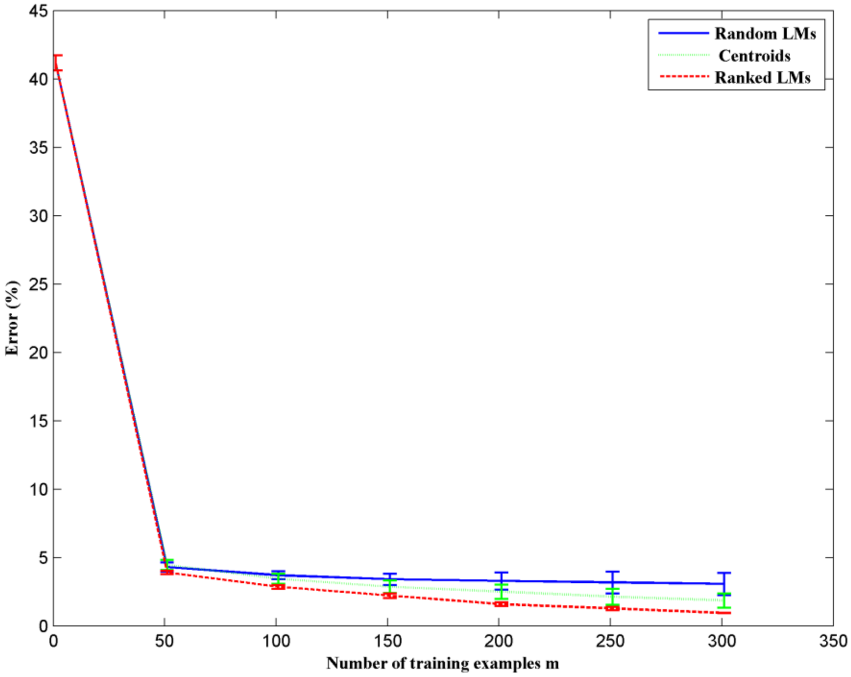


Figure 29b: Learning curves (test set): mean ± SEM (standard error of the mean)

