2015

Data Mining Framework for Monitoring Attacks In Power Systems

Prem T. Alluri

Data Mining Framework for Monitoring Attacks In Power Systems

By

Prem T Alluri

Thesis submitted to the

Benjamin M. Statler College of Engineering and Mineral Resources

at West Virginia University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Dr. Sarika Khushalani Solanki, Ph.D., Chair

Dr. Jignesh Solanki, Ph.D.

Prof. Muhammed A. Choudhry, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia

2015

Keywords: Attack location, attack detection, clustering streaming data, online divisive-agglomerative clustering, symbolic aggregate approximation (SAX), Data mining, Attack severity, Attack criticality.

ABSTRACT

Data Mining Framework for Monitoring Attacks in Power Systems

Prem T Alluri

Vast deployment of Wide Area Measurement Systems (WAMS) has facilitated in increased understanding and intelligent management of the current complex power systems. Phasor Measurement Units (PMU's), being the integral part of WAMS transmit high quality system information to the control centers every second. With the North American Synchro Phasor Initiative (NAPSI), the number of PMUs deployed across the system has been growing rapidly. With this increase in the number of PMU units, the amount of data accumulated is also growing in a tremendous manner. This increase in the data necessitates the use of sophisticated data processing, data reduction, data analysis and data mining techniques. WAMS is also closely associated with the information and communication technologies that are capable of implementing intelligent protection and control actions in order to improve the reliability and efficiency of the existing power systems. Along with the myriad of advantages that these measurements systems, informational and communication technologies bring, they also lead to a close synergy between heterogeneous physical and cyber components which unlocked access points for easy cyber intrusions. This easy access has resulted in various cyber attacks on control equipment consequently increasing the vulnerability of the power systems.

This research proposes a data mining based methodology that is capable of identifying attacks in the system using the real time data. The proposed methodology employs an online clustering technique to monitor only limited number of measuring units (PMU's) deployed across the system. Two different classification algorithms are implemented to detect the occurrence of attacks along with its location. This research also proposes a methodology to differentiate physical attacks with malicious data attacks and declare attack severity and criticality. The proposed methodology is implemented on IEEE 24 Bus reliability Test System using data generated for attacks at different locations, under different system topologies and operating conditions. Different cross validation studies are performed to determine all the user defined variables involved in data mining studies. The performance of the proposed methodology is completely analyzed and results are demonstrated. Finally the strengths and limitations of the proposed approach are discussed.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First, I would like to sincerely thank my advisor, Dr. Sarika Khushalani Solanki, for her invaluable guidance and encouragement. Next, I would like to thank my second committee member Dr. Jignesh Solanki for his constant motivation and support. I would also like to thank my other committee member Dr. Muhammed A. Choudhry, whose feedback and reviews helped me improve the quality of this thesis.

I would like to specially thank the 'Three Ladies in my life', my two sisters and my mother. Their love and affection have helped me overcome the toughest of challenges during these two and half hard years. Also I would also like to thank my dad for his support throughout my studies. Lastly, I am thankful to all my friends who made my stay at the West Virginia University a memorable and valuable experience.

I dedicate this thesis to the memory of my grandmother, for bestowing her abundant love upon me.

1 Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1. Background

Deregulation, Decentralization has improved the reliability and efficiency of the existing power industry with reduction in its operational costs. These changes have resulted in deployment of many improved and advanced technologies like WAMS, enhanced Information Management techniques and Improved Communication Protocols. Since its introduction in 1980, WAMS has facilitated in increased understanding and management of the current complex power systems. WAMS constitute of advanced measurement technologies like PMU's, informational tools and operational infrastructures that enable operators with real time knowledge of the system. Each synchrophasor based PMU transmits time-stamped power system operational parameters with high precision typically at a rate of 30 samples per second. Several PMU units were installed over the years and at present there are about thousands of PMU units streaming the current operational parameters of the system in which they were deployed. Thus in a complex and critical power system with large number of PMU units, huge volumes of nonlinear system observations are streamed at each instant. These fast streamed system parameters from remote locations of the grid are stored in a Phasor Data Concentrator. Proper analysis of these fast streaming system parameters can help in extracting grid behavioral pattern time to time, protecting critical infrastructure, providing situational awareness and also enable in real time monitoring and visualization.

Recently many trend setting research works have been proposed in this area of phasor data based real time event monitoring and disturbance detection. These algorithms are capable of extracting information and knowledge from the behavioral patterns of all the units for different disturbances in the system and are assembled such that they strive to detect the disturbances in the system and provide timely control actions. Related trend setting works in this area of data analysis, knowledge discovery based disturbance detection are as follows. References [1] and [2] provides techniques to achieve data reduction/compression and feature identification in huge databases and propose a classification algorithm in to detect voltage stability disturbances. References [3] and [4] uses data mining techniques to derive security criteria and identify the critical elements in a stressed system based on their responses in different operational conditions. References [5] and [6] employed clustering techniques to recognize coherency between the generators in the system and deduced models based on the recognized coherent groups to evaluate the disturbances in the system. Though the operational efficiency of these proposed algorithms matches with the existing industry disturbance detection practices, they cannot be implemented in real time because of the high operational speed requirements. This speed criterion is not met by the referred data mining based formulations. Currently new studies are ongoing in this direction of high speed event/ disturbance detection using phasor data.

Apart from these disturbance detection strategies from WAMS, the Information and Communication Protocols (ICP) mentioned earlier are capable of implementing intelligent protection and control actions in order to improve the reliability of the power systems. The combined coordination of both WAMS and ICP is currently providing real time control and protecting the electric power systems from all the naturally occurring disturbances, outages, dynamic events and also higher order contingencies. However, apart from these mentioned advantages, ICP has also brought a close synergy between heterogeneous physical and cyber components which unlocked access points for easy cyber intrusions [7]. These access points are acting as resources for the attackers and allowing them to inject intelligent malicious commands that are capable of incurring huge losses to the system. Hence the operators who monitor, assess, and react to disturbances must now consider the new possibility that the system is under a cyber-attack. These attacks can either be simple ones that are capable of isolating single or multiple elements from the system or they can be planned and coordinated attacks that are capable of causing large system failures. They can be very hard to detect as the major objective of the attacker is to deceive the system operators by making these attacks look like naturally occurring disturbances. Reference [8] presents cases where intelligent attacks pass through system state estimation process without being detected by operator. Whereas reference [9] shows that the behavioral patterns of measuring units for attacks at different locations in the system are exactly similar to those with genuine disturbances (faults) occurring at those locations. The motivation of the attacker to disguise the attack makes the problem of detecting an attack in the system more different and the potential event detection methodologies mentioned in the previous paragraph will not work accurately for this attack detection problem. This shows us the great need for intelligent techniques that are capable of monitoring and detecting attacks in the system using the enormous data from the measuring units and differentiating them from real events.

Along with attack detection, if the effects of the attack on the system are known in a timely manner, this will give the operator the needed knowledge about the consequences that the system is going to face because of the attack. Also as we know that the major objective of the attacker will be either to damage large parts of the system or incur huge amount of losses. An attacker can easily achieve his objective if he/she attacks critical or important elements in the system which will initiate several other contingencies and ultimately result in a large failure. The Northwest Blackout of 2003 which had affected 45 million people for 2 continuous days had initiated from a failure of single alarm system [10]. Hence along with information of how severe the attacks would be on different elements in the system, knowledge about the post attack critical elements in the system that are to be properly monitored and protected so that the attack will not propagate will be very valuable and helpful for the operator [11]. This need motivates the work in this thesis. There remains a great deal of opportunity to advance the state of the art, especially for developing suitable techniques to perform automated analysis and attack detection based on the streamed real time data from the PMU's . This work is presented with the anticipation of encouraging future extensions, especially with respect to the timely disturbance and attack identification using information from patterns in power system data.

1.2. Data Mining Background

Nowadays, terms such as 'Data Outburst' and 'Big data' are being used more frequently to describe the situation of power systems with respect to data. The work in this thesis spans several multi-disciplinary research areas, most of them are related to data mining. Data Mining (DM) is indeed a broad subject [12], so it is important to provide some clarification. This thesis mainly employs the DM concepts of Discretization, Data Clustering and Classification. In this section we introduce some of the basic topics of DM, as well as background on what conventional DM tools can provide. First, Section 1.2.1 describes knowledge discovery in large databases and various steps involved it. Then Section 1.2.2 describes preprocessing and discretization of data. Section 1.2.3 provides some essential background on the classification techniques. Section 1.2.4 gives brief description about data clustering procedure. Then, an Overview of State of Art DM techniques that are applied to solve power system problems is provided in Section 1.2.5

1.2.1 Knowledge Discovery in Databases (KDD)

The desire and need for information has led to the development of expensive infrastructure and intelligent equipment that can generate and collect massive amounts of data [13]. However, our ability to analyze this data collected and convert it into meaningful information is impeded by the size and complexity of the stored database. In fact, the sheer size of the data makes human analysis unsustainable in many instances, negating the effort spent in collecting the data [14].

The process of information or 'knowledge' retrieval from these massive databases using various tools is generally referred as Knowledge Discovery in Databases (KDD). KDD can be exactly defined as the process of mapping low-level data which are typically too voluminous for easy understanding and digestion into other forms that might be more compact, more abstract and more useful [15]. Data, in its raw form, is simply a collection of elements, from which little knowledge can be gained. With the application of data discovery techniques the value of the data is significantly improved. A variety of methods are available to assist in extracting patterns that when interpreted provide valuable, possibly previously unknown, insight into the stored data. This information can be predictive or descriptive in nature. At an abstract level, the KDD field is concerned with the development of methods and techniques for making sense of data and hence the basic problem addressed by the KDD process is the application of specific data-mining, data reduction and preprocessing methods for pattern discovery and information extraction.

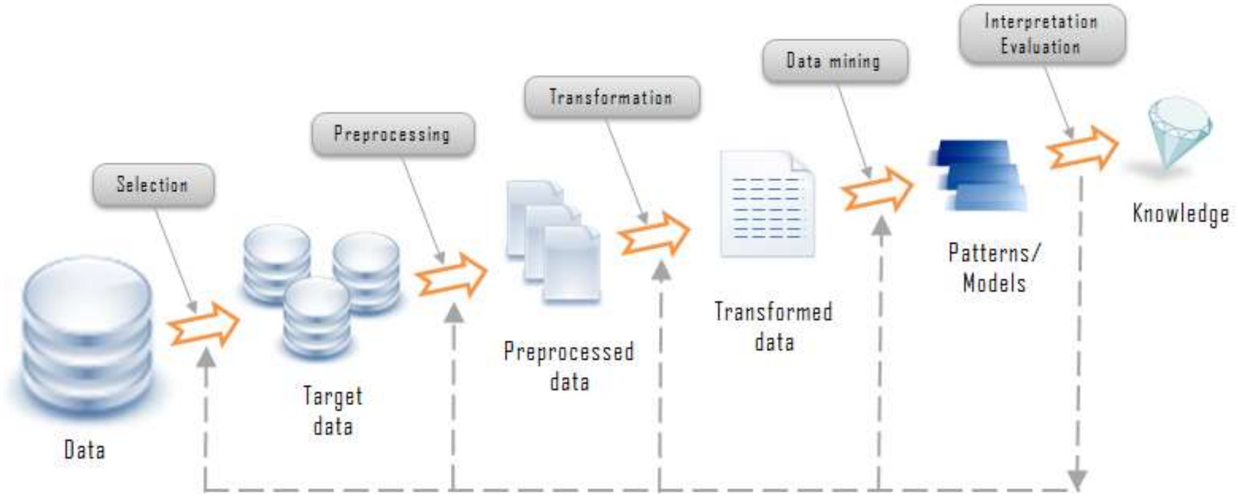


Figure 1: Overview of Steps that comprise KDD process [15]

Brief description of steps involved in KDD process is given here;

a) Data Collection: This step involves the collection of data for different test cases and storing it in proper formats so that the stored data can be easily assessable by the next KDD procedures. Prior knowledge about the data is necessary in this step as this knowledge can help in proper selection of the dataset.

b) Selection: This step involves selecting the target dataset from the entire data collected on which further testing is to be performed.

c) Preprocessing: involves cleaning of the target dataset that is selected. Cleaning involves removing noise or outliers in the dataset, formulating and choosing strategies to deal with the missing values and collecting necessary information needed for next steps.

e) Transformation: involves transformation of numerical or alphabetical dataset from the preprocessing stage into a corrected, ordered, and simplified form. The basic concept is the reduction of multitudinous amounts of data down to the meaningful parts [16].

f) Data Mining: Data mining involves discovering patterns in these reduced datasets using methods like artificial intelligence, machine learning, statistics and database systems. The overall goal of data mining process is to extract important information from these processed datasets and transform it into understandable structure for further use. Data Mining generally involves five common classes of tasks. Brief description about each step is as follows;

1. **Anomaly or Outlier Detection:** involves identification of cases, records, events and instances in the dataset that do not behave in the similar manner as that of the entire dataset. Outliers deviate from the normal behavior of other samples and records in that dataset. Three broad categories of anomaly detection generally exist they are Unsupervised anomaly detection, Supervised anomaly detection and Semi-Supervised anomaly detection.
2. **Associate rule learning:** involves the process of identifying associations or rules between the variables (or attributes) in the dataset.
3. **Clustering:** is process of grouping similar objects in the dataset based on the similarity of distance between them. More explanation about clustering is presented in upcoming sections.
4. **Classification:** involves predicting the dependent variable or class variable of the dataset based on the model that is generated using the information gained from the dataset.
5. **Regression:** involves the process of finding a function that fits on the dataset.

g) Interpretation and Evaluation: The final step of knowledge discovery process is to interpret the model presented by the data mining algorithms and verify the patterns produced on a wider data set. Not all patterns found by the data mining algorithms are necessarily valid. It is routine for the DM algorithms to find associations and rules in the training set which are not present in the general data set [17]. This is called overfitting. To overcome this, the evaluation uses a new data set called as test set, which is not seen by data mining algorithm in the previous stage. The learned patterns are applied to this test set, and the resulting output is compared to the desired output.

1.2.2 Data Discretization

Many DM algorithms prefer learning in nominal feature spaces [18]. However, the phasor data obtained from the PMU units are high frequency continuous samples where such Data Mining algorithms could not be applied unless the continuous features are first discretized. Discretization involves a process of converting continuous samples of data into discrete counterparts. Brief review of the data discretization algorithms is presented here; Uniform continuous binning procedure which transforms the data into discretized bins is one of the famous discretization techniques in DM and Machine Learning communities. However the major disadvantage of this procedure is that it does not perform any careful study on how the discretization affects the performance of the DM learner. Algorithms like decision trees do variable discretization where the continuous data samples are discretized during the learning process. Binary discretization is also one of the simplest way of discretizing data along with continuous binning and it reduces the continuous data into ranges between binary numbers [19]. The other reason for discretizing continuous data aside from the one mentioned above is the speed of the Data Mining increases in a drastic way while learning on discretized data samples [19].

Time Series Discretization is slightly different form discretizing steady state databases [20]. Here the data used in this thesis is time series data. Some of the State of Art time series discretization techniques are reviewed here. Reference [21] represents time series values as a multi connected graphs and in the discretization state, similar time series are grouped into a graphical model. Reference [22] uses Discrete Fourier Transform (DFT) where time series is converted into ordered frequencies. Reference [23] introduces Discrete Wavelet Transform where continuous variables are converted into square integral functions. Reference [24] proposes Single Valued Decomposition based data discretization where time series data is discretized by converting it into singular values.

1.2.3 Classification

Classification assigns each example or instance in the data set to target categories or classes. The goal of classification is to accurately predict the target class for each instance in the data. For example, let the rotor angles of different generators in a system for a particular time instant is given by X={x1, x2,x3,x4….xN}, where N is the number of generators present. For these particular rotor angle values of each generator, the current transient stability of the system may either be Y={Stable or Unstable}. Here the responses of the measuring units xi, i ϵ N are the independent variables which are referred as attributes. Whereas the Y, the system operational state/condition is a dependent variable or class variable as it depends on the attributes. The input for a classification algorithm is a training dataset with different number instances similar to X and a dependent variable vector which conveys for each instance in X, the stability of the system. The row length of X and Y will be equal. Now for input training dataset $\{X,Y\}_{size,\ N+1}$, where size is the number of instances or different cases of generator responses, the output of the data mining classification algorithms is a model that conveys a relationship between the attributes and the class variable Y. So now if a test instance of generator responses for which the class variable Y is now known is sent through this model, the model will predict the class variable.

Over the years, several classification algorithms are proposed. These classification algorithms are also referred as *learners* as they learn on the databases. Brief description of some of the very famous classification algorithms is presented here; Naive Bayes classifiers [25] are a family of simple probabilistic classifiers based on applying Bayes' theorem with naive independence assumptions between the features. Naïve Bayes is one of the simplest and base line classifier and has been in use from 1960's. Even today the results of Naïve Bayes classifiers are rigorously used as baseline performance standards. Support Vector Machine (SVM) [26] model represents the instances in the data set as points in space and these points are mapped so that the data points of the separate class variables are divided by a clear gap that is as wide as possible. Test dataset instances are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. SVM is generally a linear classification algorithm but it can also efficiently perform for non-linear classification problems using a kernel trick, which is implicitly mapping their inputs into high-dimensional feature spaces [26]. Decision tree learning uses a decision tree as a predictive model which maps the instances in the training database to conclusions about the class variables. It is one of the predictive modelling approaches used in statistics, data mining and machine learning. In these tree structures, leaves represent class labels and branches represent conjunctions of attributes that lead to those class labels. This thesis uses two classification algorithms to identify disturbances in the system and detect attacks. A third classifier is also used to discriminate physical attacks with malicious data. The first classifier is a Random Forest algorithm which is an ensemble learning technique that operates by constructing a multitude of decision trees on the training dataset and outputting the class variable that is the mode of the classes output by individual trees. The second classification algorithm used is k Nearest Neighbor Classifier (kNN). k-NN is an instance based non-parametric method and the input to it are the k closest training examples in the feature space and its output is the most common class among its k nearest neighbors identified. Being the simplest of all machine learning algorithms, kNN uses a user defined similarity measure to identify its neighbors from the training database. The third classifier employs Bayes technique mentioned above to discriminate attacks. More detailed information about kNN, Random Forest Classifier and Naïve Bayes Classifier is provided in next chapter.

1.2.4 Clustering

Clustering is a process of grouping objects in a dataset into small groups (clusters) such that objects in the same cluster are similar to each other when compared to the ones in the other clusters. Clustering is an unsupervised form of learning. It means the vector Y in the example mentioned in previous section is also treated as a normal independent attribute during clustering. The proposed work employs an online data stream clustering where new data will be continuously updated at cluster nodes and the clustering structure changes with the change in data. The outputs of the clustering stage in this thesis are clusters of PMU units such that PMU's in the same cluster behave in a similar way. Clustering helps in reducing the complexity of the problem. For example let us consider that there are 1000 measuring units deployed in a system and each unit will be streaming real time phasor data of its variables at each instant of time. It will be tiresome to monitor all these 1000 units at the same time and make decisions about system conditions based on their responses. Instead, from the 1000 units if there is way to choose 200 units and monitor them such that these 200 units can equivalently demonstrate or represent the behavior of all 1000 units, then this will ease the process of monitoring and also improve the operational time as we are not testing on 1000 units. The left side plot in Fig.2 shows 200 data points before clustering whereas the right side plot shows these 200 data points are grouped into three clusters based on distance between each data point.

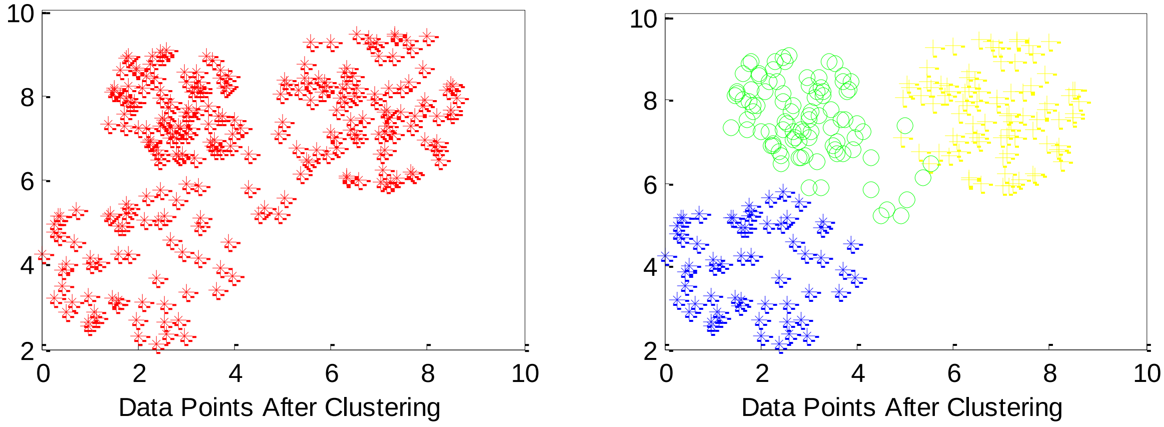


Figure 2: Clustering 200 Data Points into three clusters

Several clustering methods that are being in use today, brief outline of some of the algorithms are as follows; Hierarchical clustering methodology builds models based on the distance connectivity. Hierarchical clustering is a more usable form of clustering as it does not require a user defined variable that defines into how many clusters the objects in the dataset has to be clustered or grouped. It divides objects based on two strategies called Agglomerative (bottom up) and Divisive (top down). k-means clustering which is also one of the famous clustering methodology aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean or centroid, serving as a prototype of the cluster. The efficiency of the k-means clustering strictly depends upon the user defined parameter k, the number of clusters and also the distance measures chosen to compute the means of the clusters. In distribution based model clustering [27], clusters are built based on the statistical distributions of attributes in the dataset. Multivariate normal distribution clustering and expectation maximization clustering are two famously used distribution clustering techniques.