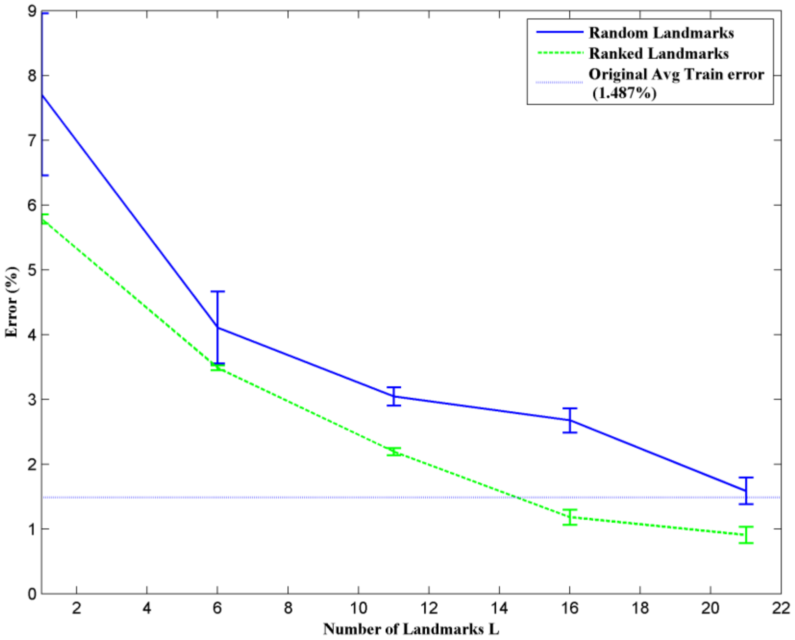


Figure 30a: %Error vs num. of landmarks (training set): mean ± SEM

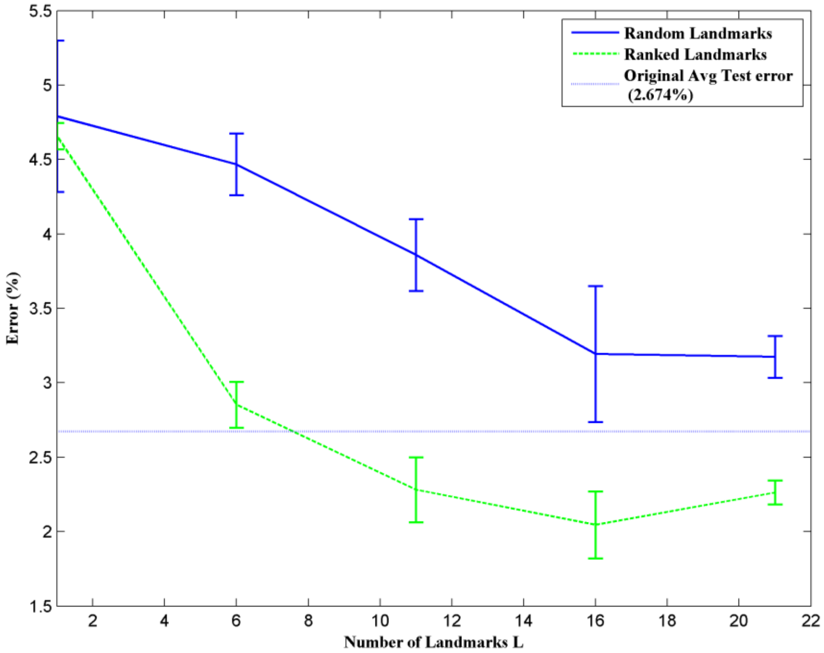


Figure 30b: %Error vs num. of landmarks (test set): mean ± SEM

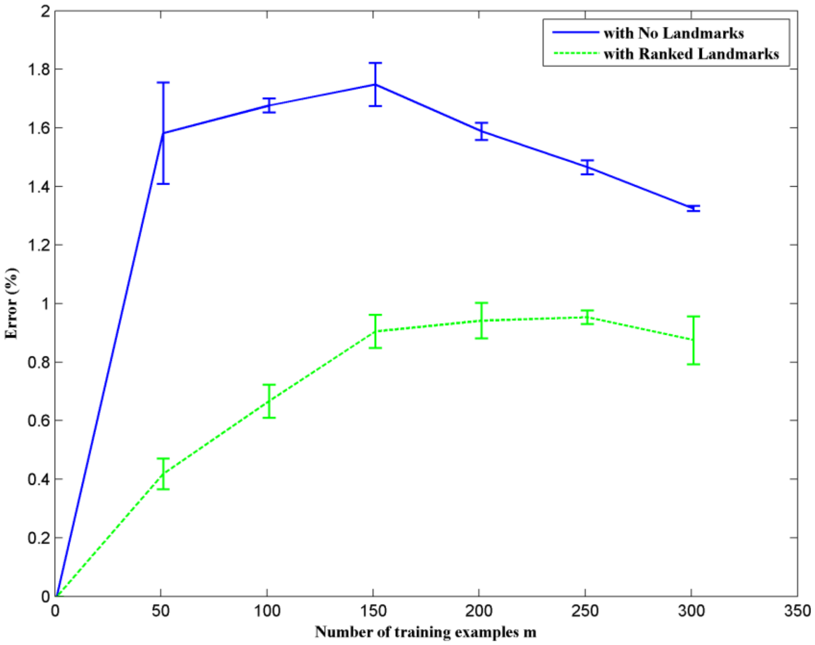


Figure 31a: Learning curves (training set): mean ± SEM

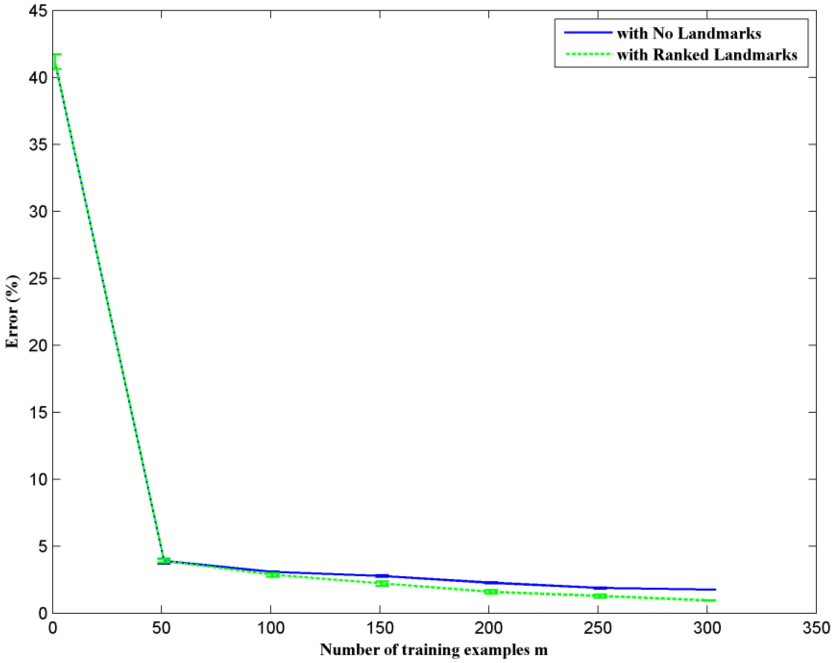


Figure 31b: Learning curves (test set): mean ± SEM

5.5 Comparison of Ranking Methodologies

This chapter presents four ranking methodologies to select best “landmarks” in the operational space of the power system under consideration. Each methodology highlights the importance of selecting a few operating points of the system as “landmarks” so as to reduce prediction errors on the database of the form given by equation (6). Such landmarks facilitate fast prediction of power system’s resiliency against any disturbance or contingency under consideration, thus enabling the grid to assess its dynamic security in real-time. Moreover, data burden and information overload can be reduced significantly by retaining only such landmarks from the database for future use. The four ranking methodologies are as follows:

1) Based on Fisher’s vector

In this approach, Fisher vector *w* is derived from database given in the form of matrix X and vector y by using equation (25). The ranking methodology is based on similarity between each operating point $x \in \mathbb{R}^{2n}$ in matrix X and the Fisher vector *w*. An index SIM(*x*) is calculated for each operating point using equation (28) and the operating points with lowest SIM values are found to generate the best landmarks.

2) Based on class separation metric

In this approach, each operating point $x \in \mathbb{R}^{2n}$ is considered as a “landmark”, one at a time and the entire matrix X is projected on to this operating point. The resulting one-dimensional representation of the entire data is then used to compute a class separation metric CSM(*x*) for that operating point using equation (31). The operating points with lowest CSM values are found to generate the best landmarks.