1.2.5 Overview of Data Mining Techniques applied for Power System Problem Solving

Data Mining has been in use in the Power System Industry since very long time. Whether it be for load forecasting, electricity price forecasting, faults classification and many other event detection problems, these intelligent data mining algorithms are making use of the enormous data generated in power industry and assisting it to operate in a more efficient and economical way. Following is a brief description of some of the most efficient data mining based formulations used for solving real time power systems problems; Reference [28] describes a methodology where decision tree algorithms are used to classify and characterize different types of consumers connected to large utility system. Reference [29] uses clustering and associative rule-learning techniques to profile different loads connected to a distribution system. Reference [30] also uses decision tree algorithm to identify the constraints involved in the economic dispatch problem. Reference [31] presents classification and regression techniques to detect the power quality issues faced for different faults in the system.

Several classification techniques are formulated to assess the transient as well as the dynamic security of the system. Reference [32] models CART decision trees to assess the dynamic security of a large system based on the frequency samples for different system operating conditions. Reference [33] uses an assemble classifier model based on decision trees for the transient stability out of step prediction problem. Reference [34] uses kernel regression trees to assess the online dynamic security of a large power system. Reference [35] uses boosting algorithm to improve the accuracy of different decision trees that are built by considering random attributes in the system using an optimization framework. Reference [36] proposes a decision tree methodology for on-line preventive dynamic security assessment of isolated power systems. The computation time taken by this algorithm is very less and is proposed to be implemented in real time. Reference [37] uses sliding surface enhanced fuzzy control technique to enhance the performance of state estimation techniques for bad or anomalous data. Reference [38] uses a response based correlation technique to identify most sensitive substation in face of disturbances. Also, two indices are formulated to compute the severity of load reduction for different disturbances. Reference [39] implements improved bayesian techniques to detect and locate various single phase faults in the system. State of art research work in the area of data mining based event/attack detection and vulnerability assessment that motivated the research in this thesis is presented in Chapter II. Brief background about Attacks in power systems is given in the next section.

1.3 Attacks in Power System

With the introduction of next generation advanced Power Systems in the form of the Smart Grid, the efficiency and reliability of existing system has been greatly enhanced. Smart gird is currently equipped with advanced computing and two way communication technologies that enable them the capabilities of distributed intelligence, situational awareness, demand response and post event action discretion. However along with the silent features of the Smart Grid, cyber security emerges to be a critical issue because of its heavy dependence on communication networks that connect millions of electronic devices. This in fact increases the risk of compromising the reliability and security of power system operations, which, nonetheless, is the ultimate objective of the Smart Grid [40]. Brief description about different types of cyber attacks in Smart grid is presented next.

1.3.1 Classes of Attacks

a) Denial of Cooperative Operation (DoS): DoS attack is a severe threat to the wide area monitoring and communication systems. In the DoS attack, the objective of the attacker is to prevent the monitoring units from sending data back to the control center, which will result in the system instability, failure or even a regional blackout. One famous category of DoS attack is the jamming attack in the physical layer of communication networks [41]. In jamming attack, a jammer emits power over the communication channel to interfere the ongoing data transmission. Such a jamming in the physical layer is effective particularly in wireless communication systems, which is a promising candidate for the communication infrastructure in smart grid. Fig.3 shows a jamming attack on measurement systems.

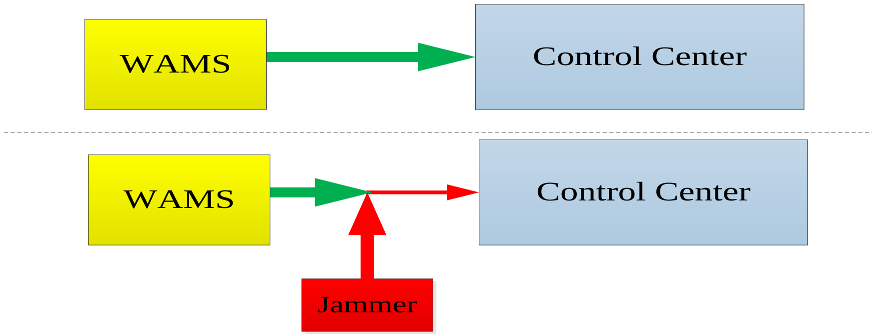


Figure 3: DoS Jamming Attack

b) Data Injection Attacks: This division of attacks poses great threat to the system. Here the adversary injects falsified or malicious data into the system through compromised meters, hacking communication networks between meters and SCADA systems, breaking into the SCADA system through a control center office LAN, breaking home area networks and neighboring area networks to compromise meters. The objective of the attackers is to disrupt genuine operation of the system. The attacks considered in this research are trip injection attacks where the attacker compromises the communication protocols and is assessable of sending a trip command to relay and circuit breakers either remotely or from the system itself.

c) Physical Attacks: Physical attacks include damaging critical infrastructure of the system with a motive to incur immediate losses. These include relay deliberate shut down, damaging high transfer capability lines, connection changes, UTC vandalism etc. Replay attacks also come into the category of physical attacks where a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried by the adversary who intercepts the data and retransmits it.

d) Man In The Middle Attacks: Man-in-middle (MITM) is a division of cyber attacks in which the attacker records the data packets from the network, modifies them, and inserts them back into the network. Under the right conditions, an attacker could insert a MITM device, capturing all outside connections.

e) *Desynchronization attacks:* The control algorithms of automated operation which are time dependent are manipulated in these attacks.

1.3.2 Cyber Attacks - Power Systems Background

Brief description of previous research works in the areas of monitoring, detecting and identifying different types of cyber attacks, except for command injection attacks are presented here. Information about research in identifying command injection attacks is presented in next chapter. Reference [42] presents an optimal data injection attack detection strategy using spatial and temporal based detection schemes to accurately identify the attacks. Least-effort attack model which enables the attacker to find the optimal set of meter measurements given the number of state variables is presented and efficient algorithms are developed to find these sets of meters. The proposed algorithm is implemented and tested on different IEEE test systems and demonstrated its attack detection superiority over random search approach. Reference [43] proposes Adaptive Partitioning State Estimation (APSE) methodology to detect data injection attacks in the system. APSE algorithm transforms a power system into a weighted undirected graph and this graph is then partitioned into a certain number of subgraphs exploiting the L bounded Graph Partition Method. Reference [44] presents a frame work for data injection attacks on the state estimator through SCADA with a perturbed or outdated model of the electric grid. Linear and Non-Linear state estimators are considered and it is shown that the more accurate the information that the attacker possesses, the greater the threat of an attack. Then Chi-squares method is applied on each subsystem to detect the bad data in that particular subsystem. These subsystems are thoroughly updated to narrow the suspicious region of bad data. Reference [45] uses Principal Component Analysis to identify and detect bad samples of measurements from real time outputs of measuring units before they are sent into the state estimation process. The proposed research is implemented on an IEEE 14 Bus system and results show how single and multiple samples of bad data are cleaned from raw measurements. Reference [46] presents adversarial strategies and countermeasures for malicious data attacks. A polynomial-time algorithm is implemented to find small but highly damaging unobservable attacks and minimum residue energy heuristic is also developed to detect worst attacks on the system. Reference [47] presents two NN based intrusion detection algorithms that are capable of analyzing huge power system data using the specific window based feature extraction techniques developed. The two proposed NN algorithms are Error Back Propagation and Levenberg Marquardt for modelling normal system behavior. Reference [48] presents a SCADA specific cyber-security test-bed which contains SCADA software and communication infrastructure. This test-bed is used to investigate an Address Resolution Protocol (ARP) spoofing based MITM attacks. Reference [49] demonstrates the impact of a data integrity attack on Automatic Generation Control (AGC), power system frequency and electricity market operation and develops an anomaly detection model to detect attacks on AGC. Results show that the proposed algorithm is capable of detecting scaling and ramp attacks with low false positive and negative rates. Reference [50] formulates various techniques to detect and mitigate data injection and MITM attacks in the system.

1.4 Problem Statement

The primary objective of this thesis is to formulate a data mining based methodology that is capable of identifying and locating remote trip injection attacks by monitoring real time streaming power system data from limited number of measuring units deployed across the system. The output of the proposed real time methodology is for the current instant of time, a prediction stating if there is a disturbance in the system and if the first prediction comes out to be true, a second prediction stating if the disturbance identified is originated from an attack in the system.

Attack Severity and Attack Criticality indices are developed and risk analysis is performed on the test system considered for attacks at different locations. This study is to provide situational awareness and the needed discretion for the operator to act properly once an attack is detected in real time. Attack Severity index presents the severity of an attack on the system based on the post attack changes in system properties. Attack criticality for all the attacks considered, finds critical elements in the post attack system, that are to be protected in preventing the attack in being a failure. Finally a prediction based detection strategy is proposed to detect and differentiate malicious data attacks that disguise as physical attacks in the system

Careful cross validation based assessment is done for choosing all the algorithms and user defined variables in the proposed methodology.

Several assumptions are made throughout this research which are explained in detail in the following chapters but few of most common assumptions are presented below,

- Measuring units are assumed at different locations of the system and are capable of streaming their operational parameters with high frequency.
- The attacker is assumed to be capable of compromising all the security protocols and is capable of tripping any line in the system.
- In Offline Attack Severity and Criticality assessment stage (given in Chapter 3), the knowledge about system topology after N-k events or attacks, complete description of the attack/events occurred and post attack/event system operating conditions are assumed to be known.

1.5 Outline

The outline of the remaining chapters is given in this section.

Chapter 2 is a comprehensive literature review and is divided into two sections. In the first section, literature review of the state of art offline, online event/ fault detection, location identification and analysis strategies that uses DM algorithms is given. The second section constitutes literature review about relay trip injection attacks, mitigation strategies and attack risk and vulnerability assessment techniques.

Chapter 3 gives an overview of the proposed methodology of detecting an attack in real time, identifying its location, attack severity and criticality assessment. Brief description about malicious data attack detection is also presented here.

Chapter 4 gives a detailed description of the software packages used in this research. The advantages offered by the software to the relevant applications are also discussed.

Chapter 5 presents the test system considered, simulation results of each stage of the proposed methodology.

Finally, Chapter 6 lists conclusion of this study and analyses its strengths and weakness. Scope of future work is also presented in this chapter.

Chapter 2: LITERATURE REVIEW

2.1 Event Monitoring using Data Mining - Motivation

The main motivation for this work has been initiated from the formulation and analysis of reference [5] where events in the system are located using a hierarchical clustering technique. Reference [5] clusters different measuring units in the system based on their angle responses and identify cluster representative units which are centroids of original cluster. The detection of event location is performed based on an index proposed which computes the sudden change in responses of the cluster representative units. The cluster representative location with highest change is the location of the disturbance origin. The proposed methodology is implemented on Western Electricity Coordinating Council (WECC) test system and different zones are created based on the cluster representatives. One of the shortcomings of [5] is the selection of similarity measure for clustering has been done randomly without any proper analysis on different similarity measures. The second shortcoming is that with the change in system output parameters from different units in the system, the clusters should also change frequently and in order to do this the clustering techniques should be performed in online mode. Whereas in [5] the clustering is performed on the offline data collected and cluster representatives are selected from the clusters. [51] proposes a methodology to predict the transient stability of the system using CART decision tree technique. The input attributes for the algorithm are a database of steady state generator mechanical input powers, kinetic energy deviations, average accelerations, KW output at the time of fault clearing and fault duration times for faults at different locations of the system. Decision trees are built on this database with random selection of attributes and they are used to predict the transient stability of the system for a test database with new unseen values of these attributes. The system that is considered is a large Iranian national grid. The prediction accuracy of the decision tree is high only when the instances in the test set are close to the ones that are in the training database. The proposed decision tree algorithm is also compared with other classification algorithms and it showed comparatively better performance. [52] proposes a framework to assess the voltage security of a power system based on decision tree approach. The proposed methodology is implemented in the online mode and system current instant responses from the PMU units are thoroughly updated and the voltage security of the system is assessed. As mentioned earlier, with new data arriving at each instant, the clusters or decision trees build on the offline database needs to be updated, this process of updating is performed in this paper using a database and model update step where the constructed decision tree model is thoroughly updated whenever the prediction accuracy falls low. This is one of the first online methodologies with decent performance accuracy. Two new ideas including multiple optimal decision trees and corrective decision trees are also introduced to improve the accuracy of the proposed approach. Reference [53] introduces an online security assessment scheme based on decision tree approach. The objectives here are to determine both transient stability as well as dynamic security of the system using real time measurements from the PMU units. Exhaustive offline analysis is performed to obtain different training cases and due to this exhaustive training, the prediction of decision tree algorithm is very high in real time even with instances that are not previously seen by the algorithm. However, this framework also has a drawback as of [52] where only the decision tree built is updated with the change in data not the security constraints or critical attributes considered in the offline mode. Reference [54] uses Neural Networks (NN) with enhanced feature selection and extraction capabilities to assess the transient stability of the system. A time domain simulation is performed to obtain data for different system faults at varying locations and NN are initially trained on this data. Then in real time, the rotor angles of the generators are sent through this NN to determine the stability of the system. Results show that the number of input features and the data influence the time taken to train the NN for respective areas. Feature selection and feature extraction method employed in this work enabled in reducing the input features. Reference [55] presents a core vector machine (CVM) based algorithm for online voltage security assessment. CVM has been initially trained on data from different contingencies and decision tree based feature section algorithm has been employed to select important features form the training data and use it in the online stage to assess the voltage security of the system. The effectiveness of the proposed algorithm is investigated and the results are compared with other security assessment techniques. Results show that the proposed approach is more effective and takes minimal time when compared to other algorithms.

This thesis employs a discretization technique called Symbolic Aggregate Approximation (SAX) to convert the continuous samples of responses into discrete ones. SAX has been used previously to solve a couple of power system problems and a brief description about those works will follow next. Reference [56] uses SAX to discretize the data from the rooftop Photo Voltaic (PV) units in a real Australian distribution system and proposes an advanced analysis tool for the assessment of the data from the clustered PV's. The proposed tool intelligently searches the data that is effective for analysis and enables identifying hidden patterns and anomalies in a huge archived time series database. Further, the tool also employs a load flow algorithm demanding less computation effort and performs in-depth analysis of the identified data of interest. Reference [57] uses SAX to extract important features of electricity prices from a real time market data. Electricity price curves are initially discretized using SAX and then a k-means algorithm is employed to extract different features. Experimental results show that the data after discretization has improved output accuracy with the k-means classifier when compared to continuous data. These outstanding works mentioned till now are the primary basis of this thesis.

2.2 Command Injection Attacks in Power System

These are the sub category of data injection attacks where a trip-command is sent to the relays, circuit breaker or other protection equipment by compromising the security protocols. As modern relays support communication protocols such as IEC 61850 and are Internet protocol (IP) ready [40]. An attack on the relay communication infrastructure or a malicious change to the control logic could result in unscheduled tripping of critical equipment and distribution feeders by leaving several load segments unserved. The loss of 377 MW load in Salt River Project is due to tripping of 147 circuit breakers which are activated by a less integrated control command from load scheduling logic [58]. The Outage in Superdome New Orleans, during the second half of super bowl 2013 is due to the mal operation of a ZSD overcurrent relay [59]. These incidents show that with accessing these tripping circuits, the attacker can cause significant damage to the system by removing important elements from it. Brief description about some of the earlier research works in this area of trip injection attacks that significantly motivated our research is presented here.

The main support and motivation for going in this direction of protection equipment tripping is from research in [59], where a Cyber-to-Physical bridge (C2P) is developed which links cyber-attack vectors to resulting events in the Electric Power Grid (EPG). Along with C2P, the Reliability Impacts from Cyber Attack (RICA) provide a potential foundation for the quantitative evaluation of impacts caused by EPG to the system. The RICA method allows the calculation of degraded reliability caused by presumed cyber-attack and this analysis can be understood as a means to calculate the averaged measures for the ongoing value of good cyber security. Both C2P and RICA provide established pathways between cyber components and physical impacts that assist in constructing models that determine the severity of these effects. The work models cyber-attack in terms of unexpected outages to grid equipment and in this way, for a given probability of cyber-attack, the additional degradation to system reliability that may result can also be determined quantitatively. After simulation, the difference in reliability (with or without cyber-attack) is the average grid impact for a given attack probability.

Aurora generator Test conducted by Idaho National Laboratory (INL) in 2007 demonstrated that if the attacker can gain access to the relay that signals the circuit breaker connecting the diesel generator to the system, with rapid open and close commands to the circuit breaker by the attacker, the generator can be permanently damaged [60]. This vulnerability is a concern because most of the grid equipment communicates using Modbus and other legacy communications protocols that were designed without security concern. Their inability to support authentication, confidentiality, or replay protection means that any attacker who can communicate with the device can control it and use the Aurora

Vulnerability to destroy it [60]. This is a serious threat to the system, as the failure of even a single generator could cause widespread outages and possibly cascading failure of the entire power grid, like the one occurred in the Northeast blackout of 2003 [10]. Reference [61] presents results for the impact of a possible aurora attack on the system with an actual objective to see if the aurora attack is a myth or it's a reality. Research results show that on a moderately protected system, aurora attack has a devastating impact. In order to verify the system protection integrity and protect it from a possible attack the following regulations are to be implemented. The regulations include the tie points, protection logic through all the breaker connections are to be reviewed along with results of power generation and power flow to estimate the rate of change of frequency when a bus-tie breaker opens and optionally closes under load. Informed decisions and protection signals are to be made to determine if the generators are susceptible to attack. Reference [7] proposes framework for attack modeling using vulnerability of information and communication technologies (ICT) in the electric grid network. Concepts like discovery, access, feasibility, communication speed and detection threat are used along with graph theory to assess the vulnerability of ICT network. Aurora like event is generated by operating circuit breakers connecting the generator resources using a common attack vector based on cyber and physical access points. The proposed methodology is implemented on two test systems in real time using a Real Time Digital Simulator and SEL overcurrent relays as IED's. The results on these test cases present the impact of an integrated cyber physical attack on the system. Reference [62] presents six cyber attack scenarios in the SCADA system and a Bayesian attack graph model is used to evaluate the probabilities of successful cyber attacks which aim in tripping the circuit breakers in the system. A forced outage rate (FOR) model is proposed considering the frequencies of successful attacks on the generators and transmission lines. The loss of load probability values are estimated on an IEEE 79 Bus RTS for increased FOR values and the results show that the test system becomes less reliable as the frequency of successful attacks increases. Reference [63] uses defense graphs and Influence diagrams to model the security of a Wide Area Network (WAN) in a power communication system. The proposed methodology is also capable of managing uncertainties, both related to the efficacy of countermeasures and the actual posture of the supervisory control and data-acquisition system. Finally a model based Wide Area Network (WAN) attacks are also analyzed and the possible counter measures are explained. Finally in reference [64] the problem of selecting the small subsets of measurements that can be made immune to make the whole system immune from data injection attack is solved. Since this problem becomes really complex due to the large system size, a fast greedy algorithm is used for placing secured PMUs.

Chapter 3: Formulation of the Proposed Methodology

3.1 Overview of the Proposed Attack Detection Methodology

The proposed methodology constitutes of three stages, they are

a) Offline Analysis and Training Database Construction Stage.

b) Online Model Construction, Verification and Validation.

c) Real Time prediction stage.