3) <u>Based on Fisher information metric</u>

As in the previous approach, here again the entire matrix X is projected on to each operating point $x \in \mathbb{R}^{2n}$ considered as "landmark" one at a time and the resulting one-dimensional representation of the data is used to compute a Fisher information metric FI(x) for that operating point using equation (32). The operating points with lowest FI values are found to generate the best landmarks.

4) <u>Based on Gaussian mixture model</u>

In this approach, data is assumed to belong to a probability distribution model given by equation (34), which is a mixture of two Gaussians; one for "stable" operating points and the other for "unstable" operating points. The model parameters are derived by using "supervised" and "unsupervised" approaches. A class information metric CI(x) is calculated for each operating point $x \in \mathbb{R}^{2n}$ using equation (39) and the operating points with lowest CI values are found to generate the best landmarks.

The above four ranking methodologies are found to generate better "landmarks" as compared to the following two general approaches,

1) Selecting random operating points from matrix X as landmarks
2) Selecting centroids generated from full matrix X as landmarks

Therefore, by enabling selection of best landmarks from the database, the ranking methodologies presented in this chapter can play an important role in reducing data burden and thereby improving computational efficiency.

The four methodologies are compared by testing their performance on the original 14-bus database generated earlier in the form of matrix X and vector y. **Figures 32a and 32b** show that ranking methodology based on Fisher's vector performs best for the 14-bus database used in this work. In particular, the best ranking methodology would depend on the database under consideration.

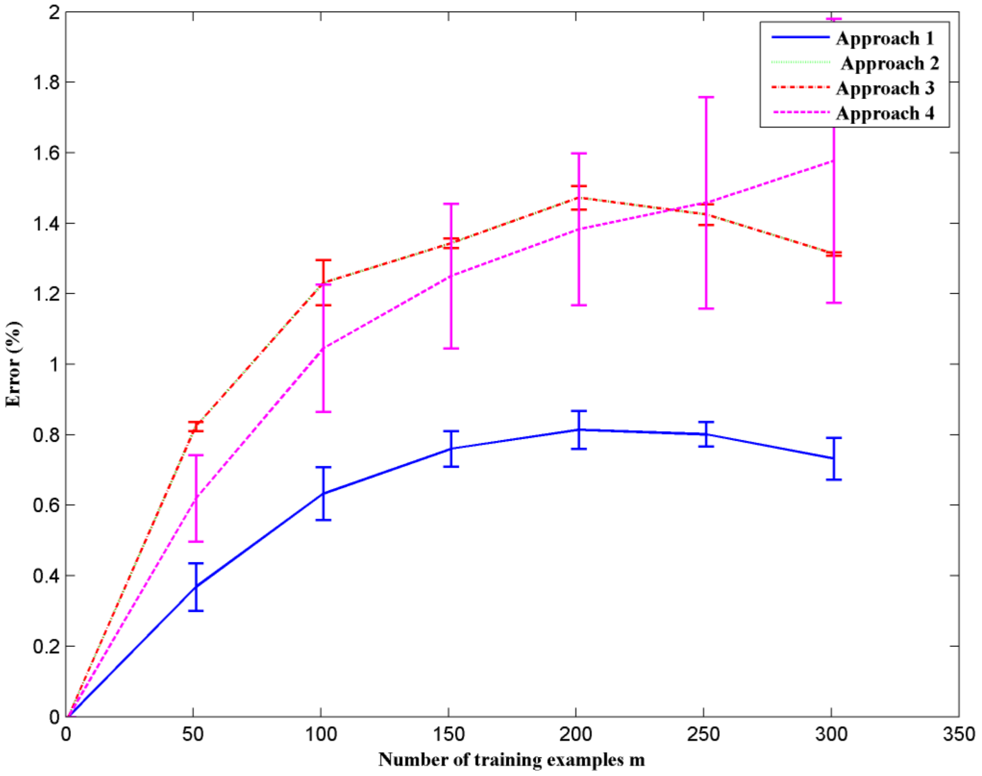


Figure 32a: Learning curves (training set): mean ± SEM (standard error of the mean)

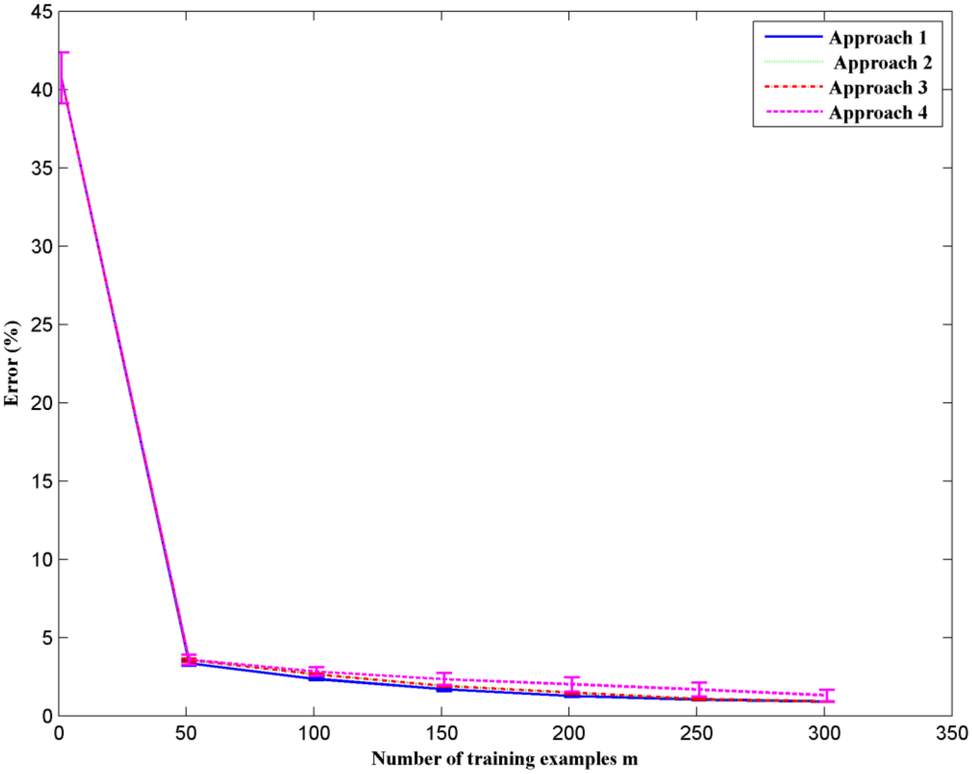


Figure 32b: Learning curves (test set): mean ± SEM (standard error of the mean)

CHAPTER VI

CONCLUDING REMARKS AND FUTURE WORK

The ability to assess the current state of the power system instantaneously is the key attribute needed to identify and establish enhanced grid resilience. Electric power entities carry out offline studies on power system models of different sizes, thus generating offline databases. Machine learning techniques can be employed to use such databases in order to learn the inherent non-linear relationships that exist among different power system parameters. Such useful information can be used later for online real-time security analysis of the power system.