A detailed pictorial representation of the proposed approach is presented in Fig. 4;

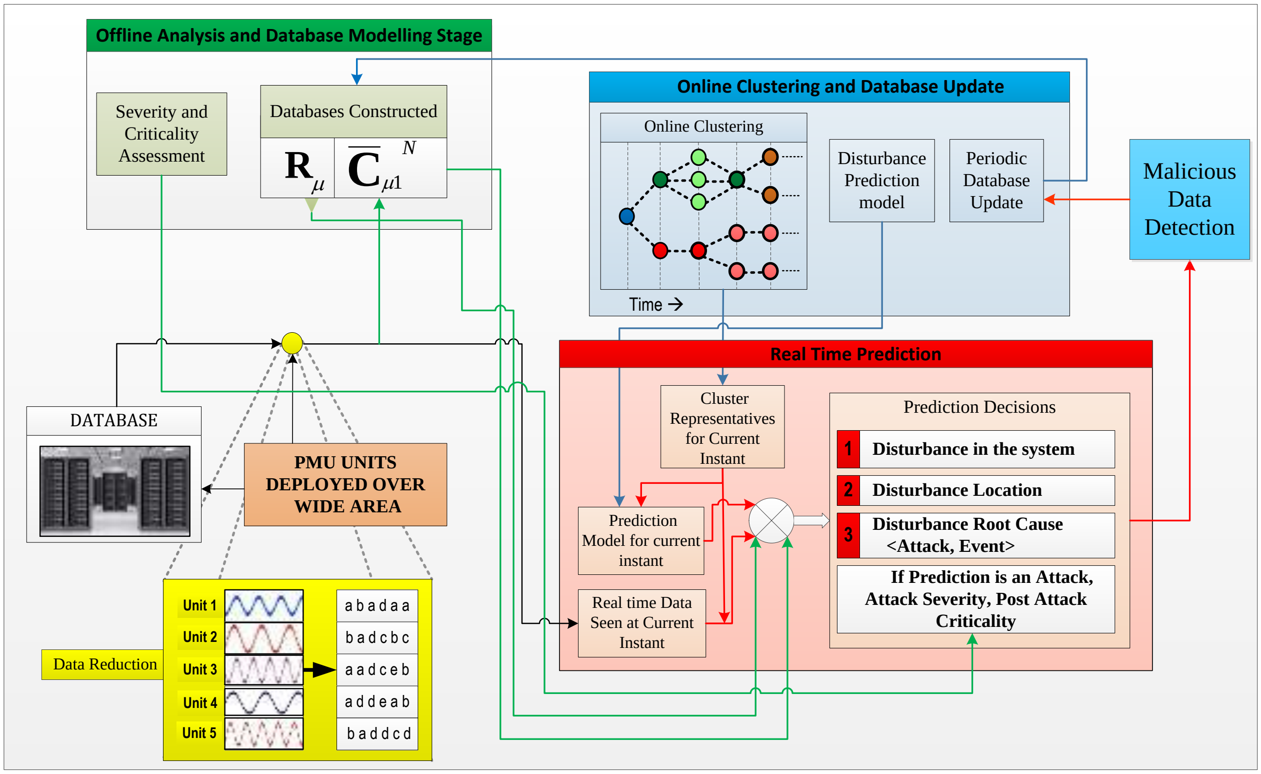

Offline Analysis and Database Modelling Stage
Severity and Criticality Assessment
Databases Constructed
R μ
C̄ μ1 N
Online Clustering and Database Update
Online Clustering
Time →
Disturbance Prediction model
Periodic Database Update
Malicious Data Detection
Real Time Prediction
Cluster Representatives for Current Instant
Prediction Decisions
1 Disturbance in the system
2 Disturbance Location
3 Disturbance Root Cause <Attack, Event>
If Prediction is an Attack, Attack Severity, Post Attack Criticality
Prediction Model for current instant
Real time Data Seen at Current Instant
DATABASE
PMU UNITS DEPLOYED OVER WIDE AREA
Data Reduction
Unit 1
Unit 2
Unit 3
Unit 4
Unit 5
abadaa
badcbc
aadceb
addeab
baddcd

Figure 4: Detailed Representation of the proposed approach with green block representing the offline stage, blue block showing the online stage and red block showing real time stage. The Output predictions of the proposed methodology are shown in the prediction decision block (red).

In the offline stage, responses of the measuring units deployed in the system are collected for trip injection attacks with varying attack locations, operating conditions and different system topologies. A discretization technique as shown in the yellow block of Fig.4 is used on database collected to convert the continuous responses converted into discretized symbols. The variables used in the discretization process are obtained through a cross validation study. Attack Severity and Criticality assessment study is performed for all the attacks considered to rank the attacks based on their severity and identify post attack critical elements in the system that are to be protected. This ranking is to provide situational awareness for known attacks and also to protect critical elements following an initiating attack to prevent the attack in being a cascading failure. The offline training databases are also used in the online stage to build data mining models that are capable of identifying disturbances in the system.

Online stage involves clustering of measuring units in the system using online divisive agglomerative clustering. This clustering is to monitor only few units in the system, referred as cluster representatives units. The cluster representatives selected, change continuously with system operating conditions based on the real time data arrived at each instant. The offline databases are also updated in this online stage with the real time responses seen by all the units and the predictions made in real time by the proposed algorithm for those responses.

The cluster representatives selected in the online stage, the model built and real time data are used in the real time stage to detect if there is a disturbance in the system at the current instant of time. If a disturbance is predicted in the system, its probable location is identified using a variance index generated and also a third prediction stating whether the disturbance is from a genuine event or from a remote trip injection attack is made in this stage. For the third prediction, a classification model based on the database ($\bar{C}_{\mu 1}^{N}$) is used to make the predictions.

The complete methodology of the proposed as shown in Fig.1 with the prediction output decisions shown in the red blocks (In the Red Prediction Decisions block on the right). Finally a malicious data detection algorithm is implemented to discriminate physical attacks from malicious or duplicate data representing an attack in the system. More detailed description of all processes involved in each stage is given in the upcoming sections. At first the formulation of attack severity and attack criticality are explained, later the attack detection methodology is presented.

3.2 Attack Severity and Criticality Assessment

This research primarily concentrates on remote tripping command injection attacks on various elements in the system. Tripping command is sent to relay or circuit breaker by compromising respective physical security, communication channels and protection devices.

L target elements are selected in this study where an element can be a load, generator, bus node and a transmission line. Scenarios are generated such that attacks occur randomly on the selected elements along with the natural events. Attack al on an element l can be an initiating attack or al may follow after N-k contingencies in the existing system or multiple attacks on different elements can occur one after other. For example, following a 3 phase short circuit fault on a bus, multiple tripping attacks can be initiated by the attacker on other elements in the system or following an attack on critical infrastructure, several N-k natural contingencies can occur. Hence on randomly choosing the attack location, attack order and the number of attacks occurring, several scenarios are obtained with different system topologies and operating conditions. Due to the large number of obtained scenarios, the computation performed in this study is extremely tiresome, but it is implemented to answer the following questions.

For an attack al on an element l,

a) How severe is this attack on the system?
b) What are the post attack critical elements that are to be closely monitored or protected to prevent the attack in being a failure?

This analysis is to protect the system by closely monitoring the subset of post attack/event vulnerable elements in the system to prevent the occurred attack/event in being a failure. It also gives the operator the needed discretion and awareness to act following an attack. In order to meet all the goals mentioned above, the proposed research assumes that the knowledge about system topology after N-k events or attacks, complete description of the attack/events occurred and post attack/event system operating conditions are known. Severity and criticality identification procedures formulated in this paper are as follows.

3.2.1 Attack Severity

Severity of an attack is dependent on post attack elements isolated from the system, percentage loss of load and generation, change in the system variables, closeness of the variables to their respective limits and the post attack stability of the system. For the attack a_i on element i, severity can be given by (1)

$$Severity_i^k = \frac{S_{i-1}^{k-1}}{S_i^k}\left\{a_1 * \sum_{g=1}^{G} \Delta GLC_g + a_2 * \left(Change_i + Reach_i\right) + a_3 * \left(1 - TS_i\right)\right\} \qquad (1)$$

where, attack a_i occurred after k events and k={1,2,3..} in *N-k*. If k=1, attack L_i is the initiating event in the system and S_{i-1}^{k-1}, $S_i^k$are the number of pre and post attack elements connected to the system. $a_{1,2,3}$

represents the weights assigned to each factor. TS_i is the transient stability of the system following the attack L$_i$ and is

$$\mathrm{TS_i} = \begin{cases} 0, \text{transiently unstable system} \\ 1, \text{transiently stable system} \end{cases} \tag{2}$$

ΔGLC_g is the percentage loss of cumulative generation and load in the system with respect to their pre-attack values at all the generation and load buses g ϵ {1..,.G}.

$Change_i = \sum_{j=1}^{S} Change_i^j$ is the summation of post attack change in system parameters (voltage and injections) of all the elements j, where j∈ S, and is given in (3)

$$Change_i^j = \left(\left|\frac{B_i^j - B_{i-1}^j}{B_{i-1}^j}\right|\right) \tag{3}$$

$Reach_i = \sum_{j=1}^{S} Reach_i^j$ is the summation of the closeness of the system variables to their limit values for all the elements j, where j∈ S, and is given in equation 3.

$$Reach_i^j = \left(1 - \left|\frac{B_{Lim}^{j*} - B_i^j}{B_{Lim}^{*}}\right|\right) \tag{4}$$

B_i^j, B_{i-1}^j are the post and pre attack magnitudes or voltage and injections of element j. B_{Lim}^{j*} is the maximum or minimum value of the limit imposed on an element *j* and is selected as

$$B_{Lim}^{j*} = \begin{cases} B_{Lim}^{j,\max} \;\; when \;\; B_i^j \geq \frac{B_{Lim}^{j,max} + B_{Lim}^{j,min}}{2} \\ B_{Lim}^{j,min} \;\; when \;\; B_i^j < \frac{B_{Lim}^{j,max} + B_{Lim}^{j,min}}{2} \end{cases} \tag{5}$$

(4) gives the idea of how close the system variables are to their limits, higher the value of (4), closer are the system variables to their limits. (1) gives us the impact of the attack on the entire system. Higher the magnitude of (1), more severe is the attack occurred. This severity index is computed for all the attack scenarios generated and are ranked accordingly based on (1) and the order k in which they occur. Now that the severity of the attack occurred is known, information about the critical elements in the system following a severe attack will give the power system operator necessary knowledge of subset *s* of the elements in the current system that are to be properly monitored and protected for preventing the attack in being a large failure.

3.2.2 Attack Criticality

The proposed critical element identification framework uses a probabilistic relation to identify the criticality of an element j in the existing system following the attack a_i. Criticality of an element depends upon the variation **C** observed in the system parameters following the attack and the frequency of occurrence of variation **C** in element j for different attacks in the system. The intuition here is not just to identify elements which have large change in their operational parameters but to identify elements in which these large changes occur frequently. Theoretically criticality is equivalent to Risk index [65]-[66] used in risk analysis study. Risk index of an event is equal to the product of probability of occurrence of an event (Outage, fault, failure etc) and the impact it creates on the system.

The proposed approach uses Symbolic Aggregate Approximation (SAX) which is explained in the next section to reduce the continuous values of variation in system parameters of element j for different attacks in the system except on element j. Criticality of each element j in the existing system after an attack a_i is given by

$$Criticality_i^j = \rho_{1_k}^j * Change_i^j + \rho_{2_k}^j * Reach_i^j + \rho_{3_k}^j * Structural\,Criticality_i^j \qquad (6)$$

$\rho_{1_k}^j, \rho_{2_k}^j, \rho_{3_k}^j$ are the probabilities of occurrence of the discretized ranges of these factors for all the events. For different attacks, these variables (Change, Reach and Structural Criticality) will have different set of values and repetition of exact same value for two different scenarios will be rare, hence the continues values of the variables (Change, Reach and Structural Criticality) for attacks at different locations are discretized into ranges using SAX and the probabilities tells us how frequent is a particular range. The significance of these probabilities can be explained with the following example, let us say 4 tripping attacks occurred in the system shown in Fig.5. Line 1-2 (line 1-2is j, the element under consideration and it is not tripped or attacked in the four cases) shown is reaching its limit for only one attack on an element p and for the other three attacks on elements other than p, it is reaching to ¼ of its limit. Now we can say that even though the line reaches its limit for attack on p, which makes it critical in that particular case. However Line 1-2 it is not a frequent critical line in the system which needs proper monitoring right after every attack. Now let us consider the same line is reaching its limit in three cases out of four considered, now the line is a critical element after an attack as it has higher probability of reaching limit and hence it needs proper monitoring and control actions in order to prevent the tripping of line 1-2 due to the initiating attack.

$Structural\ Criticality_i^j$ of an element j in the existing system after an attack a_i is the total number of elements that will be isolated from the system with tripping or removal of j. Fig.5 shows the structural vulnerability of element (Bus) 1.

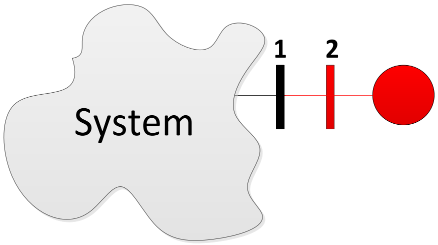


Figure 5: Model System. The Structural Criticality of Bus 1 is 2

$Structural\ Criticality_i^1$ of bus 1 following attack a_i is 2 as with the isolation of bus 1, line 1-2 and bus 2 are also isolated from the system. (6) gives the aggregate of the variations in the parameters of the element j because of attack a_i and the total number of elements lost from the system with the removal of j if the attack propagates. Hence after every attack a_i, criticality ranking is computed for all the elements still connected to the system. The elements with highest ranking should be properly monitored and protected. Similar to the severity index, the criticality of the existing elements in the system is computed for all the attack scenarios considered and are stored. The integrative flow diagram of the criticality and severity assessment scheme performed in the offline stage is provided in Fig.6. The algorithm stops analysis if and only if significant numbers of elements (Q) are lost from the system for an attack in the system. The value Q selected in this work is shown in the Chapter V.

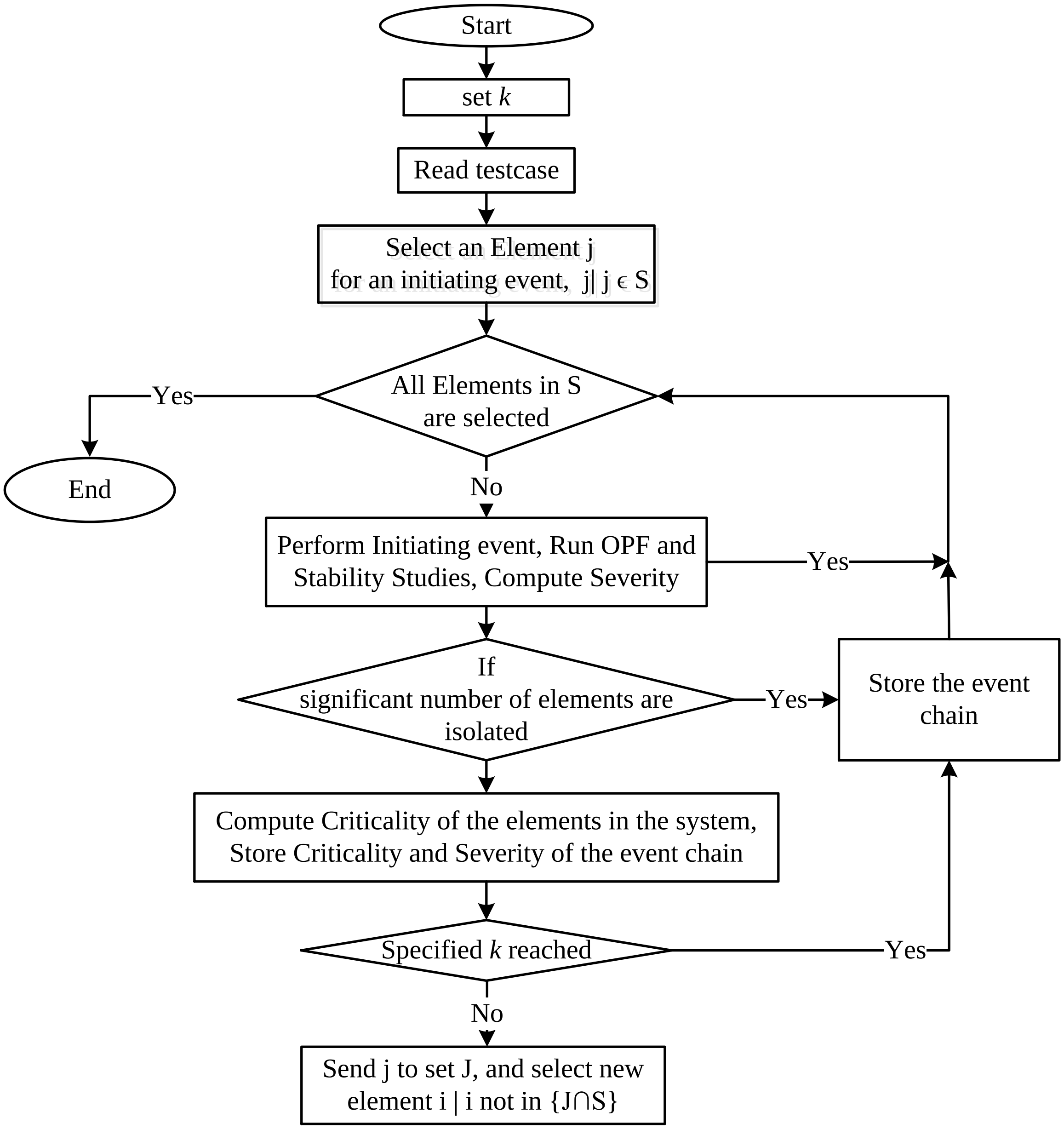

Start
set *k*
Read testcase
Select an Element j for an initiating event, j| j ϵ S
All Elements in S are selected
Yes
End
No
Perform Initiating event, Run OPF and Stability Studies, Compute Severity
Yes
If significant number of elements are isolated
Yes
Store the event chain
Compute Criticality of the elements in the system, Store Criticality and Severity of the event chain
Specified *k* reached
Yes
No
Send j to set J, and select new element i | i not in {J∩S}

Figure 6: Algorithm Employed for Severity and Criticality Assessment

3.3 Offline Analysis and Database Construction Stage

3.3.1 Symbolic Aggregate Approximation

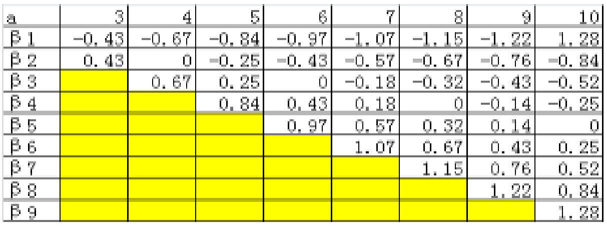
SAX is a process of symbolic representation of time series with an approximate distance function with the lower bound of the Euclidean distance [56]. SAX was proposed by Eamonn Keogh and Jessica Lin in the year 2002. It uses Piece Aggregate Approximation (PAA) in which each sequence of time-series data is divided into *s* segments of equal length (Sliding Window segments) and the average value of each segment is used as a coordinate of an s-dimensional feature vector. Actual Time series T of length n is represented in s-dimensional space as a vector shown in equation (1) [67].

$$T = T_1, T_2 \ldots\ldots T_n \Leftrightarrow \overline{C} = \overline{C}_1, \overline{C}_2 \ldots\ldots \overline{C}_S \qquad (7)$$

The i^{th} element of $\overline{C}$ is calculated by the equation

$$\bar{C}_i = \frac{s}{n} * \sum_{k}^{\frac{n}{s}i} T_k \qquad (8)$$

Where, k= $\frac{n}{s}$(i-1) + 1. From the above expressions, a time series of length n is divided into s equal sized frames. The mean values of data falling into particular frames are calculated and a vector of these values becomes the reduced representation of data. Hence, the original time series can be obtained by the linear combination of this vector. These time series inputs are initially normalized and then converted into lower dimension vector using PAA and hence, comparison of two different time series on a single frame is acceptable. After transforming time series using PAA, discretization step is performed to produce symbols of equal probability. The division point on the numeric axis is decided to equalize the cumulative probability in the division section according to the number of digitization (break point size β). Table 1 shows ten possible break points of equal probability on Gaussian distribution curve.

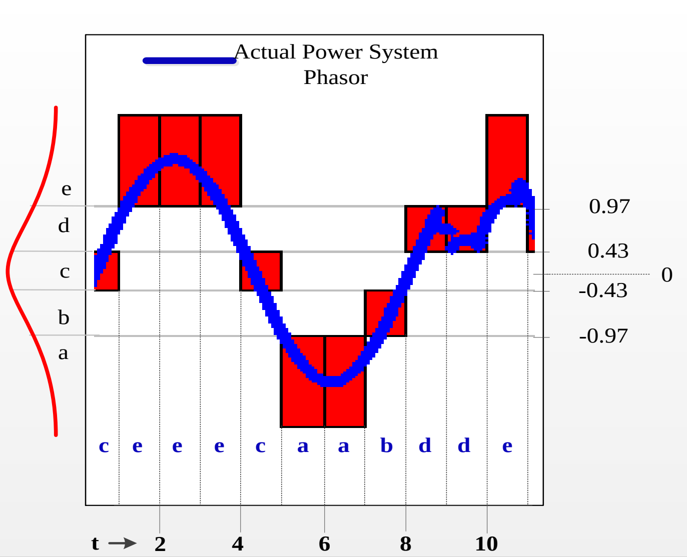


a	3	4	5	6	7	8	9	10
β 1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	1.28
β 2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β 3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β 4			0.84	0.43	0.18	0	-0.14	-0.25
β 5				0.97	0.57	0.32	0.14	0
β 6					1.07	0.67	0.43	0.25
β 7						1.15	0.76	0.52
β 8							1.22	0.84
β 9								1.28

Table 1: Different Break Points Obtained from Gaussian Distribution Curve

Once the breakpoint β size and the sliding window length (number of equal width points in which the series is divided ‘s’) are specified, the normalized curve is now discretized into symbols as shown in Fig.7.

Optimal number of breakpoints and sliding window length should be specified for proper representation of any particular dataset using SAX and these parameters may be obtained by a trial and error process. Once the time series is converted into a string of alphabets, these alphabets are then written in N-grams forms. The optimal order of the N-gram can also be obtained by trial and error process. Right side of the Fig.7 shows how a time series is converted into its N-gram representation for N=4 & 5.