This work presents a framework to apply machine learning (ML) techniques for real-time assessment of the grid resilience against any disturbance or contingency with respect to its static and dynamic security by using offline databases. Further, this work demonstrates a strategy to select a few operating points of the system as “landmarks” in order to improve prediction accuracy without compromising computational efficiency. The four ranking methodologies presented in this work can be used to rank all operating points of the system in order to select the best landmarks for future use. Only the best landmarks can be retained thereby reducing data burden and information overload. The proposed heuristics are tested on a database generated from IEEE 14-bus power system.

Moreover, the proposed approach can be extended for analyzing grid resilience under multiple contingencies. Metrics for grid resilience can be developed based on such multi-contingency analyses. With large-scale penetration of renewable energy in to the current grid and emergence of micro-grids, future grid applications would require real-time training in order to extract useful information on a continuous basis. Machine learning techniques can accommodate such complex requirements posed by the continually changing nature of the electric grid and hence, would definitely play an important role in realizing next-gen real-time power system applications.

Future work would mainly focus on further improvement of the landmarks selection strategy and the prediction accuracy. Essentially, it would be desirable to select the minimum number of landmark points so as to improve computational efficiency further. The following concepts would be explored in order to achieve the desired results,

- Mutual Information: To select those landmark points which contain maximum information about the output vector y
- k-means with Bayesian nonparametric models such as Dirichlet process (DP): To select optimal number of cluster centroids as landmark points
- An approach to model power system data as “probabilistic mixture models”
- Comparison of prediction accuracy using different types of kernels
- Study different types of metrics available in the ML and statistics domain
- Incorporate larger power system models and different ML techniques

Finally, all the research outcomes need to be incorporated in to a ML framework which could handle multiple contingencies of the power system under consideration. Such a framework can lead to a comprehensive tool for assessment of grid resiliency in real-time.

REFERENCES

[1] P. Kundur*, et al.*, "Definition and classification of power system stability IEEE/CIGRE joint task force on stability terms and definitions," *IEEE Transactions on Power Systems, ,* vol. 19, pp. 1387-1401, 2004.

[2] L. Wang and K. Morison, "Implementation of online security assessment," *Power and Energy Magazine, IEEE,* vol. 4, pp. 46-59, 2006.

[3] A. Fouad*, et al.*, "Dynamic security assessment practices in North America," *IEEE Transactions on Power Systems,* vol. 3, pp. 1310-1321, 1988.

[4] L. L. Grigsby, *Power system stability and control*, Third ed.: CRC press, 2012.

[5] J. Jardim, C. Neto, and M. G. dos Santos, "Brazilian system operator online security assessment system," in *Power Systems Conference and Exposition*, 2006, pp. 7-12.

[6] J. Tong and L. Wang, "Design of a DSA tool for real time system operations," in *International Conference on Power System Technology*, 2006, pp. 1-5.

[7] S. C. Savulescu, *Real-time stability assessment in modern power system control centers*: John Wiley & Sons, 2009.

[8] H.-D. Chiang, J. Tong, and Y. Tada, "On-line transient stability screening of 14,000-bus models using TEPCO-BCU: Evaluations and methods," in *Power and Energy Society General Meeting*, 2010, pp. 1-8.

[9] Z. Yao and D. Atanackovic, "Issues on security region search by online DSA," in *Power and Energy Society General Meeting*, 2010, pp. 1-4.

[10] J. Ekanayake, N. Jenkins, K. Liyanage, J. Wu, and A. Yokoyama, *Smart grid: technology and applications*: John Wiley & Sons, 2012.

[11] Y.-H. Song, A. Johns, and R. Aggarwal, *Computational intelligence applications to power systems* vol. 15: Springer Science & Business Media, 1996.

[12] K. Warwick, A. Ekwue, and R. Aggarwal, *Artificial intelligence techniques in power systems*: Iet, 1997.

[13] W. Ongsakul and V. N. Dieu, *Artificial intelligence in power system optimization*: CRC Press, 2013.

[14] V. Ajjarapu and C. Christy, "The continuation power flow: a tool for steady state voltage stability analysis," *IEEE Transactions on Power Systems,* vol. 7, pp. 416-423, 1992.

[15] *Power Systems Test Case Archive*. Available: http://www.ee.washington.edu/research/pstca/

[16] *F. Milano. PSAT, Matlab-Based Power System Analysis Toolbox.* . Available: http://faraday1.ucd.ie/psat.html

[17] D. Ruiz-Vega and M. Pavella, "A comprehensive approach to transient stability control. I. Near optimal preventive control," *IEEE Transactions on Power Systems,* vol. 18, pp. 1446-1453,2003.