N-gram Representation	
Actual discretized responses (c e e e c a a b d d e)^T	
N=5 5-grams	N=4 4-grams
c e e e c	c e e e
e e e c a	e e e c
e e c a a	e e c a
e c a a b	e c a a
c a a b d	c a a b
a a b d d	a a b d
a b d d e	a b d d
	b d d e

Figure 7: SAX based discretization and n-gram representation of a response

3.3.2 Offline Training Databases Generation

The phasor responses r^k of measuring unit k for all kϵNe, where Ne is the total number of units in the system are collected for all the considered system operating conditions and attacks. Phasor response of each unit is given by $r_{o1}^{k} = [p^{k}(t_{o1\_1}), p^{k}(t_{o1\_2}), p^{k}(t_{o1\_3}), p^{k}(t_{o1\_4}), p^{k}(t_{o1\_n})]^{T}$, where o1 is the system operating condition which persisted for a duration $t_{o1} =\{t_{o1\_1},,,t_{o1\_n}\}$ and p is the power system parameter (phase voltage) being considered. Similarly for all the operating conditions (OC’s), fault or other disturbance conditions **D** and attacks **i**={i1, i2,…}, the responses of all the units is represented as (9),

$$\boldsymbol{R}_{\mu} = \{r_c^{\ k} \mid k \in Ne \wedge c \in \mu, where\ \mu = \{OC's, D, i\}\} \qquad (9)$$

For Ne measuring units, the number of responses in $\boldsymbol{R}_{\mu}$ are <3 x Ne> if all the three phase responses of parameter p are considered and it is Ne if just single phase responses are considered. Two databases are created in this research; the first database is used in the disturbance identification process and the second database is used in attack detection process. First database uses $\boldsymbol{R}_{\mu}$ and a dependent variable or class variable (Y1). The dependent variable Y1={0,1} constitutes the information of the disturbance in the system. For r_{o1}^{k} shown above, $Y1_{o1}$ will also be of same length and is $\{0\}_{to1,1}$ if there is no disturbance in the system for the period to1 and $Y1_{o1}$ is $\{1\}_{to1,1}$ if there is a disturbance event or attack in the system for the duration to1. A classifier model is built on this database $\boldsymbol{R}_{\mu}$ in the online stage and it is used in the real time stage to predict $Y1_t$, the information about disturbance in the system at the current instant t.

The second database constitutes only the during and post attack /disturbance phasor responses of all the units for attacks **i** and faults or disturbance conditions **D** at different locations in the system. This database can be expressed as (10),

$$\boldsymbol{R}_{\mu 1} = \{r_{c1}^{\ k} \mid k \in Ne \wedge c1 \in \mu 1, \mu 1 = \{\mu \cap \{OC's\}\}\} \qquad (10)$$

The column length of $\boldsymbol{R}_{\mu 1}$ is similar to that of $\boldsymbol{R}_{\mu}$. SAX is used here and each phasor response r_{in}^{k}, $i_n \in \mathbf{i}$ is converted into symbolic format and can be expressed as $\overline{C_{i_n}^{k}}$ where $\overline{C_{i_n}^{k}}. = [\beta_{in}^{\ k}(s_1), \beta_{in}^{\ k}(s_2),,,,\beta_{in}^{\ k}(s_n)]^T$. The conversion of a phasor into symbolic string can be seen in the left side of Fig 7. where a continuous voltage phasor is represented as a string of discretized symbols. With the usage of SAX, the database $\boldsymbol{R}_{\mu 1}$ is converted into $\bar{C}_{\mu 1}$, this database is then represented in its N-gram form as $\bar{C}_{\mu 1}^{N}$ and the value of N is also a user defined parameter similar to s and β. $\overline{C_{i_n}^{k}}$ can be represented in its N-gram form, as $\overline{C_{i_n}^{k}}^{,N} = [\beta_{in}^{\ k}(s_1),\beta_{in}^{\ k}(s_2),,,\beta_{in}^{\ k}(s_N)\ ;\ \beta_{in}^{\ k}(s_2),,\beta_{in}^{\ k}(s_{\ (N+1)});\ldots.;..;\ \beta_{in}^{\ k}(s_{\ (n-N)}),,\beta_{i2}^{\ k}(s_n)\ ]$. N-gram representation of the symbolic string can be seen in the right side of Fig 7. Each row in $\overline{C_{i_n}^{k}}^{\ N}$ represents the behavioral pattern of the unit k for an attack i_n and can be referred as ‘instance’. The dependent variable or class variable for this database is constructed such that Y2j constitutes the information **μ1** and is a column vector of row length equal to the row length of $\bar{C}_{\mu 1}^{N}$. This database is further processed by removing the exact duplicate instances and in the real time; the responses from the measuring units are searched for these patterns or instances in this database $\bar{C}_{\mu 1}^{N}$ to detect the attacks in the system. From now, this database $\{\bar{C}_{\mu 1}^{N}$, Y2} is referred as attack pattern database throughout this thesis.

3.4 Online Model Construction, Verification and Validation

Clustering is employed in the proposed work to monitor only limited number of measuring units out of all the units distributed across the system to detect the occurrence of disturbances or attacks. Clustering is not implemented on the offline database; it is only implemented on the data accumulated at the units in the real time.

Clustering involves searching for objects with similar features and the objects here are the voltage responses of high sampling rates from all the PMU's in the system. These responses change with changing system μ and with that, the existing clusters and PMU units in each cluster should also change simultaneously. To deal with this issue effectively unlike [6] where clusters are unaltered in offline and in real time, this research employs an Online Divisive Agglomerative Clustering (ODAC) proposed in [68], where the clustering system continuously monitors the existing clusters and incrementally updates them. With the change in responses of the measuring units for different μ in real time (current instant), the clustering system employs a top-down strategy and the existing clusters are incrementally updated using a cluster split and aggregate criterion. The Cluster Aggregate and Split criterions depend upon the diameters of the existing clusters and a cluster diameter is the maximum distance (dissimilarity) between the responses of the measuring units in that particular cluster. Let a, b be the voltage (phase A) phasor responses or voltage data streams of unit k1 and k2 ϵ Ne. a(tmin)= [v1(1), v1(2), v1(3)….v1(tmin)]^T and b(tmin)= [v2(1), v2(2)….v2(tmin)]^T be their measurements from the instant t=0 to t=tmin with tmin being the current instant. The dissimilarity between the data streams a and b is given by;

$$dissimilarity(a,b) = \sqrt{\frac{1 - Coerr(a,b)}{2}} \qquad (11)$$

$$Coerr(a,b) = \frac{P - \frac{AB}{n}}{\sqrt{A_2 - A_2^2 / n} * \sqrt{B_2 - B_2^2 / n}} \qquad (12)$$

Where, $A = \sum a$, $B = \sum b$, $A_2 = \sum a^2$, $B_2 = \sum b^2$ and n is the number of measurement samples in a and b for the time duration t=0 to t=t_{min}. Using (11), the clustering system continuously computes the dissimilarity measure between all the data streams. The total number of data streams to be monitored depends upon the power system parameter p being considered and the number of units deployed. As mentioned in the previous section, the total number of data streams similar to a, b are Ne, if only a single power system parameter (single phase voltage phasor) is considered for each unit. Initially at the beginning of clustering, all the data streams are in a single cluster. With real time responses being updated at each instant, the clustering algorithm uses the test and split criterions to divide the data streams into several clusters. Algorithm 1 in Fig.8 gives the overview of online clustering and exact cluster split and aggregate conditions are presented in [68]. The output of this online clustering stage is P clusters, with each cluster having varying number of data streams in it. From the output P clusters, representative streams rep:={ crep1 , crep2 ,..., crepg } are selected and each element in the rep is called a cluster representative stream and these representatives are chosen such that it represents the behavior of majority streams in that particular cluster to which it belongs. The representatives selected for each cluster are the cluster diameter streams and a stream which is very close to the centroid of the cluster. From all the units in the system, only the representative streams are monitored in the real time stage to detect if there is an attack in the system. With the new data accumulating at the streams at each instant, the clusters change and with the change in clusters, the cluster representatives also change.

Algorithm 1:

Input:

a) $\boldsymbol{R_t}$:< r_t^1, r_t^2, r_t^3, r_t^4,…. r_t^{Ne}> be the streams from all the Units Ne

b) **t** 🡐 current time instant **, $\mathbf{t_d}$**🡐 desired testing time

Output:

a) **HC <$\boldsymbol{c^1}$, $\boldsymbol{c^2}$,..$\boldsymbol{c^g}$>,** for each $\boldsymbol{c^k}$ **|** $\boldsymbol{c^k}$ **:=**<r_t^a…r_t^b>, ∀ k ϵ (1,..g), (a,..b) ϵ Ne and (r_t^a,.. r_t^b) ϵ $\boldsymbol{R_t}$,be the cluster division of streams in $\boldsymbol{R_t}$

b) **crep: <$\mathbf{rep^1}$, $\mathbf{rep^2}$, $\mathbf{rep^3}$,…$\mathbf{rep^p}$>**, $\mathbf{rep^i}$| $\mathrm{rep}^{\mathbf{i}}$ ϵ $\boldsymbol{c^k}$ be the cluster representative unit of cluster $\boldsymbol{c^k}$

```
if t < 2 cycles
    HC = < $\boldsymbol{c^1}$ > = ( $r_t^1, r_t^2, r_t^3, r_t^4, \ldots. r_t^{Ne}$); all units are in one cluster for t < 2.
    $rep^1$ and $rep^2$ are chosen as representative streams of $\boldsymbol{c^1}$ | $rep^1 = r_t^a$, $rep^2 = r_t^b$ $\forall\, r_t^a, r_t^b \in \boldsymbol{R_t}$ & dissimilarity( $r_t^a, r_t^b$ ) is maximum of dissimilarity measure between any other streams in the cluster .
    $rep^3$ is chosen as representative stream for $\boldsymbol{c^1}$ | $rep^3 = r_t^c$ $\forall\, r_t^c \in \boldsymbol{R_t}$ & ($r_t^c$ – centroid ($\boldsymbol{R_t}$)) is minimum for any other stream in the cluster.
    representative streams of $\boldsymbol{c^1}$ = { $\boldsymbol{rep^1}$, $\boldsymbol{rep^2}$, $\boldsymbol{rep^3}$ }.
else
    for every 2 cycles of t until the desired $t_d$
        Read existing HC
        $\forall\, \boldsymbol{c^i} \in$ HC, test Cluster Split condition
        if Cluster Split Holds
            split $\boldsymbol{c^i}$ | $\boldsymbol{c^i} = < \boldsymbol{c^{i1}}, \boldsymbol{c^{i2}} >$ using the Cluster Split Criterion
            Update Current HC
        end if
        $\forall\, \boldsymbol{c^i}, \boldsymbol{c^j} \in$ HC, test Cluster Aggregate condition
        if Cluster Aggregate Holds
            aggregate $\boldsymbol{c^i}, \boldsymbol{c^j}$ | $\boldsymbol{c^{i\,new}} = < \boldsymbol{c^i}, \boldsymbol{c^j} >$
            Update Current HC
        end if
        $\forall\, \boldsymbol{c^i} \in$ HC find cluster representatives $rep^a$, $rep^b$ | $rep^a, rep^b \in \boldsymbol{c^i}$ & dissimilarity ( $r_t^a, r_t^b$ ) is maximum of dissimilarity measure between any other streams in the cluster.
```

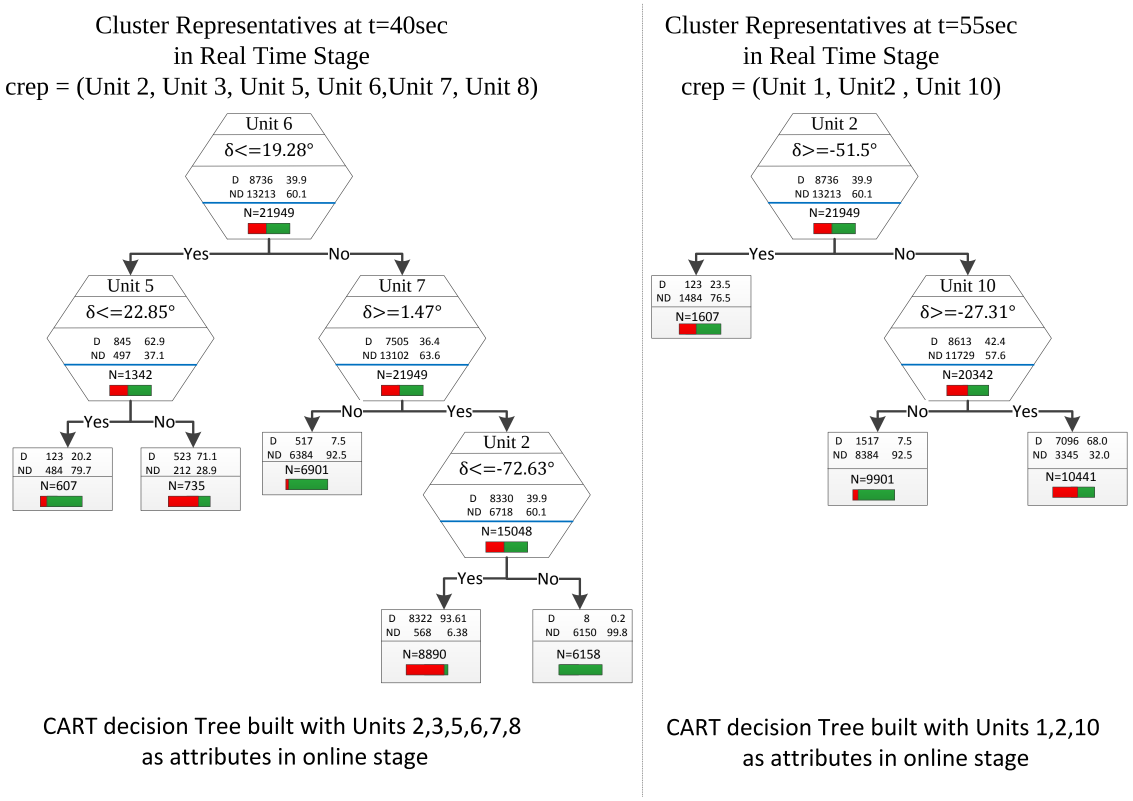
```
$\forall c^i \in$ HC find cluster representative $rep^c \mid rep^c = r_t^c \; \forall r_t^c \in \boldsymbol{R_t}$ & ($r_t^c -$ centroid ($\boldsymbol{R_t}$)) is minimum for any other stream in the cluster.
Output HC and current time instant crep
end for
```

Figure 8: Algorithm I, employed for online clustering

3.4.1 Model Construction for Identification of a Disturbance

Training Database { Rμ, Y1} constructed earlier is used to build a classifier model that is used to detect the occurrence of a disturbance (any D or i) in the system at the current instant based on the responses of cluster representatives. As mentioned in the previous section, the cluster representatives change with new data incoming, hence the classification model built on one set of representatives cannot hold for the next instant if the representatives which are the attributes (variables on which the model is built) for the model change continuously. Hence the decision tree techniques used in mentioned in literature review section will not be used here because of this issue of changing attributes. Fig.9 explains this, where a decision tree built using CART classification algorithm [32] at t = 40 seconds cannot be used for predictions at t = 60 seconds.

To deal with this situation of changing attributes, a Random Forest classifier [69] is implemented in this thesis. Using this Random Forest algorithm, decision trees are built considering random number of attributes (each response or each column in Rμ). Brief description about random forest classifier is presented next



Cluster Representatives at t=40sec in Real Time Stage crep = (Unit 2, Unit 3, Unit 5, Unit 6,Unit 7, Unit 8)

CART decision Tree built with Units 2,3,5,6,7,8 as attributes in online stage

Cluster Representatives at t=55sec in Real Time Stage crep = (Unit 1, Unit2 , Unit 10)

CART decision Tree built with Units 1,2,10 as attributes in online stage

Figure 9: Left and Right figure shows decision tree built at t=40 sec and t=55sec using their respective cluster representative units as attributes

3.4.1.1 Random Forest Classifier

Traditional tree learners like CART and C4.5 cannot scale changing database problems since they assumes data is loaded into main memory and executed within one thread**.** Random Forest Classifier is an assemble learning methodology where a multitude of decision trees are built during training time with varying number of attributes. In the prediction stage, the class variable of a training sample is the mode of the class variables from all the trees in the forest. The method combines Breiman's "bagging" idea [69] and the random selection of features in [70]-[71] to construct a collection of decision trees with controlled variance. The input to the Random Forest Classifier during training period is a training database Rμ, m the number of attributes (or responses of the measuring units) on which the tree and the post pruning variables. Post pruning is implemented because the trees that are grown very deep tend to learn highly irregular patterns and they may even over fit their training sets, because they have low bias, but very high variance in order to prevent this, post pruning is performed to remove the nodes is the trees that are grown due to over fitting.

3.4.2 Model Verification, Validation and Database Update:

The entire algorithm is simulated such that in real time new data will be incoming and the algorithms built is used to detect if there is an attack in the system. The databases { $\boldsymbol{R_{\mu}}$, Y1} and {$\bar{C}_{\mu 1}^{N}$,Y2} are constantly updated with the real time responses. The classifier model built in the previous section is also updated with the new database { $\boldsymbol{R_{\mu}}$, Y1} only when the prediction accuracy of classifier falls low. The attack patterns database {$\bar{C}_{\mu 1}^{N}$, Y2} is also carefully updated after each attack prediction. If this database is not properly updated, the accuracy of the attack predictions changes drastically. For this all the instances that are predicted as attack responses are updated in the attack pattern database.

3.5 Real time Prediction Stage

As mentioned before, the input of this stage is the real time data from all the measuring units in the system. The outputs of this stage are for the current instant,

a) Prediction stating whether there is a disturbance in the system?

If the prediction comes out be a disturbance in the system at the current instant, then

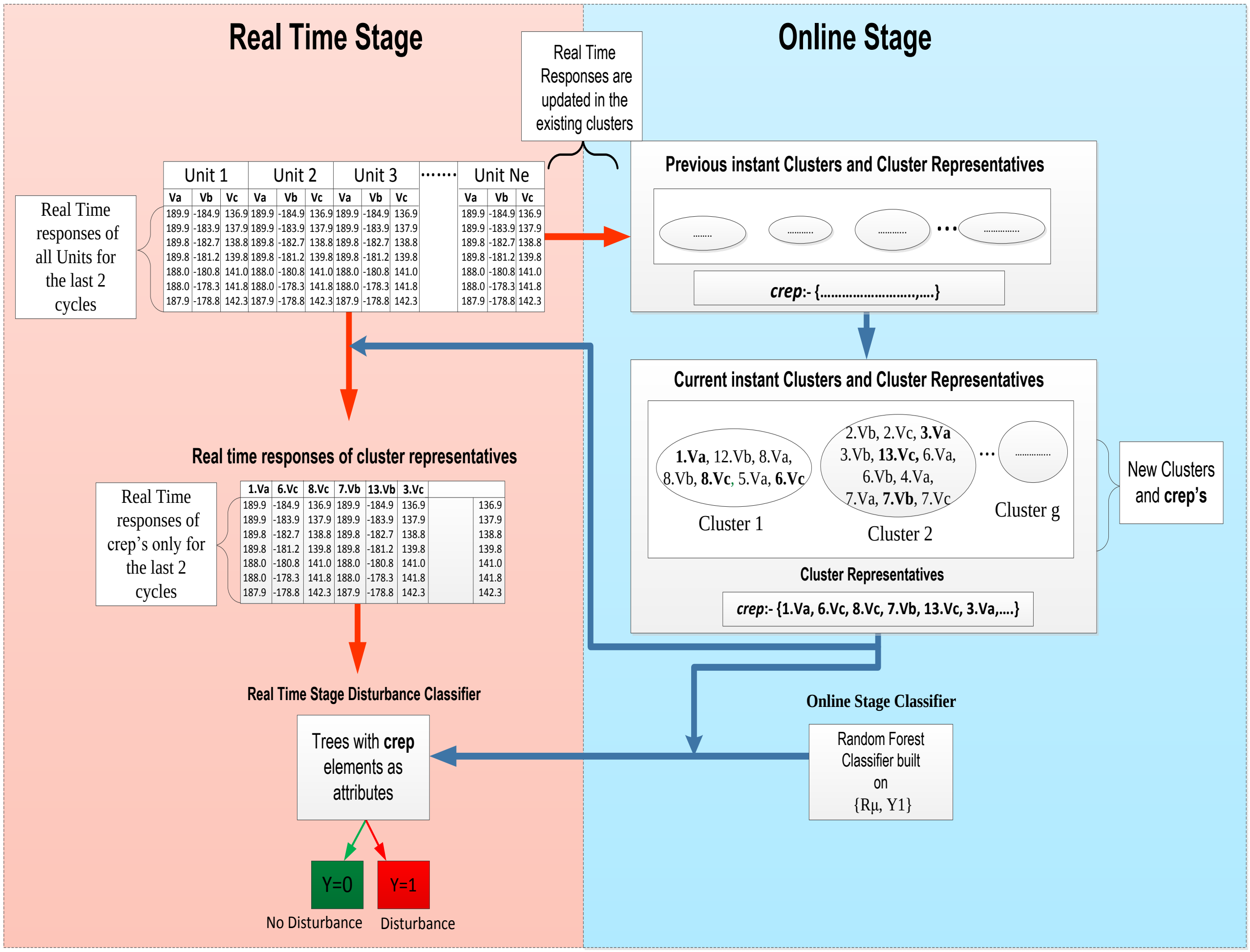
b) Prediction of the location where the disturbance initiated.

c) Prediction stating if the disturbance is originated from an attack in the system?

As shown in Algorithm 1, the clusters are updated after every n_{min} cycles of new data arrival. The real time stage is simulated such that voltage data samples will be arriving after every 2 cycles to the system. These 2 cycle data samples arrived and the cluster representatives for the current instant are sent to the disturbance prediction model explained in the online model construction stage of the last section to identify if there is a disturbance in the system.