[18] A. Fouad, Q. Zhou, and V. Vittal, "System vulnerability as a concept to assess power system dynamic security," *IEEE Transactions on Power Systems,* vol. 9, pp. 1009-1015, 1994.

[19] D. Cirio, E. Gaglioti, S. Massucco, A. Pitto, and F. Silvestro, "An extensive Dynamic Security Assessment analysis on a large realistic electric power system," in *Power Tech, 2007 IEEE Lausanne*, 2007, pp. 285-292.

[20] D. Semitekos and N. Avouris, "A Machine Learning Toolkit for Power Systems Security Analysis," *Proc. Med Power Athens,* 2002.

[21] A. Edwards, K. Chan, R. Dunn, and A. Daniels, "Transient stability screening using artificial neural networks within a dynamic security assessment system," *IEE Proceedings-Generation, Transmission and Distribution,* vol. 143, pp. 129-134, 1996.

[22] V. Brandwajn, A. Kumar, A. Ipakchi, A. Bose, and S. D. Kuo, "Severity indices for contingency screening in dynamic security assessment," *IEEE Transactions on Power Systems,* vol. 12, pp. 1136-1142, 1997.

[23] J. M. G. Alvarez and P. E. Mercado, "A new approach for power system online DSA using distributed processing and fuzzy logic," *Electric power systems research,* vol. 77, pp. 106-118, 2007.

[24] G. Bizjak, U. Kerin, R. Krebs, E. Lerch, and O. Ruhle, "Vision 2020 dynamic security assessment in real time environment," in *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, 2008, pp. 1-7.

[25] J. Geeganage, U. Annakkage, T. Weekes, and B. Archer, "Application of energy-based power system features for dynamic security assessment," 2014.

[26] T. Assis, A. Nohara, and T. Valentini, "Power system dynamic security assessment through a neuro-fuzzy scheme," in *Intelligent System Applications to Power Systems, 2009. ISAP'09. 15th International Conference on*, 2009, pp. 1-6.

[27] K. Verma and K. Niazi, "Rotor trajectory index for transient security assessment using Radial Basis Function Neural Network," in *PES General Meeting| Conference & Exposition, 2014 IEEE*, 2014, pp. 1-5.

[28] E. Ciapessoni, D. Cirio, and A. Pitto, "Cascadings in large power systems: Benchmarking static vs. time domain simulation," in *PES General Meeting| Conference & Exposition, 2014 IEEE*, 2014, pp. 1-5.

[29] C. Fu and A. Bose, "Contingency ranking based on severity indices in dynamic security analysis," *IEEE Transactions on Power Systems,* vol. 14, pp. 980-985, 1999.

[30] N. Kandil, S. Georges, and M. Saad, "Rapid Power System Transient Stability Limit Search Using Signal Energy and Neural Networks," in *Industrial Electronics, 2006 IEEE International Symposium on*, 2006, pp. 1658-1661.

[31] H.-D. Chiang, C.-S. Wang, and H. Li, "Development of BCU classifiers for on-line dynamic contingency screening of electric power systems," *IEEE Transactions on Power Systems,* vol. 14, pp. 660-666, 1999.

[32] V. Chadalavada*, et al.*, "An on-line contingency filtering scheme for dynamic security assessment," *IEEE Transactions on Power Systems,* vol. 12, 1997.

[33] C. Tang, C. Graham, M. El-Kady, and R. Alden, "Transient stability index from conventional time domain simulation," *IEEE Transactions on Power Systems,* vol. 9, pp. 1524-1530, 1994.

[34] M. Khan*, et al.*, "Parallel detrended fluctuation analysis for fast event detection on massive pmu data," *IEEE Transactions on Smart Grid,* vol. 6, pp. 360-368, 2015.

[35] P. Du, "Using disturbance data to monitor primary frequency response for power system interconnections," in *PES General Meeting| Conference & Exposition, 2014 IEEE*, 2014, pp. 1-1.

[36] U. Jayatunga, S. Perera, P. Ciufo, and A. P. Agalgaonkar, "Voltage unbalance emission assessment in interconnected power systems," *IEEE Transactions on Power Delivery,* vol. 28, pp. 2383-2393, 2013.

[37] U. Kerin and E. Lerch, "Dynamic security assessment to improve system stability," in *Science and Technology, 2011 EPU-CRIS International Conference on*, 2011, pp. 1-7.