3.5.1 Disturbance Prediction Stage

Random forest classifier built on the database { R_{μ}, Y1} and the responses of *rep* for the 2 cycles of the new data arrived is used to predict if there is a disturbance in the system for the past two cycles. This disturbance detection is explained using Fig.10.


Real Time Stage
Online Stage
Real Time Responses are updated in the existing clusters
Real Time responses of all Units for the last 2 cycles
Previous instant Clusters and Cluster Representatives
crep:- {……………………..,….}
Current instant Clusters and Cluster Representatives
1.Va, 12.Vb, 8.Va, 8.Vb, 8.Vc, 5.Va, 6.Vc
Cluster 1
2.Vb, 2.Vc, 3.Va 3.Vb, 13.Vc, 6.Va, 6.Vb, 4.Va, 7.Va, 7.Vb, 7.Vc
Cluster 2
Cluster g
New Clusters and crep's
Cluster Representatives
crep:- {1.Va, 6.Vc, 8.Vc, 7.Vb, 13.Vc, 3.Va,….}
Real time responses of cluster representatives
Real Time responses of crep's only for the last 2 cycles
Real Time Stage Disturbance Classifier
Trees with crep elements as attributes
Y=0
Y=1
No Disturbance
Disturbance
Online Stage Classifier
Random Forest Classifier built on {Rμ, Y1}

Unit 1			Unit 2			Unit 3			…….	Unit Ne		
Va	Vb	Vc	Va	Vb	Vc	Va	Vb	Vc		Va	Vb	Vc
189.9	-184.9	136.9	189.9	-184.9	136.9	189.9	-184.9	136.9		189.9	-184.9	136.9
189.9	-183.9	137.9	189.9	-183.9	137.9	189.9	-183.9	137.9		189.9	-183.9	137.9
189.8	-182.7	138.8	189.8	-182.7	138.8	189.8	-182.7	138.8		189.8	-182.7	138.8
189.8	-181.2	139.8	189.8	-181.2	139.8	189.8	-181.2	139.8		189.8	-181.2	139.8
188.0	-180.8	141.0	188.0	-180.8	141.0	188.0	-180.8	141.0		188.0	-180.8	141.0
188.0	-178.3	141.8	188.0	-178.3	141.8	188.0	-178.3	141.8		188.0	-178.3	141.8
187.9	-178.8	142.3	187.9	-178.8	142.3	187.9	-178.8	142.3		187.9	-178.8	142.3

1.Va	6.Vc	8.Vc	7.Vb	13.Vb	3.Vc		
189.9	-184.9	136.9	189.9	-184.9	136.9		136.9
189.9	-183.9	137.9	189.9	-183.9	137.9		137.9
189.8	-182.7	138.8	189.8	-182.7	138.8		138.8
189.8	-181.2	139.8	189.8	-181.2	139.8		139.8
188.0	-180.8	141.0	188.0	-180.8	141.0		141.0
188.0	-178.3	141.8	188.0	-178.3	141.8		141.8
187.9	-178.8	142.3	187.9	-178.8	142.3		142.3

Figure 10: Flow of control in online stage (blue) and real time prediction stage (light red)

As shown in Fig.9, trees containing streams in *rep* as attributes are only used to decide if there is a disturbance in the system. Online clustering can also be seen in Fig.10, where all the units are clustered into different groups and representatives are selected from each group. The new data from these representatives and trees build on these representative streams from the database { R_{μ}, Y1} are selected to make the prediction of disturbance in the system in the real time stage.

With significant changes in network topology, load levels and system variables, the data incoming in real time also change. Accuracy of prediction will be low if the tress generated on the old database is used to predict this new data. Hence the classifier needs to be repeatedly updated combining the existing database and the new data whenever the accuracy of prediction falls low.

3.5.2 Disturbance Cluster Location

This process of identifying the location of the disturbance cluster only starts when the prediction about the disturbance in the system comes out to be true (Y1 for instant is equal to 1). Variance in the measurements of each representative stream is used as an index to locate the origin of the disturbance occurred. We assume that there is only one disturbance in the system at a time instant and from the conclusions of [6]-[72], the changes in measurements of the monitoring units near by the disturbance origin are high when compared to units which are far away from the disturbance origin. Similar variance index is also formulated in [6] to identify the location of the generators where the transient event has occurred.

Variance in responses of each cluster representative unit is given by (13)

$$Var^{ck} = \frac{\sum_{t=t-}^{t} \left| crep^{ck}(t+1)^2 - crep^{ck}(t)^2 \right|}{T^{ck}} \tag{13}$$

Where $crep^{ck}$ is the cluster representative unit, $crep^{ck}$ ϵ rep, $crep^{ck}(t - to\ t)$ are the responses seen by the unit ck from ti ={t$^-$ to t} which is 2cycles and t being the current time and n is the total number observations (voltages) available for the duration ti. *Var*^{ck} is computed for all the cluster representative streams and representative with largest *Var*^{ck} is selected. The cluster to which the selected representative stream belongs to is considered as the disturbance cluster **DC**. It is inferred that the identified disturbance has originated in the electrical neighborhood of the measuring units in that cluster. The responses of all the measuring units in the disturbance cluster are further examined to determine whether the originated disturbance is from an attack in the system or a genuine event.

This method of identifying disturbance origin will be appropriate only when there is a sudden change in the responses of the localized group of measuring units near to the disturbance. The method would not be appropriate for the disturbances where the oscillations in system variables accumulate over a long period of time following the disturbance.

3.5.3 Attack Prediction

The disturbance cluster **DC** from disturbance location stage, cluster representatives rep and the attack pattern database $\bar{C}_{\mu 1}^{N}$from the offline stage are used to detect if the occurred disturbance is a known attack. k-Nearest Neighbor algorithm (kNN) [73] is implemented in this research for the attack detection process. kNN is an instance based non-parametric method used for classification and regression. The input to the k-NN classification algorithm are the k closest training examples in the feature space and its output is the most common class or dependent variable among its k nearest neighbors identified. Being the simplest of all machine learning algorithms, kNN uses a user defined similarity measure to identify its neighbors from the training database; hence the performance of the algorithm crucially depends upon the similarity measure chosen [73].

kNN classification, for a given a query string Q and a collection of strings S in the training database $\bar{C}_{\mu 1}^{N}$, returns a set of strings $\bar{S}$ such that $|\bar{S}|$=k, and $\forall\, \bar{s} \in \bar{S}$, $\forall$ s $\in$ S-$\bar{S}$, *dist*($\bar{s}$,Q) ≤ *dist*(s,Q), where k is a user specified constant and *dist* is the similarity measure used.

Three different similarity measures are tested individually here to identify the neighbor instances from the training database. This is to find a better distance measure that can efficiently and accurately identify neighbor samples from the training database.

Let **DC**=[e_1,e_2…e_{g1}] be the elements (measuring units) in cluster **DC** including its representatives. The real time phasor responses of these measuring units $e_i \in$ **DC** for the duration ti are converted into symbolic format and then represented as N-grams. The values for parameters s, β and N used here are equal to the ones that are used in the offline database construction stage. Let the converted real time responses is given by $\bar{C}^{N,D,ti}$. Neighbors (strings with similar symbols) are searched in the training database for the strings in $\bar{C}^{N,D,ti}$ using similarity or distance measure and the dependent variable Y2 <disturbance is due to an attack, disturbance is not due an attack> of the majority neighbors is the decision of attack in the system.

The similarity measures used in this paper are assessed based on their accuracy in identifying attacks in the test system. Brief description about the similarity measures uses is as follows; Let A and B are two sequences, similarity between them using different techniques is given by;

a) Euclidean Distance: Euclidean distance between two streams or sequences is the ordinary point to point distance between the samples in the streams.

Consider the sequences A={a1,a2,a3,a4,a5…at} and B={ b1,b2,b3,b4,b5…bt}. The Euclidean Distance between strings A and B is given by;

$$\text{Euclidean Distance (A, B) = } d(A,B)=\sqrt{\sum_{i=1}^{t}(a_i-b_i)^2} \qquad (14)$$

Note for Euclidean Distance measure, the strings should be of similar size and also if there are missing values in it, predefined values (commonly missing values are considered zero) are to be assigned to these missing values.

b) Dynamic Time Warping: [75] finds an optimal alignment between two given sequences in order to generate the most representative distance measure of their overall difference. The DTW algorithm uses dynamic programming technique to solve the problem of non-linear time series warping by intuitively stretching or shrinking the sequences along the time axis. From (14) we can observe that Euclidean distance measure uses an obvious way to compare the time series signals by adding the point to point differences along the signals. However, problem arises when there is any slight discrepancy in the alignment of the signals. DTW can automatically deal with these time deformations, discrepancies and different speeds associated with the fast streaming time-series data [75].

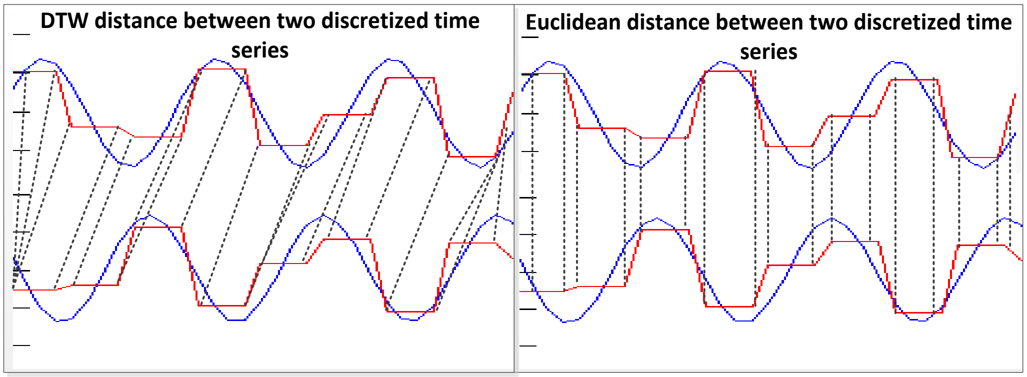


Figure 11: Time series distance alignment based on DTW and Euclidean distance.

The total cost $c_w(A,B)$ of warping path W between A and B with respect to local cost measure c is defined as:

$$c_w(A, B) := \sum_{l=1}^{l} c(a_{nl}, b_{ml}) \qquad (16)$$

The DTW distance between discretized streams A and B is then defined as the total cost of w^*:

$$\text{DTW}\ (A,B) := c_w^*\ (A,B) \qquad (17)$$

This distance measure is used as an objective function to find the similarity between the processed responses of the measuring units. Generally DTW computations takes time, still the accuracy of the output decisions and algorithm speeding techniques proposed in [76] made us choose this method.

c) Levenshtein distance or Edit Distance: [77] is a string metric for computing how similar or dissimilar two strings are by counting the number of point mutations taken for one string to turn into the other, where a point mutation is defined as a single character change. Each mutation in Edit distance can be either done by

1) Insertion (i),

2) Deletion (d),

3) Substitution (s)

Dynamic programming is used to compute these minimum number steps required and edit distance can be given as

$$\textbf{Edit distance (A, B)} = \sum \min(i, d, s)\ for\ converting\ A \rightarrow B \qquad (18)$$

The greater the edit distance is, the more different the strings are. Edit distance has several applications in text and document mining field and some of them are document retrieval, spell checking, speech recognition, DNA analysis, plagiarism detection

Euclidean distance is the simplest and more generalized form of distance measure whereas DTW is computationally tiresome. Euclidean distance is used here only to manifest the efficiency and accuracy achieved by using more relevant and computationally tiresome distance measures like DTW. This is to show that the algorithms proposed in [6] choose Euclidean distance measure randomly for clustering and this should not be the case as the behavior and performance of different databases on different similarity measures will be quite different. Thus several distance measures are to be tested and the ones that have decent performance outputs are to be selected. In this thesis, several distance measures other than those presented above are tested in a cross validation for this particular problem and the distance measures that performed very well are only chosen. DTW and EDIT are the algorithms which performed very well during testing and in real time implementation stage. The performance of these measures is presented in Chapter V of this thesis. The algorithm implemented for attack prediction stage is given below in Fig.12.

Algorithm 2:

```
Input: D, <$\bar{C}^{N,D}_{\mu 1}$, Y2>, $\bar{C}^{N,D,ti}$
Output: Y2 for $\bar{C}^{N,D,ti}$
Read $\bar{C}^{N,D,ti}$, $\bar{C}^{N,D}_{\mu 1}$
for each ei ϵ DC
    c ← column length of $\bar{C}^{N,ei,ti}$;
    for each $\bar{C}^{N,ei,ti}$(c,:)
        R1 = dist ($\bar{C}^{N,ei,ti}$(c,:) , $\bar{C}^{N,ei}_{\mu 1}$ ) ,
        R2(c, ei) = Y2 of symbolic string in $\bar{C}^{N,ei}_{\mu 1}$ with min(R1)
    end for
end for
Decision = find Y2 < attack, not an attack> from R2 which is the mode of the predictions.
```

Figure 12: Algorithm II, employed for real time attack detection

3.5.4 Malicious Data Attack Detection

The proposed framework in the previous sections will notify the operator if there is an attack in the system and announce its location, severity and criticality. Now with this information, the operator can maneuver protective actions to mitigate the attack or reduce its impact on the system. An important question instigates from the work proposed in [78] where malicious data pretending to be an attack in the system renders protective control actions that finally incur huge economic losses to system. The reasons for this huge loses are the control actions of load rescheduling and load shedding. To counter these malicious data attacks from the sensor units, this research proposes a prediction based malicious data detection strategy. Here the important assumption considered is that the attacker has weak malicious data attack regime. This means that the attacker can only compromise limited number of measuring units in the system.

So with this assumption described, the responses of the measuring units that are compromised can be predicted with the help of non-compromised units in the system. However this prediction can only be possible if there is a large behavioral database of all the measuring units in system with correlational information between a disturbance at one location in the system and consequent behavior of units at other locations of the system. To explain this with an example, let us consider a two bus system with buses A and B. For different disturbances (events and attacks) at bus A, the behavior of measuring units at bus B is known. Now for a known disturbance d_i at A, if the behavior of units at bus B is not according to the database collected, for this case we can conclude that either d_i is a new disturbance whose behavior is not in the database or it may be because of measuring units at A are compromised and malicious data representing an attack d_i is transmitted.

As shown in Algorithm 2, the prediction of attack in the system is based on the mode of class variable outputs of the Nearest Neighbors. So what if all the units are not predicting an attack in the system or vice-versa? This thesis introduces a new variable ξ, which presents the confidence of kNN attack prediction stage based on the number of units giving similar output. ξ presents the confidence of the kNN prediction by giving the ratio of number of units predicting a particular output and outside the disturbance cluster (considering few units in disturbance cluster are compromised) over the total number of units being considered for the prediction. So if ξ is somewhere less that a threshold considered, we infer that significant number of units being considered are predicting a different output for the current instant responses and hence further examination needs to be done on the current instant responses and the output prediction.

However, the scope of this thesis to initiate the proposed examination is limited only for the prediction "*Attack in the System"* from the kNN stage and with ξ < than a threshold considered. The term ξ can be expressed as

$$\xi = \frac{Number\ of\ Units\ outside\ the\ disturbance\ cluster\ predicting\ an\ Attack}{Number\ of\ Units\ Outside\ the\ disturbance\ cluster} \qquad (19)$$

Only when conditions a) and b) mentioned below comes true, the process of differentiating an attack with malicious data starts;

a) Output prediction (Y2) of kNN → " Attack in the System"

b) ξ < Threshold

To achieve this goals the proposed malicious data detection, a Naïve Bayes classifier is employed to predict the responses of units in the disturbance cluster **DC** by considering their output as dependent variables and the responses of the measuring units that predicted that there is no attack in the system as attributes. These predicted responses are compared with the actually seen responses to detect the possibility of a malicious data attack. The main reason for selecting Bayes classifier over other complex techniques is that it considers all attributes as independent. The second reason for choosing Naïve Bayes classifier is that it requires a database of instances all the time to make predictions. But these are already created in the training phase. The proposed detection system uses database $\bar{C}_{\mu 1}^{N}$ to identify possible malicious data. This is done as follows;

Let **DC** be the disturbance cluster and O := **crep** ∩ **DC**, be the cluster representative units of the other clusters. Now just units in O are considered as attributes and the responses of cluster representatives of units of **DC** is made one at a time. $crep_{DC}$ be the cluster representatives of disturbance cluster **DC**. Now real time discretized responses (symbols) of each unit in $crep_{DC}$ are predicted using responses of units in O. Let $f := \{o1, o2, ...\}$ be the responses of units in O. $P(fcrep_{DC} | fo1, fo2 ...)$ denote the probability of response $fcrep_{DC}$ at a measuring unit in $crep_{DC}$ for responses $\{fo1, fo2, fo3 ...\}$ at the other units belonging to O. Based on the independence property, we can write the expression for probability of a response $fcrep_{DC}$ with the responses of other attributes set as $\{fo1, fo2, fo3, ...\}$

$$P(Y = fcrepDC \ | fo1, fo2 \ldots) = \frac{P(Y=fcrepDC) \prod i \ P\ (foi|Y=fcrepDC)}{\sum_{oj} P(Y=foj) \prod i \ P\ (foi|Y=foj)} \qquad (20)$$

Where *foj* are the all non-duplicate responses of the element *oj* in *O*. The output of (20) will be zero if the response *fcrepDC* is not seen by the units in O till now. If the responses are not discretized, this computation cannot be possible as each measuring unit in the system can encounter infinite number of unique values. The prediction of output response Y for the responses set {*fcrepDC*} can be expressed as

$$Y \leftarrow \operatorname{arg\,max}_{y_k} \frac{P\left(Y = foi\right)\prod_{i} P(Xi \mid Y = foi)}{\sum_{j} P\left(Y = foj\right)\prod_{i} P\ (\, foi \mid Y = foj)} \tag{21}$$

This predicted output Y is computed individually for all the elements in $\mathrm{crep}_{\mathrm{DC}}$. These predicted responses are compared with the responses that are actually seen to detect a possible malicious data attack. This comparison is performed using DTW distance and an upper bound (λ) is calculated in the offline analysis such that if the DTW(Y, *fcrepDC*) falls above λ, this instance is classified as malicious data and stored in the database. If the difference falls below the bound, it is classified as 'unknown' and sent into offline stage for further analysis. The algorithm employed for this detection process is as follows;

Algorithm 3:

```
Input: DC, <$\bar{C}^{N,D}_{\mu 1}$, Y2>, $\bar{C}^{N,crepDC,ti}$, $\bar{C}^{N,O,ti}$

Output: Decision that the responses in $\bar{C}^{N,crepDC,ti}$ are due to malicious data attack or an unseen or unknown condition

Read $\bar{C}^{N,crepDC,ti}$, $\bar{C}^{N,O,ti}$, $\bar{C}^{N,D}_{\mu 1}$,
for each unit in $\mathrm{crep}_{\mathrm{DC}}$
    c ← column length of $\bar{C}^{N,crepDC,ti}$;
    for each $\bar{C}^{N,crepDC,ti}$ (c,:)
        Use Y, {fo1,fo2,fo3,…} responses of crep's in O to predict fcrepDC (c,:)
    end for
end for
Compute DTW ($fcrepDC$ , $\bar{C}^{N,crepDC,ti}$),
if DTW >threshold set for majority responses
    Malicious Data Attack,
else
    unseen or unknown,
end if
```

Figure 13: Algorithm III, employed for malicious data attack detection

Unknown or Unseen above means that the responses in $\bar{C}^{N,crepDC,ti}$ are newly seen by the system and are to be further analyzed or stored in the database. This approach is not implemented in real time mode in the methodology presented in Fig.4. It is only formulated in offline to see how it reacts to the cases where malicious data attacks representing actual attacks are sent through the system.

Chapter 4: SIMULATION TOOLS AND SOFTWARE

This chapter briefly describes various software packages used in this thesis for creating rigorous training databases, analyzing several data mining algorithms on these training databases generated and finally performing online simulations and real time predictions. The test power system considered in this thesis is modelled and simulated in MATLAB to create rigorous training databases. Data mining techniques are implemented through Weka and R statistical programming language tools. Brief description of MATLAB is presented first in this chapter. Comprehensive description about both Weka 3 and R follows next.

4.1 MATLAB

MATLAB®, Matrix Laboratory, is a high-level language and numerical computing environment for logic development, data visualization, data analysis and numeric computations. Some of the important features of MATLAB are [79].