[38] A. Fouad, F. Aboytes, and V. Carvalho, "Dynamic security assessment practices in North America," *IEEE Transactions on Power Systems,* vol. 3, pp. 1310-1321, 1988.

[39] J. Tong and L. Wang, "Design of a DSA tool for real time system operations," in *Power System Technology, 2006. PowerCon 2006. International Conference on*, 2006, pp. 1-5.

[40] J. Viikinsalo, A. Martin, K. Morison, L. Wang, and F. Howell, "Transient security assessment in real-time at southern company," in *Power Systems Conference and Exposition, 2006. PSCE'06. 2006 IEEE PES*, 2006, pp. 13-17.

[41] J. Chen, T. Mortensen, B. Blevins, C. Thompson, and P. Du, "ERCOT experience in using online stability analysis in real-time operations," in *Power & Energy Society General Meeting, 2015 IEEE*, 2015, pp. 1-5.

[42] Q. F. Zhang*, et al.*, "Advanced grid event analysis at ISO New England using PhasorPoint," in *PES General Meeting| Conference & Exposition, 2014 IEEE*, 2014, pp. 1-5.

[43] Y. Mansour*, et al.*, "BC Hydro's on-line transient stability assessment (TSA) model development, analysis and post-processing," *IEEE Transactions on Power Systems,* vol. 10, pp. 241-253, 1995.

[44] C. Neto, M. Quadros, M. Santos, and J. Jardim, "Brazilian system operator online security assessment system," in *IEEE PES General Meeting*, 2010.

[45] W. Wu, B. Zhang, H. Sun, and Y. Zhang, "Development and application of on-line dynamic security early warning and preventive control system in China," in *Power and Energy Society General Meeting, 2010 IEEE*, 2010, pp. 1-7.

[46] L. Loud*, et al.*, "Hydro-Québec's challenges and experiences in on-line DSA applications," in *Power and Energy Society General Meeting, 2010 IEEE*, 2010, pp. 1-8.

[47] Z. Yao and D. Atanackovic, "Issues on security region search by online DSA," in *Power and Energy Society General Meeting, 2010 IEEE*, 2010, pp. 1-4.

[48] F.-X. Bouchez, B. Haut, L. Platbrood, and K. Karoui, "HPC for power systems in the framework of PEGASE project," in *Power and Energy Society General Meeting, 2012 IEEE*, 2012, pp. 1-8.

[49] I. M. Dudurych*, et al.*, "Tools for handling high amounts of wind generation in National Control Centre in Ireland," in *Power and Energy Society General Meeting, 2012 IEEE*, 2012, pp. 1-8.

[50] L. A. Wehenkel, *Automatic learning techniques in power systems*: Springer, 1998.

[51] Y. Mansour*, et al.*, "Large scale dynamic security screening and ranking using neural networks," *IEEE Transactions on Power Systems,* vol. 12, pp. 954-960, 1997.

[52] A. Machias, "Application of a pattern recognition method to the transient stability problem," *Electric power systems research,* vol. 21, pp. 155-160, 1991.

[53] P. M. Anderson and A. Bose, "A probabilistic approach to power system stability analysis," *IEEE Transactions on Power Apparatus and Systems,* pp. 2430-2439, 1983.

[54] C. Liu*, et al.*, "A systematic approach for dynamic security assessment and the corresponding preventive control scheme based on decision trees," *IEEE Transactions on Power Systems,* vol. 29, pp. 717-730, 2014.

[55] 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 Transactions on Power Systems,* vol. 22, pp. 1935-1943, 2007.

[56] T. Guo and J. V. Milanovic, "Identification of power system dynamic signature using hierarchical clustering," in *PES General Meeting| Conference & Exposition, 2014 IEEE*, 2014, pp. 1-5.

[57] J. Lopes and M. H. Vasconcelos, "On-line dynamic security assessment based on kernel regression trees," in *Power Engineering Society Winter Meeting, 2000. IEEE*, 2000, pp. 1075-1080.

[58] R. Tiako, D. Jayaweera, and S. Islam, "Real-time dynamic security assessment of power systems with large amount of wind power using Case-Based Reasoning methodology," in *Power and Energy Society General Meeting, 2012 IEEE*, 2012, pp. 1-7.

[59] Y. Xu, Z. Y. Dong, K. Meng, R. Zhang, and K. P. Wong, "Real-time transient stability assessment model using extreme learning machine," *IET generation, transmission & distribution,* vol. 5, pp. 314-322, 2011.