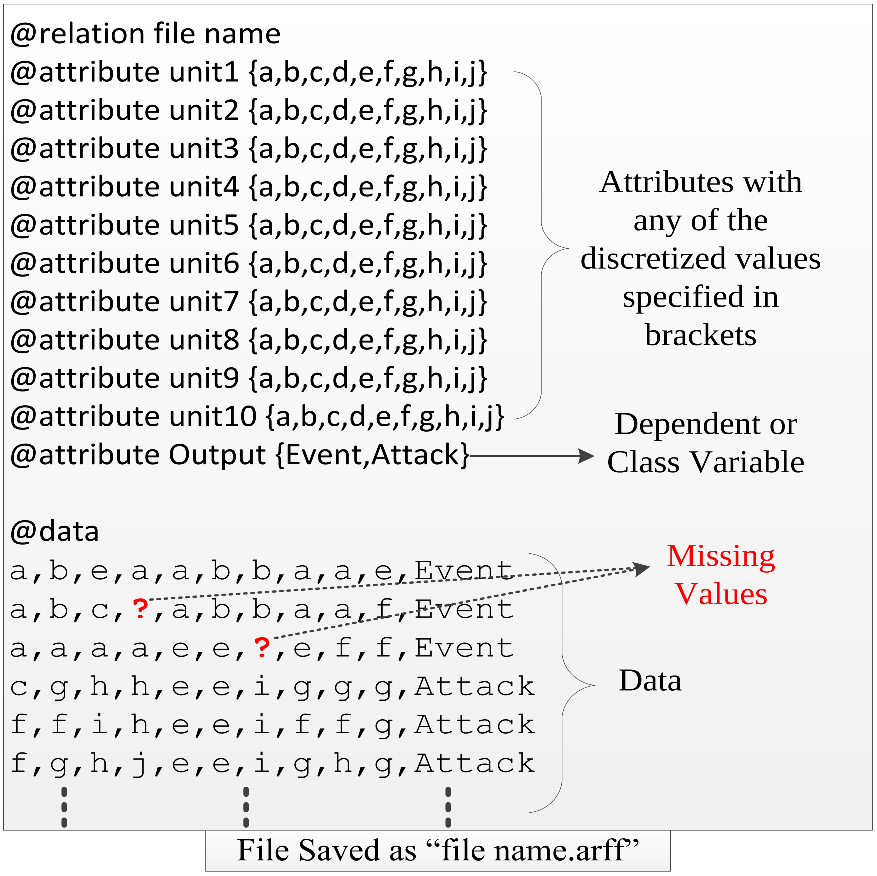
- Communicating environment for iterative programming, design and problem solving
- High level language for technical computing
- Integrated graphics for imagining data and tools for making custom plots
- Can be integrated with external applications such as C, C++, FORTRAN, Java, and Excel
- Tools for developing applications with custom graphical interfaces

In this thesis, numerous MATLAB functions were used to efficiently simulate the test power system considered for different system operating conditions, attacks and scenarios.

Along with the above mentioned cause, MATLAB is also used in this thesis to program all the data mining algorithms employed. Although these data mining algorithms are available in open source tools R and Weka 3, they are still programmed in MATLAB because of its friendly communicating environment and high speed computing. The high speed computing is predominant in the real time prediction stage as the predictions made are to be extremely fast matching the speeds of the industry equipment that are used for similar disturbance detection cause.

4.2 WEKA

Weka is a collection of state-of-the-art machine learning techniques, data mining algorithms and data preprocessing tools. It provides extensive support for the whole process of experimental data mining, including preparing the input data, evaluating learning schemes statistically, and visualizing the input data and also the results obtained from the learning process [80]. Along with the wide variety of extensive learning algorithms, it also includes a wide range of preprocessing tools. This diverse and comprehensive toolkit is accessed through a common interface so that the users can compare different methods and identify those that are most appropriate for the database at hand and the problem that is being dealt. Weka is actually a Java based program and issued under General Public License. However, it can be implemented in many platforms including Windows, Linux and Macintosh OS. Weka also contains built in high level algorithms like decision trees, clustering and regression techniques that can be implemented easily of different vast datasets. Firstly the databases are to be prepared in the form of ARFF tables before uploading them into Weka. The format of a database in the ARFF tables can be seen in Fig.14 with a sample training dataset.



```
@relation file name
@attribute unit1 {a,b,c,d,e,f,g,h,i,j}
@attribute unit2 {a,b,c,d,e,f,g,h,i,j}
@attribute unit3 {a,b,c,d,e,f,g,h,i,j}
@attribute unit4 {a,b,c,d,e,f,g,h,i,j}
@attribute unit5 {a,b,c,d,e,f,g,h,i,j}
@attribute unit6 {a,b,c,d,e,f,g,h,i,j}
@attribute unit7 {a,b,c,d,e,f,g,h,i,j}
@attribute unit8 {a,b,c,d,e,f,g,h,i,j}
@attribute unit9 {a,b,c,d,e,f,g,h,i,j}
@attribute unit10 {a,b,c,d,e,f,g,h,i,j}
@attribute Output {Event,Attack}

@data
a,b,e,a,a,b,b,a,a,e,Event
a,b,c,?,a,b,b,a,a,f,Event
a,a,a,a,e,e,?,e,f,f,Event
c,g,h,h,e,e,i,g,g,g,Attack
f,f,i,h,e,e,i,f,f,g,Attack
f,g,h,j,e,e,i,g,h,g,Attack
```

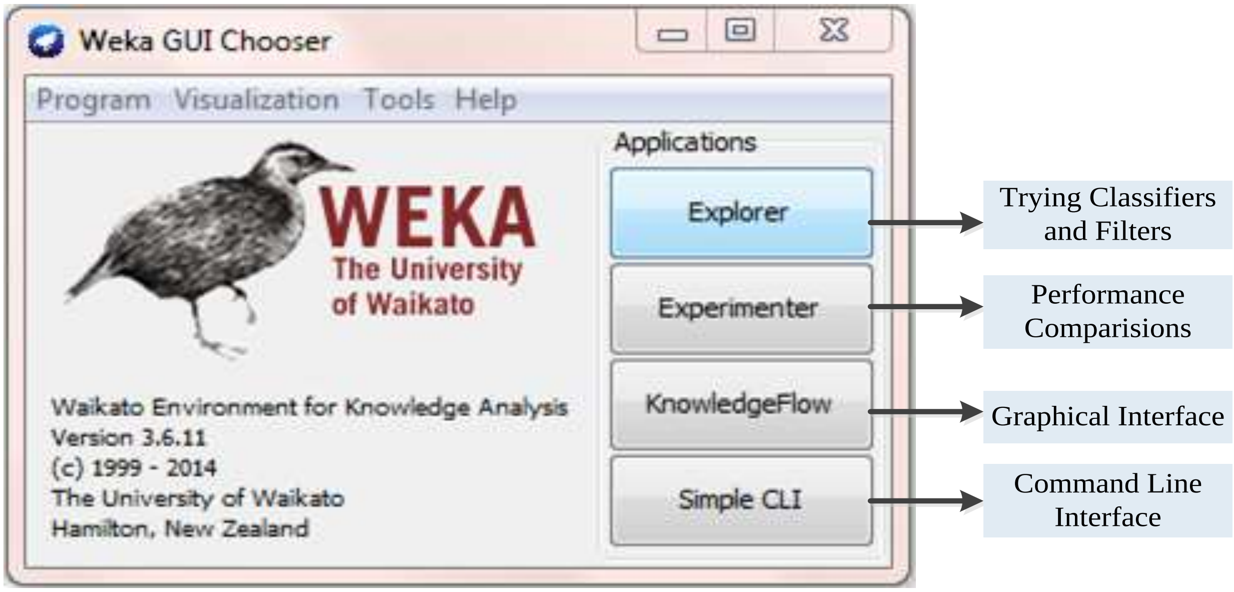
Attributes with any of the discretized values specified in brackets
Dependent or Class Variable
Missing Values
Data
File Saved as "file name.arff"

Figure 14: File format in .arff

Weka can be used for either of the following;

- To apply a learning technique on a dataset and analyze its output in order to learn more about the data in the dataset.
- Use learned models to generate predictions on newly seen testing instances.
- To apply several different learners and compare their performance in order to choose one for prediction.

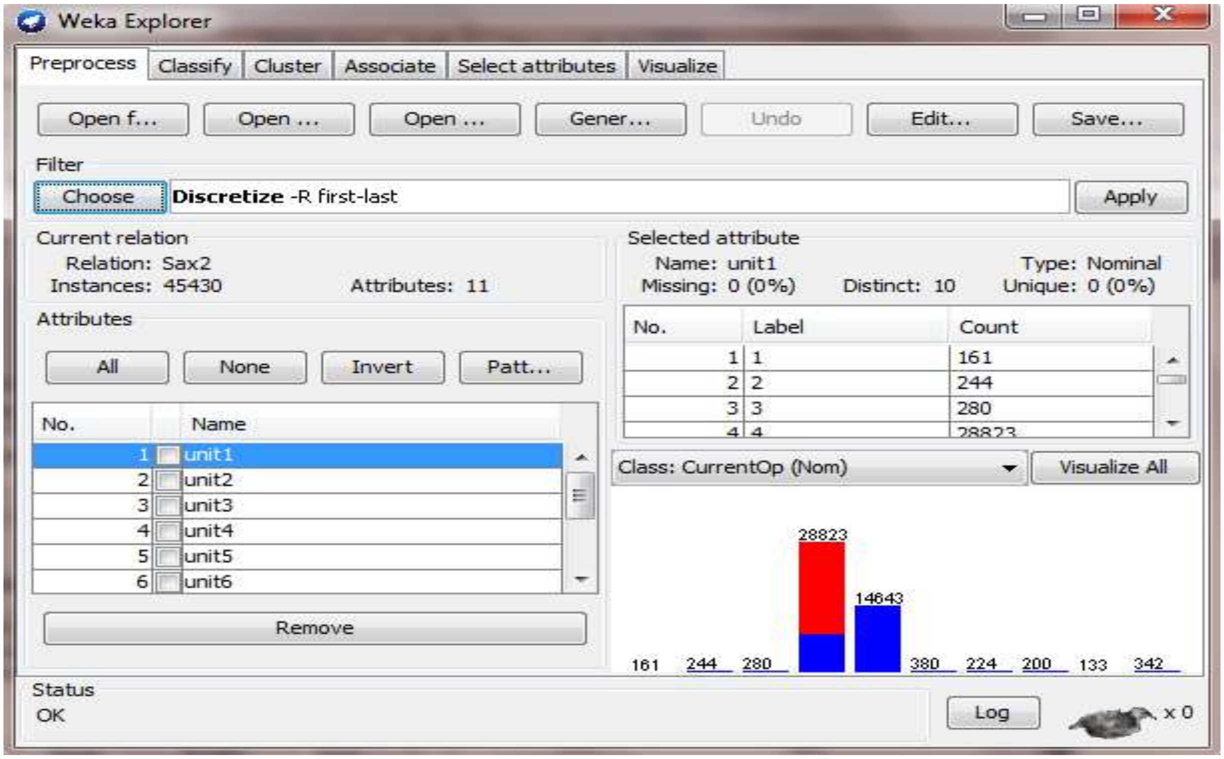
Weka can be easily used through the graphical user interface called *Explorer*. Fig.15 shows the GUI of Weka. The dataset shown earlier in Fig.14 can be loaded into the Explorer using the open commands.


Weka GUI Chooser

Waikato Environment for Knowledge Analysis
Version 3.6.11
(c) 1999 - 2014
The University of Waikato
Hamilton, New Zealand

Figure 15: WEKA front graphical user interface (GUI).

Once the dataset is loaded, the attributes and data in each attribute can be seen in the window as shown in Fig.16. As soon as the dataset is properly uploaded, the data mining algorithms in tabs classify, cluster, associate, select attributes and visualize, activate. There are two other graphical user interfaces to Weka. The Knowledge Flow interface allows you to design configurations for streamed data processing. A fundamental disadvantage of the Explorer interface is that it holds everything in main memory when you open a dataset, it immediately loads it all in. This means that the Explorer can only be applied to small- to medium-size problems. However, Weka contains some incremental algorithms that can be used to process very large datasets. The Knowledge Flow interface lets you drag boxes representing learning algorithms and data sources around the screen and join them together into the configuration needed. Knowledge flow interface is shown in Fig.17.



Filter

Discretize -R first-last

Current relation

Relation: Sax2
Instances: 45430 Attributes: 11

Selected attribute

Name: unit1 Type: Nominal
Missing: 0 (0%) Distinct: 10 Unique: 0 (0%)

No.	Label	Count
1	1	161
2	2	244
3	3	280
4	4	28823

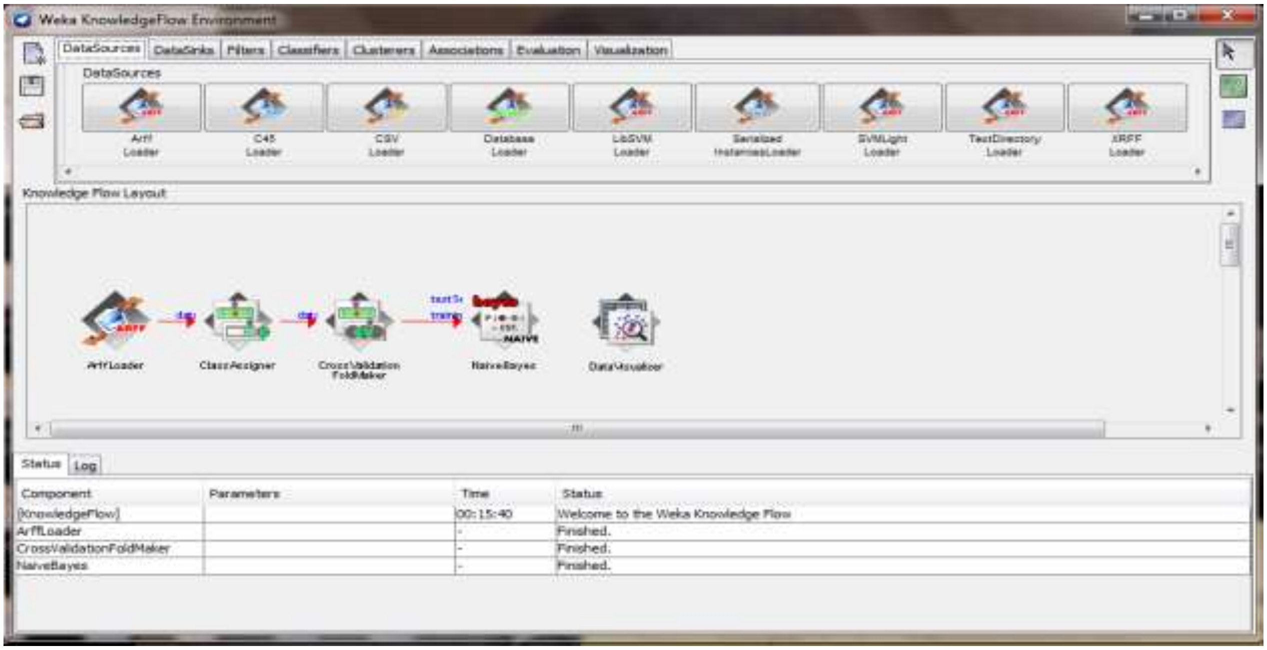
Attributes

No.		Name
1	□	unit1
2	□	unit2
3	□	unit3
4	□	unit4
5	□	unit5
6	□	unit6

Class: CurrentOp (Nom)

Status
OK

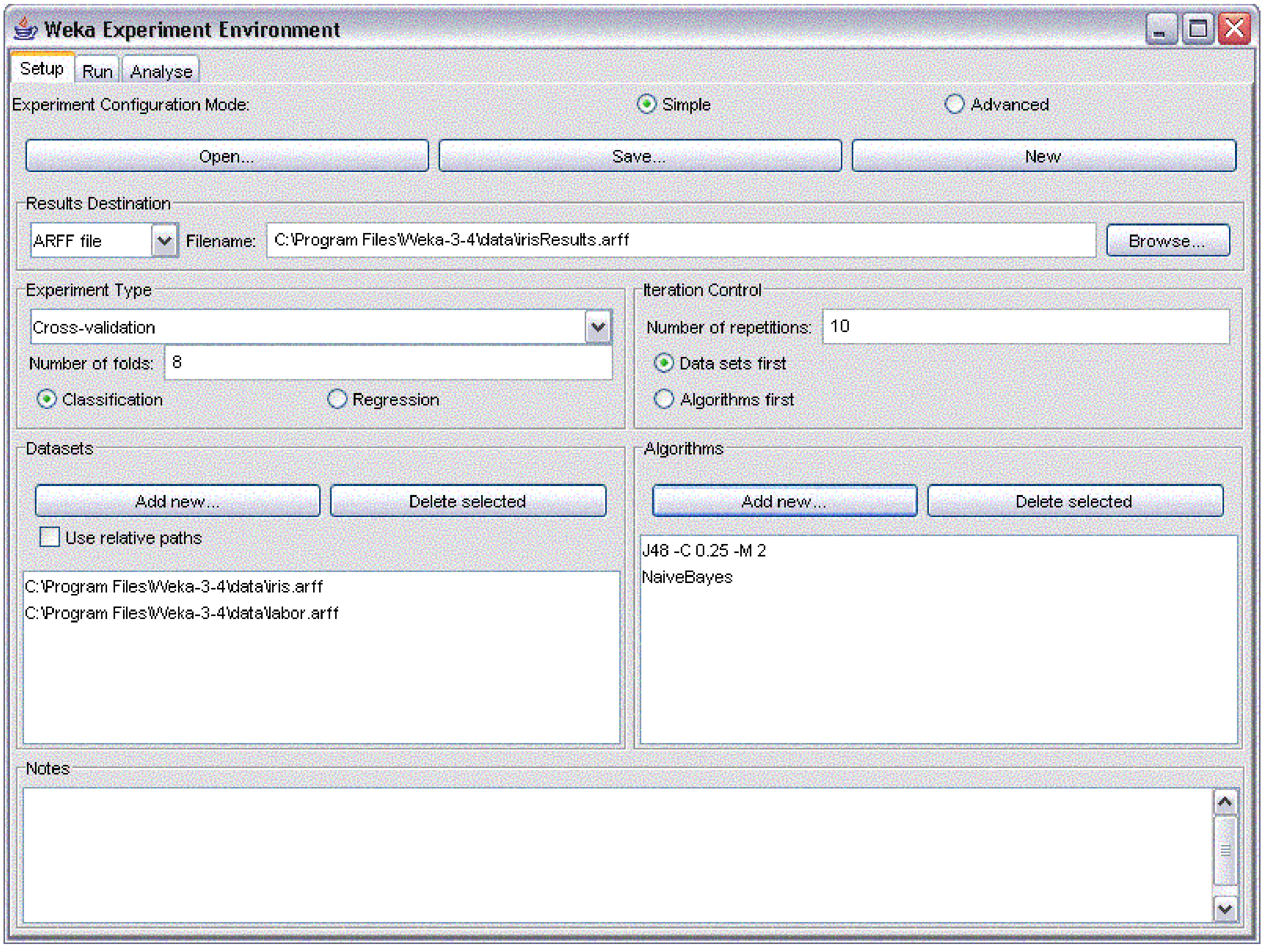
Figure 16: WEKA explorer graphical user interface (GUI).



Component	Parameters	Time	Status
[KnowledgeFlow]		00:15:40	Welcome to the Weka Knowledge Flow
ArffLoader		-	Finished.
CrossValidationFoldMaker		-	Finished.
NaiveBayes		-	Finished.

Figure 17: WEKA Knowledge flow environment with a model built.

Weka’s third interface, Experimenter, is designed to find methods and parameter values which work best for a given dataset and learning algorithms. Weka enables users to compare a variety of learning techniques using this interface. However, the Experimenter interface allows you to automate the process by making it easy to run classifiers and filters with different parameter settings on a corpus of datasets, to collect performance statistics, and to perform significance tests. Advanced users can employ Experimenter to distribute the computing load across multiple machines using Java remote method invocation (RMI). In this way, you can set up large-scale statistical experiments and leave them to run. Fig.18 shows the GUI of WEKA Experiment Environment.



Experiment Configuration Mode: ● Simple ○ Advanced

Results Destination: ARFF file Filename: C:\Program Files\Weka-3-4\data\irisResults.arff

Experiment Type: Cross-validation

Number of folds: 8

● Classification ○ Regression

Iteration Control: Number of repetitions: 10

● Data sets first

○ Algorithms first

Datasets:

C:\Program Files\Weka-3-4\data\iris.arff
C:\Program Files\Weka-3-4\data\labor.arff

Algorithms:

J48 -C 0.25 -M 2
NaiveBayes

Figure 18: WEKA Experiment Environment

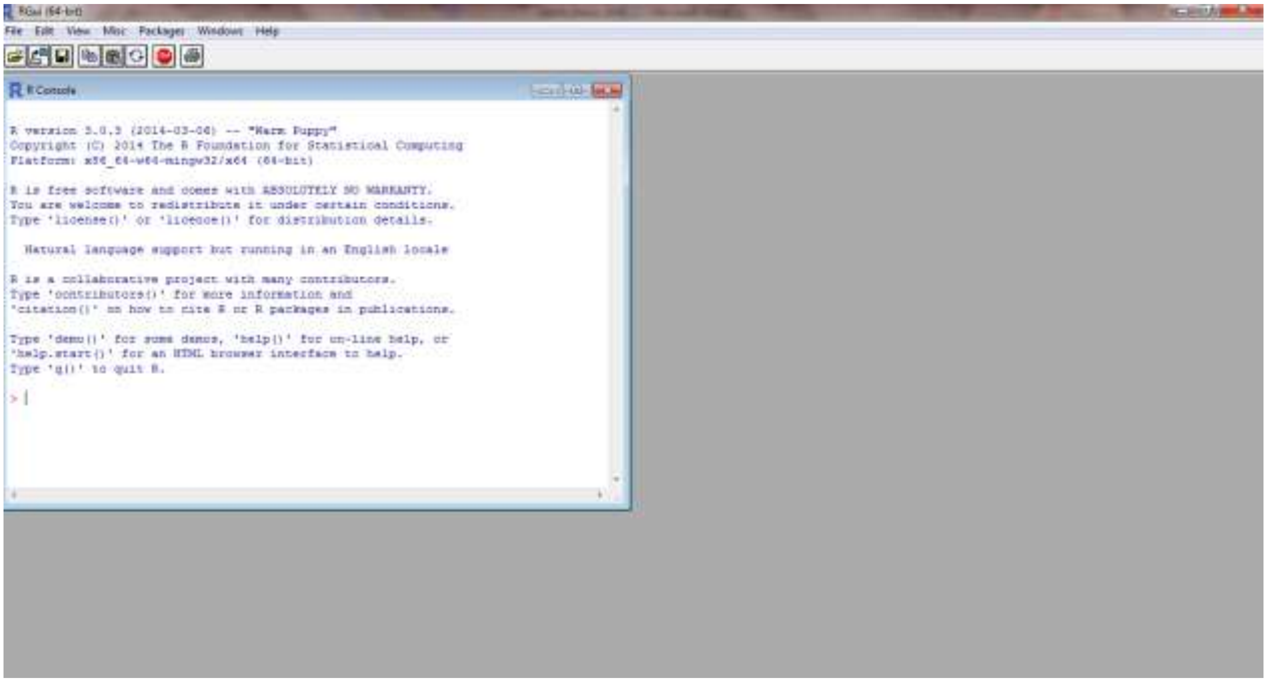
4.3 R Statistical Programming

R is an integrated *environment* of software facilities for data manipulation, analysis, statistics, calculation and graphical visualization. Following are the features of R;

- R has an effective data handling and storage facility,
- R has a suite of operators for calculations on arrays, in particular matrices,
- R has a large, coherent, integrated collection of intermediate tools for data analysis,
- R has graphical facilities for data analysis and data visualization

The term “environment” is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools. R is also a simple and easily adaptable platform for newly developing methods of interactive data analysis. R has developed rapidly over the years, and has been extended by a large collection of packages. However, most programs written in R are essentially ephemeral, written for a single piece of data analysis. R is distributed by the “Comprehensive R Archive Network” (CRAN) and is available at url: http://cran.r-project.org.