[60] Y. Xu, Z. Y. Dong, Z. Xu, K. Meng, and K. P. Wong, "An intelligent dynamic security assessment framework for power systems with wind power," *IEEE Transactions on Industrial Informatics,* vol. 8, pp. 995-1003, 2012.

[61] 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 Transactions on Power Systems,* vol. 27, pp. 1253-1263, 2012.

[62] R. C. Borges Hink*, et al.*, "Machine learning for power system disturbance and cyber-attack discrimination," in *Resilient Control Systems (ISRCS), 2014 7th International Symposium on*, 2014, pp. 1-8.

[63] F. Luo*, et al.*, "Advanced Pattern Discovery-based Fuzzy Classification Method for Power System Dynamic Security Assessment," *IEEE Transactions on Industrial Informatics,* vol. 11, pp. 416-426, 2015.

[64] E. Voumvoulakis, A. Gavoyiannis, and N. Hatziargyriou, "Application of machine learning on power system dynamic security assessment," in *Intelligent Systems Applications to Power Systems, 2007. ISAP 2007. International Conference on*, 2007, pp. 1-6.

[65] E. S. Karapidakis, "Machine learning for frequency estimation of power systems," *Applied Soft Computing,* vol. 7, pp. 105-114, 2007.

[66] K. S. Swarup, "Artificial neural network using pattern recognition for security assessment and analysis," *Neurocomputing,* vol. 71, pp. 983-998, 2008.

[67] A. M. Haidar, A. Mohamed, A. Hussain, and N. Jaalam, "Artificial Intelligence application to Malaysian electrical powersystem," *Expert Systems with Applications,* vol. 37, pp. 5023-5031, 2010.

[68] A. M. Haidar, A. Mohamed, and F. Milano, "A computational intelligence-based suite for vulnerability assessment of electrical power systems," *Simulation Modelling Practice and Theory,* vol. 18, pp. 533-546, 2010.

[69] I. Kamwa, S. Samantaray, and G. Joos, "Catastrophe predictors from ensemble decision-tree learning of wide-area severity indices," *IEEE Transactions on Smart Grid,* vol. 1, pp. 144-158, 2010.

[70] C. F. Kucuktezan and V. I. Genc, "Dynamic security assessment of a power system based on probabilistic neural networks," in *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, 2010, pp. 1-6.

[71] M. He, J. Zhang, and V. Vittal, "A data mining framework for online dynamic security assessment: Decision trees, boosting, and complexity analysis," in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, 2012, pp. 1-8.

[72] A. Ng. *Machine Learning*. Available: http://cs229.stanford.edu/

[73] *fminunc:Find minimum of unconstrained multivariable function (MATLAB)*. Available: http://www.mathworks.com/help/optim/ug/fminunc.html

[74] D. Barber, *Bayesian reasoning and machine learning*: Cambridge University Press, 2012.

[75] A. Smola and S. Vishwanathan, *Introduction to Machine Learning*: Cambridge University Press, Cambridge, United Kingdom, 2008.

[76] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research,* vol. 1, pp. 211-244, 2001.

[77] K. P. Murphy, *Machine learning: a probabilistic perspective*. Cambridge, Massachusetts: MIT press, 2012.

[78] C. M. Bishop, *Pattern recognition and machine learning*: Springer, 2006.

[79] S. I. R. Costa, S. A. Santos, and J. E. Strapasson, "Fisher information distance: A geometrical reading," *Discrete Applied Mathematics,* vol. 197, pp. 59-69, 2015.

[80] *PMTK: Probabilistic Modeling Toolkit for Matlab/Octave*. Available: https://github.com/probml/pmtk3

VITA

Navin Kodange Shenoy

Candidate for the Degree of

Doctor of Philosophy

Thesis: AN APPROACH TO ASSESS THE RESILIENCY OF ELECTRIC POWER GRIDS

Major Field: Electrical Engineering

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2015.

Completed the requirements for the Master of Science in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma/USA in 2010.

Completed the requirements for the Bachelor of Engineering in Electrical Engineering at University of Mumbai, Mumbai, Maharashtra/India in 2006.

Experience:

Research Assistant at Engineering Energy Laboratory, School of Electrical and Computer Engineering, Oklahoma State University from January 2009 to December, 2015.

Teaching Assistant at School of Electrical and Computer Engineering, Oklahoma State University during Spring 2013 and Spring 2014.