Fig.19 shows the GUI of R.



```
R version 3.0.3 (2014-03-06) -- "Warm Puppy"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

Figure 19: R graphical user interface.

4.3.1 Time Series Data Analysis with R

Since its introduction, R has gained wide range of users who can make use of the existing packages and also at the same time contribute new processes and techniques in the form of *add-on* packages. This contribution facility allowed R to continuously update with the new packages and also fabricated it into a ‘Black Box’ with thousands of capabilities.

One disadvantage or advantage of R, depending on the point of view, is that it can be used within a command–line interface, which imposes a slightly steeper learning curve than other software. But, once this burden has been taken, R offers almost unlimited possibilities for statistical data analysis [81]. The Data Analysis capabilities of R include time series component analysis, decomposition, discretization, reduction, smoothing, time series clustering using different distance measures, time series classification, regression and also visualization.

Firstly large sizes of databases can be imported into R in different ways and a few of them include importing ASCII and binary data, data from several other applications or even for database connections. One of the most used source for importing data into R is Microsoft Excel file. The dataset is stored as a matrix–object and in order to access particular elements of objects, square brackets ([ ]) are to be used. This Thesis uses the time series analysis, clustering and visualization capabilities. Highly integrated and timesore algorithms like Dynamic Time Warping are available in R. This thesis uses the DTW package in R to verify its concreteness and division capability when compared to other simple and easy clustering algorithms. Fig.20 shows DTW based and Euclidean distance based clustering of responses of different units in the system using both Euclidean and DTW measures.

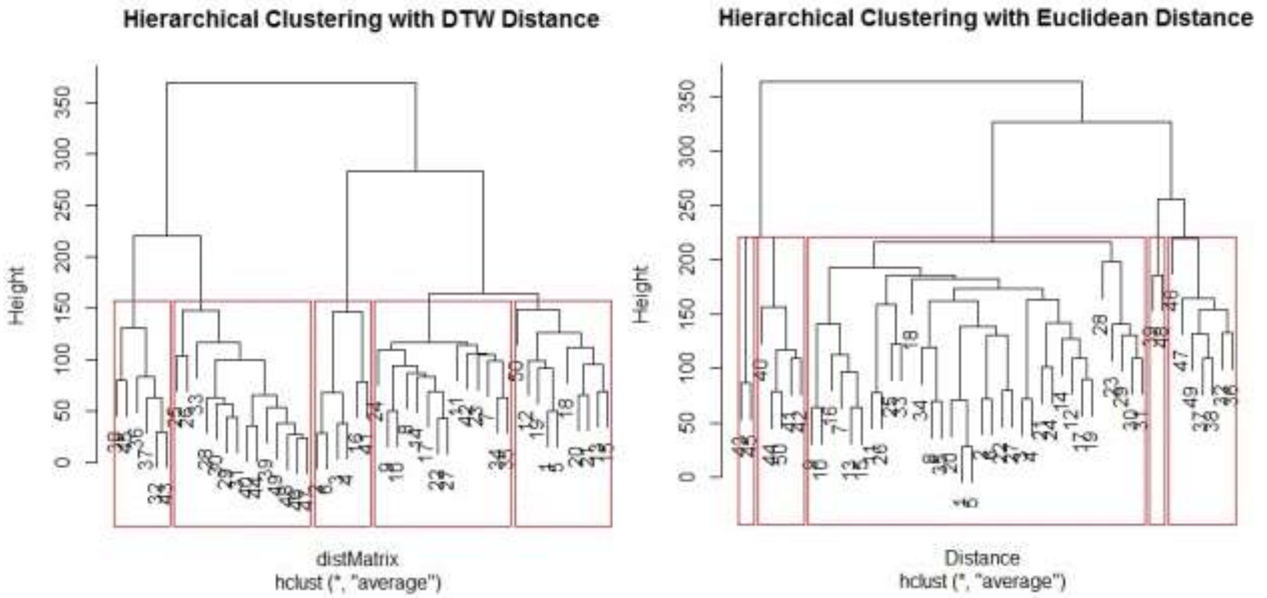


Figure 20: Time Series clustering of responses using Euclidean and DTW in R.

Chapter 5: SIMULATION AND RESULTS

This chapter discusses all the results of this research by introducing the test system used for the analysis first and then the simulation of proposed algorithms on the responses of the test system for different operating conditions. Interpretations are drawn from these resultant plots to back the proposed ideas and goals of this thesis work.

The proposed research is implemented on an IEEE 24 Bus reliability test system for real time attack detection, location identification, severity and criticality assessment problem. The responses of single phase voltage angles and apparent powers from a set of observable buses resembling PMU measurements are simulated for various remote trip injection attacks, faults and loading conditions.

These responses are used for database construction, severity and criticality assessment study as explained in previous chapters. To determine the real time accuracy and effectiveness of the proposed solution, the test system considered is tested in real time by considering the simulated outputs at a frequency of 30 samples per second resembling real time phasor responses. The majority parts of the proposed methodology are implemented in MATLAB and detailed explanation about the performance, accuracy, operational characteristics and real time implementation capability are explained in detail in this chapter.

5.1 IEEE 24 Bus Reliability Test System

Fig.21 shows the IEEE 24 bus Reliability Test System (RTS) used in this thesis to analyze the performance of the proposed methodology. The system constitutes of 11 Generator units and 38 branches connecting all the elements in the system. The bus data, line data and the results of optimal power flow are presented in [63].

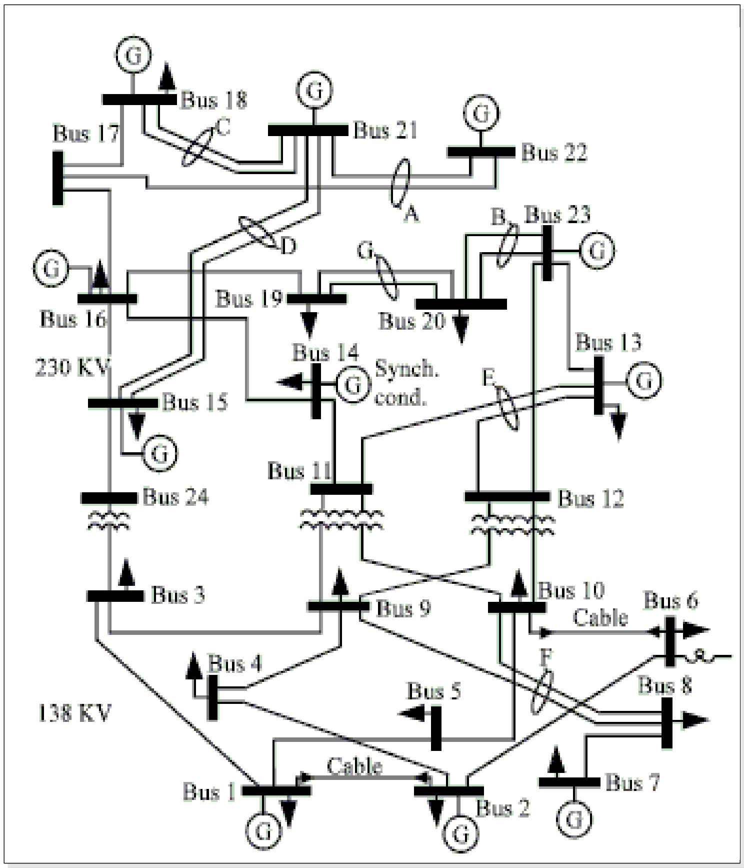


Figure 21: IEEE 24 Bus Reliability Test System

The measuring units are assumed to be located at each generator node in the system and the single phase voltage angle outputs of all these nodes are the ‘responses’ of measuring units. These voltage angles of all the generator units are being referred as ‘responses’ throughout this thesis.

5.2 Offline Analysis, Training Database Construction and Severity and Criticality Assessment

5.2.1 Training Database Construction

The process of database construction is the most tedious part in the entire research. Table 2 shows various events/attacks that are simulated in the system. After every event/attack mentioned, the responses (voltage angles) of the measuring units in the system are collected until the post disturbance responses settle down and the measuring units resumes its normal behavior.

A remote trip injection attack is considered as a removal of an element in the system without any initiating disturbances like faults. The elements mentioned here are primarily the branches connecting all the nodes in the system. Fig.22 shows the responses of all the units in the system for an attack on line 26-36 at t = 0.24 sec.

Training Database		
Subset	Event detail	Initiation
ATK	Remote tripping attack on elements connecting in the system	5 cycles, 18 cycles, 24 cycles
P3G	Three Phase to Ground fault at all the buses at which the PMU's are assumed and the buses adjacent to them	5 cycles, 18 cycles, 24 cycles, 40 cycles
N-k	Combination of the Events ATK, P3G, LL at the mentioned subset of buses till k=2,3	5 cycles
Instances Seen by the Algorithm in Real Time		
Subset	Event detail	Initiation
ATK, P3G, N-k	Random subset of Attacks/Events at all the elements as of the training database	3 cycles, 10 cycles, 12 cycles, 20 cycles

Table 2: Events that are simulated during training phase training databases and the ones that the proposed methodology encounters in real time stage.

In real time, attacks and events which are similar to the ones that are seen in the training database but with different clearing times, initiating angles, load and generation tripping times are sent to the proposed algorithm and the algorithm's performance is tested. The proposed research can be very accurate in identifying and locating attacks and dynamic events occurred at the nodes where measuring units are located and also the nodes adjacent to them. The size of the original responses obtained and their size when reduced using SAX and stored in database $\bar{C}_{\mu 1}^{N}$ is shown in Table 3.

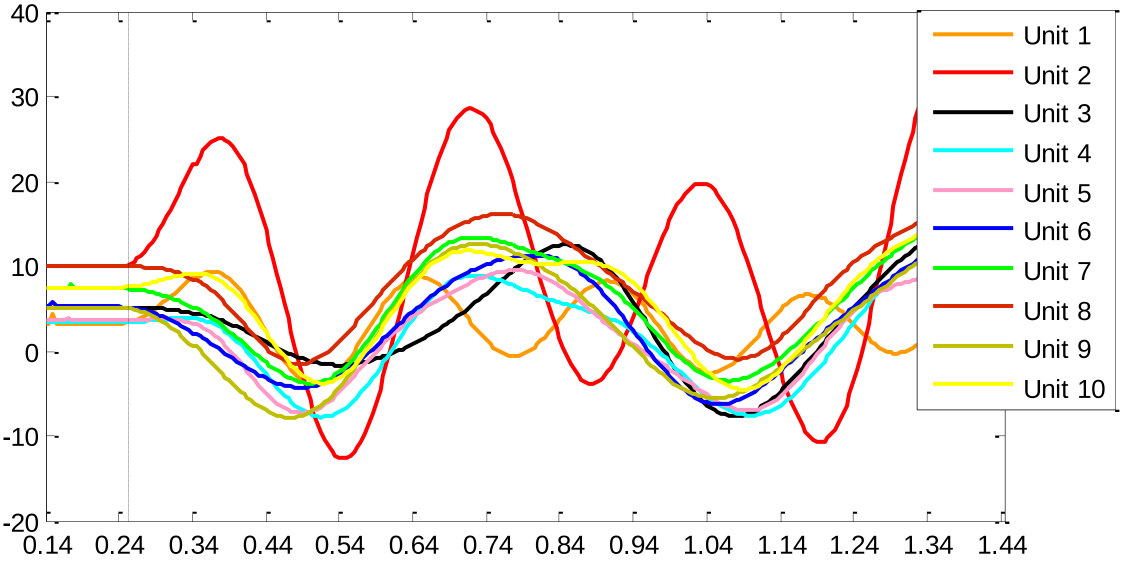


Figure 22: Responses of measuring units for an attack in the system

Number of Instances in original R_{μ}	61798
Number of Instances in $\bar{C}_{\mu 1}^{N}$	2433

Table 3: Number of instances in the Training Databases R_{μ} and $\bar{C}_{\mu 1}^{N}$

5.2.2 Parameter Selection for Symbolic Aggregate Approximation

As mentioned in Chapter 3, SAX is used twice in this thesis. Firstly, for the computation of probability terms in the Criticality equation. Here the continuous values of Change, Reach and Structural Criticality for all the elements are discretized and the frequency of occurrence of each SAX symbol is calculated to obtain the $\rho_{1_k}^{j}, \rho_{2_k}^{j}$ and $\rho_{3_k}^{j}$ terms. Secondly, SAX is used in the creation of the database $\bar{C}_{\mu 1}^{N}$ where the responses of the units for different disturbances and attacks in the system are discretized and stored for attack prediction.

For the first application mentioned, SAX is just used to discretize the different unique values of variables (Change, Reach and Structural Criticality) obtained for different attacks in the system and compute the frequencies of the SAX discretized ranges (symbols) and to determine the most common or frequent range and give suitable weight to the most frequent one. These ranges are not further used in any data mining applications. For this reason the values of SAX variables (β and *s*) are chosen randomly and the values chosen are given in the Criticality results section.

Whereas in the second case, the SAX discretized responses are analyzed using a data mining learner to predict if there is an attack in the system. Hence the values of SAX variables β and *s* in this case are to be chosen with proper care as the accuracy of the data mining learner on the discretized data will largely depend on these variables [56].

For this, different combinations of the SAX variables are considered. The training responses collected (as given in previous section) are discretized into symbolic strings with different values for variables β and *s*. The performance of DM learners on these different data sets is analyzed using the measures of accuracy, precision and recall. The *s* is chosen such that each cycle in the time series waveform is divided into equal number of segments. For example;*'s'* of 4 divides each cycle into 4 equal segments. Table 4 shows datasets created with possible outcomes of β (break points or BP) and sliding window length *s*.

Dataset Name	β	s	Dataset Name	β	S
BP_3_SL_4	3	4	BP_3_SL_4	3	14
BP_4_SL_4	4	4	BP_4_SL_4	4	14
BP_6_SL_4	6	4	BP_6_SL_4	6	14
BP_8_SL_4	8	4	BP_8_SL_4	8	14
BP_10_SL_4	10	4	BP_10_SL_4	10	14
BP_3_SL_6	3	6	BP_3_SL_6	3	16
BP_4_SL_6	4	6	BP_4_SL_6	4	16
BP_6_SL_6	6	6	BP_6_SL_6	6	16

BP_8_SL_6	8	6	BP_8_SL_6	8	16
BP_10_SL_6	10	6	BP_10_SL_6	10	16
BP_3_SL_8	3	8	BP_3_SL_8	3	18
BP_4_SL_8	4	8	BP_4_SL_8	4	18
BP_6_SL_8	6	8	BP_6_SL_8	6	18
BP_8_SL_8	8	8	BP_8_SL_8	8	18
BP_10_SL_8	10	8	BP_10_SL_8	10	18
BP_3_SL_10	3	10	BP_3_SL_10	3	20
BP_4_SL_10	4	10	BP_4_SL_10	4	20
BP_6_SL_10	6	10	BP_6_SL_10	6	20
BP_8_SL_10	8	10	BP_8_SL_10	8	20
BP_10_SL_10	10	10	BP_10_SL_10	10	20
BP_3_SL_12	3	12	BP_3_SL_12	3	22
BP_4_SL_12	4	12	BP_4_SL_12	4	22
BP_6_SL_12	6	12	BP_6_SL_12	6	22
BP_8_SL_12	8	12	BP_8_SL_12	8	22
BP_10_SL_12	10	12	BP_10_SL_12	10	22

Table 4: Different discretized datasets created with different values of β and s

These datasets created are tested with various DM learners and there performance is analyzed using different measures. The dataset that performed very well is picked, the β and s values of the picked dataset are chosen finally.

The performance measures used to compare the outputs of different data mining learners are as given as follows; Let a DM learner be a fault detector that detects the faults and signals an alarm when there is a fault. The confusion matrix of this learner can be seen in Fig 23.

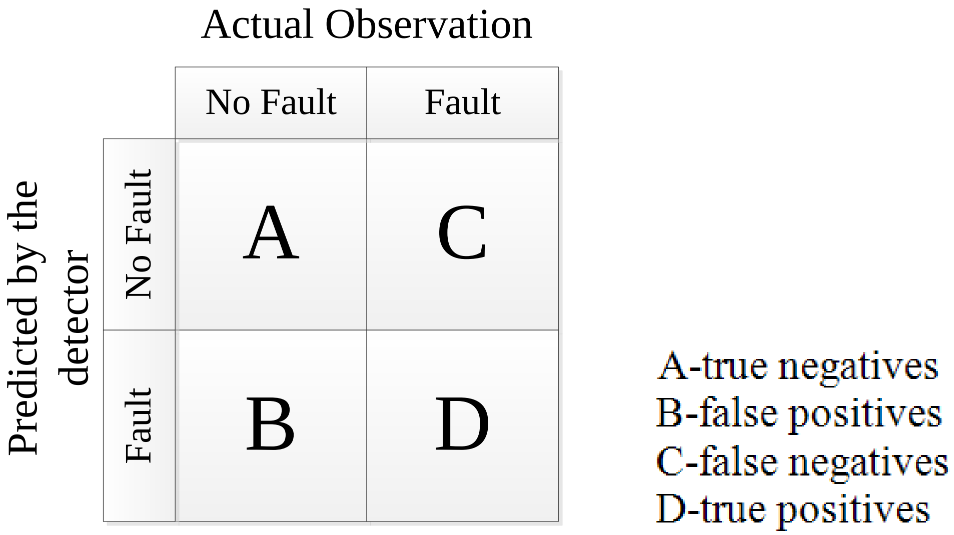


Figure 23: Confusion Matrix.

Using the confusion matrix, the measures of Accuracy, Precision and Recall can be defined as;

Accuracy is the proportion of true results in the population and can be given as (22)

$$Accuracy = \frac{A+D}{A+B+C+D} \tag{22}$$

Accuracy gives the ratio of actual number of correct predictions by the algorithm

Recall is proportion of actual positives which are correctly identified and can be given as (23)

$$Recall = \frac{D}{C+D} \tag{23}$$

Recall gives the ratio of total number of fault cases that are correctly identified as fault to the total number of fault cases present in the test database.

Precision is the proportion of the true positives against all the positive results

$$Precision = \frac{D}{B+D} \tag{24}$$

Precision gives us the ratio of cases where the detector correctly alarms when there are faults in the system to the total number of alarms given out by the detector.

High accuracy, precision and recall values depict the efficiency of a detector (data mining learner) in accurately predicting whether there is fault in the system (class label or dependent variable) for each instance in the testing data using the training database. Accuracy measure is not reliable when the number of occurrences or frequency count of dependent variables in the dataset is not uniformly distributed. Hence precision and recall measures are to be considered in that case. The performance of all the considered datasets on different data mining algorithms in predicting the dependent variable using a 10x10 cross validation (Given in Appendix 1) is shown below in Fig.24, Fig.25 and Fig.26.

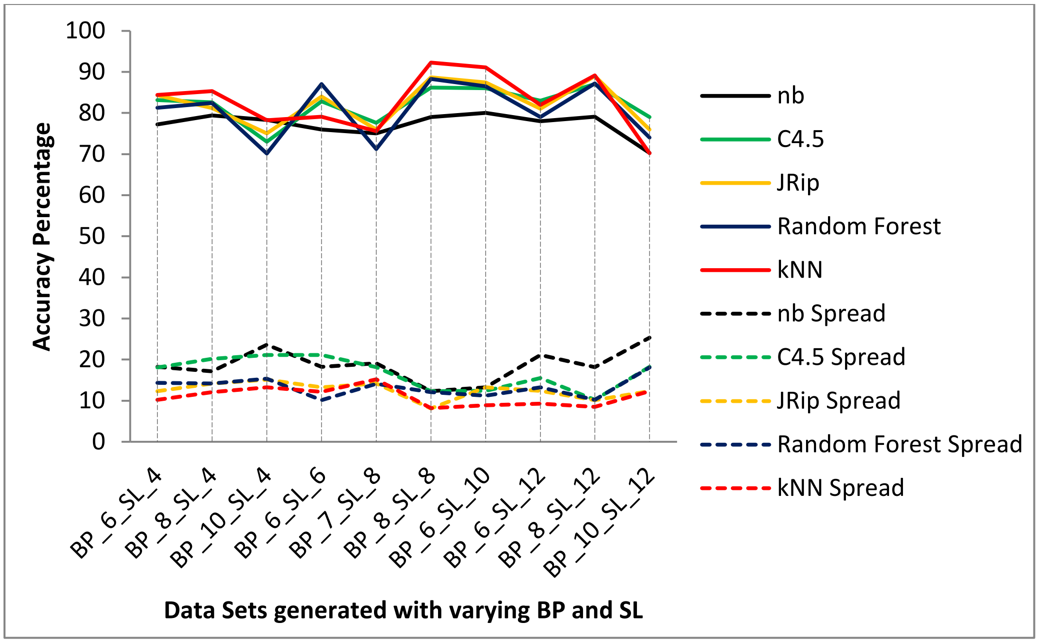


Figure 24: Median (Solid) and spread (75th-50th percentile, dotted line) values of accuracy for the data sets using 5 different learners for a prediction (dependent variable) 'Attack'. Note the X axis entries represent different data sets with varying sliding window length and breakpoints

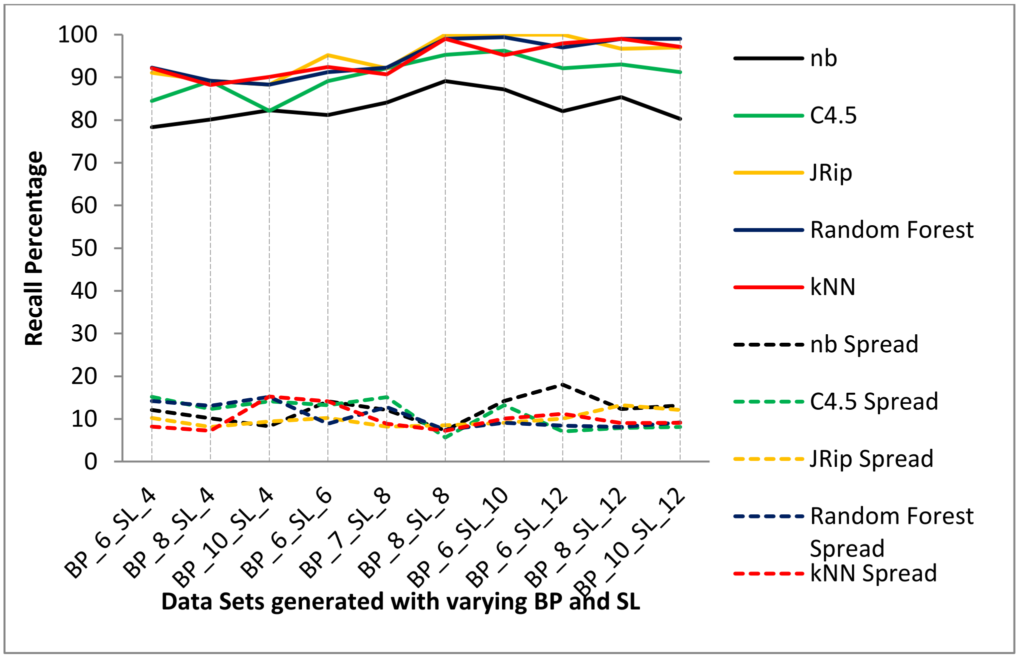


Figure 25: Median (Solid) and spread (75th-50th percentile, dotted line) values of Recall for the data sets using 5 different learners for a prediction (dependent variable) 'Attack'.

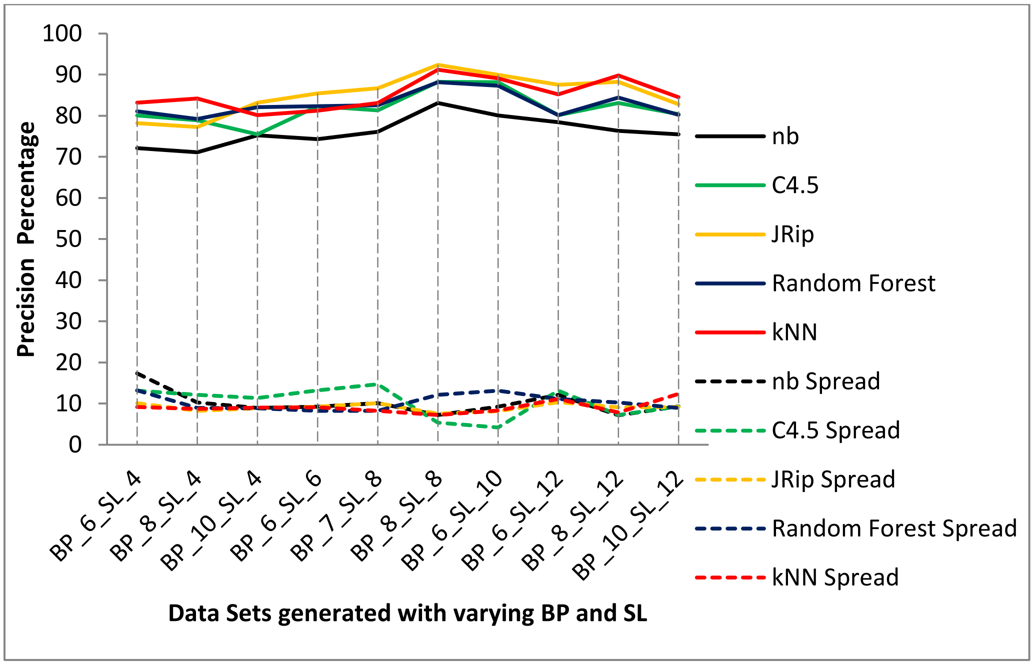


Figure 26: Median (Solid) and spread (75th-50th percentile, dotted line) values of Precision for the data sets using 5 different learners for a prediction (dependent variable) 'Attack'.

Fig.5-4, Fig.5-5 and Fig 5-6. show only the datasets with high values of accuracy, precision and recall.

Following inferences can be drawn from these results;

1) All the learners have decent values of accuracy, precision and recall for the datasets shown above.

2) Datasets BP_8_SL_8, BP_6_SL_10 and BP_8_SL_12 have high and almost equal values of accuracy, precision and recall. But when we see the spread (75th-25th percentile) for these datasets, BP_8_8 has minimum value of spread in all the cases and hence the variables in this dataset can be selected.

Sometimes it will be hard to come to conclusion of selecting a dataset, based on the median and spread values. At that time, statistical ranking techniques can be used to rank these datasets based on their cross validation results. [67] shows a similar approach where J12 ranking is employed to pick a dataset from various other datasets which all have equal values of median and spread.

5.2.3 Severity and Criticality Assessment

Severity and Criticality assessment is performed on the selected system for trip injection attacks at different locations in the system. The events and attacks are continued for N-3 contingencies and the results of severity and critical elements following an attack are stored. In the real time stage with the prediction of an attack, the severity of that particular attack is given out by selecting its respective severity from the stored analysis. The post attack critical elements in the system are also notified so that they can be properly monitored from preventing the attack in being a failure. The severity of an N-1 initiating attack on different system elements is shown in Fig.27.

The critical elements in the system are computed using the criticality index given in (6). For the probability terms in the criticality index, all the obtained continuous values for change, reach and structural criticality are discretized using SAX. The SAX variables used are β=4 and s=number of continuous values obtained for each parameter of Change, Reach and Structural Criticality. To explain this discretization process, let us consider initiating attacks occurred on different elements in the system except on element L 2-6. Now the change parameter of the element L 2-6 for initiating attacks on other considered (lines joining buses with PMU and the ones that are adjacent to these buses) elements in the system can be shown in Table 5.

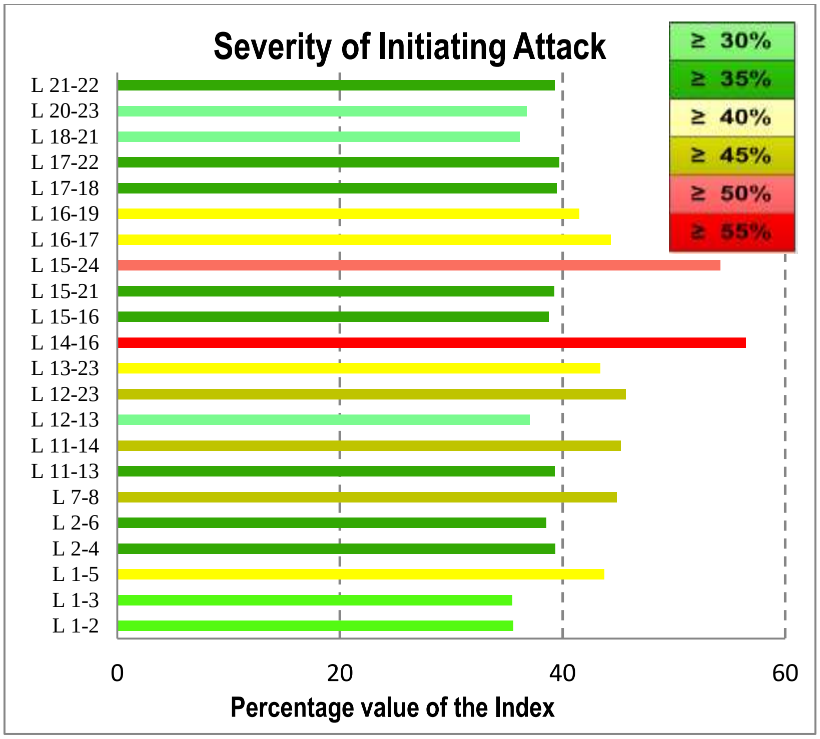


Figure 27: Severity of the initiating attacks on different system elements (branches), Y axis shows the elements in system, for example L 21-22 represents a branch joining buses 21 and 22.

Attack i on	Change in L2-6 for i	After Sax discretized Symbols	Frequency of each Symbol	Probability $\rho_{2_1}^{L\,2-6}$
L 1-2	0.2156	b	a=4	4/21
L 1-3	0.0921	a		
L 1-5	0.6913	d		
L 2-4	0.2702	b		
L 7-8	0.7213	d		
L 11-13	0.0813	a		
L 11-14	0.2213	b	b=8	8/21
L 12-13	0.0217	a		
L 12-23	0.6573	d		
L 13-23	0.6932	d		
L 14-16	0.7432	d		
L 15-16	0.1834	b		
L 15-21	0.3215	b	c=3	3/21
L 15-24	0.6091	c		
L 16-17	0.5632	c		
L 16-19	0.5121	c		
L 17-18	0.2856	b		
L 17-22	0.6712	d	d=6	6/21
L 18-21	0.2312	b		
L 20-23	0.2477	b		
L 21-22	0.0212	a		

Table 5: Computation of probability terms in Criticality Index

The criticality of elements connected following initiating N-1 attacks on any of the system elements are given in Fig.28.

Index	1	2	3	4	5	6	7	8	9	10	11
Element	L 1-2	L 1-3	L 1-5	L 2-4	L 2-6	L 7-8	L 11-13	L 11-14	L 12-13	L 12-23	L 13-23
Index	12	13	14	15	16	17	18	19	20	21	22
Element	L 14-16	L 15-16	L 15-21	L 15-24	L 16-17	L 16-19	L 17-18	L 17-22	L 18-21	L 20-23	L 21-22

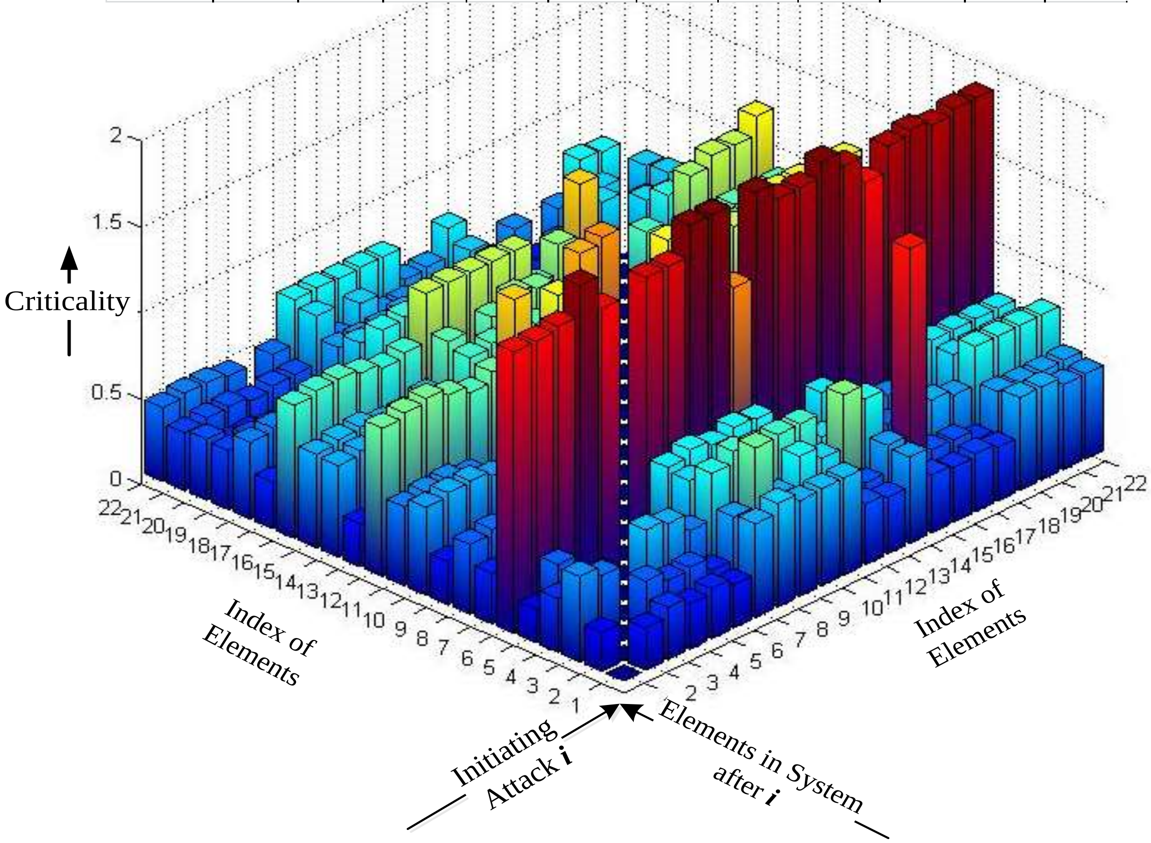


Figure 28: Criticality of the lines following an initiating event. X axis shows the initiating events, Y axis gives the elements in the existing system following the initiating attack and Z axis shows the criticality of the existing elements based on the criticality index formulated.

5.3 Online Incremental Clustering, Model Construction and Database Update

5.3.1 Online Incremental Clustering

The proposed methodology uses an online incremental clustering technique to cluster the measuring units and select cluster representatives to monitor the events in the system. The minimum number of samples necessary to start the splitting and aggregation of existing clusters is given by the value n_{min}. The n_{min} value chosen in this research is 2 cycles. Hence after every two cycles of new data arrival, the proposed methodology tests for splitting and aggregating the existing clusters and presents the new clusters along with their new centroids and diameter pivots. The elements closer to the centroids and the new pivots are chosen as representative elements for all the measuring units. The cluster hierarchy for a simulation with 3 attacks in the system is shown below in Fig.29.

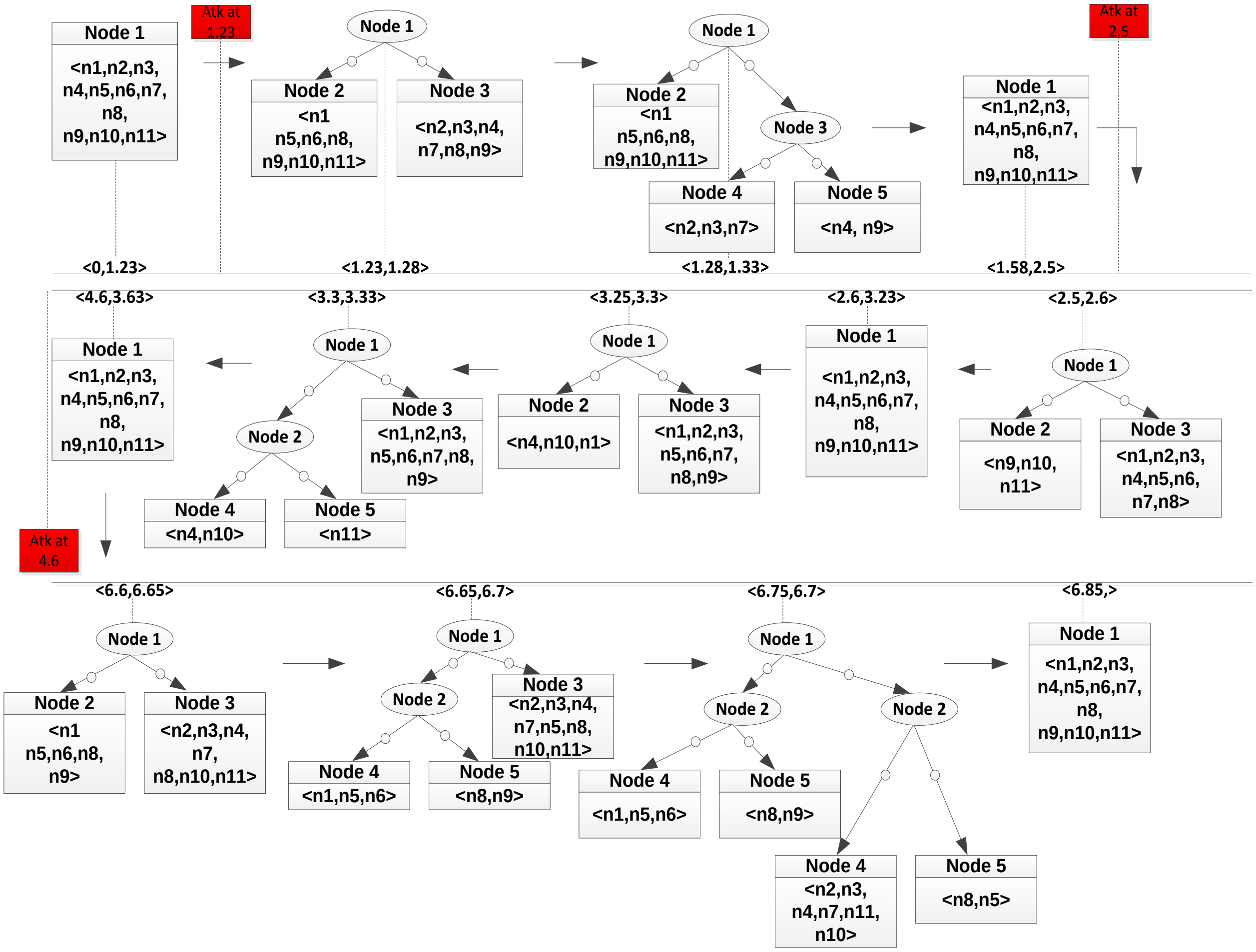


Figure 29: Online Incremental Clustering.

5.3.2 Online Database and Model periodic Update

Here the databases (R_μ , Y1) and $\bar{C}^N_{\mu 1}$ are periodically updated with the current instant responses seen by all the measuring units in the system for the last 2 cycles. The accuracy of the disturbance prediction model in correctly identifying the disturbance in the system is always monitored and if the accuracy fall be a threshold (Here Prediction Accuracy of Disturbance in System<=80%) the random forest model is again built on the database (Rμ, Y1) that is updated with the recent responses of the units in the system. This process of monitoring prediction accuracy and updating trees in the forest will continue until the real time simulation comes to an end.

5.4 Real Time Disturbance Prediction, Location Identification and Attack Prediction

5.4.1 Real Time Disturbance Prediction and Location Identification

Real time predictions about disturbance in the system are performed using the individual decision tree models built on the database Rμ using a random forest classifier. In the prediction stage, trees which are built only on the cluster representative attributes are selected and the decision about disturbance in the current instant is made using the responses obtained in the real time. The performance of this classifier in real time stage for a 10 minute interval is presented in Table 6 based on the values of accuracy, precision and recall. Once a disturbance is identified, its probable location is the cluster which has the maximum value of variance measure as given in (13). The disturbance location accuracy using the variance index is presented in Fig.30 where change in accuracy of the prediction with the variation in number of units in each cluster can be seen.

<table>
<tr><td colspan="2">Number of Disturbances seen During the Interval</td><td colspan="2">Number of Disturbances Correctly Classified</td></tr>
<tr><td colspan="2">12372</td><td colspan="2">12124</td></tr>
<tr><td>Accuracy</td><td>Precision</td><td>Recall</td><td>Out of Bag Error</td></tr>
<tr><td>97.99</td><td>98.91</td><td>98.91</td><td>0.00187</td></tr>
</table>

Table 6: Performance of the random forest classifier in identifying disturbances in the system in real time simulation

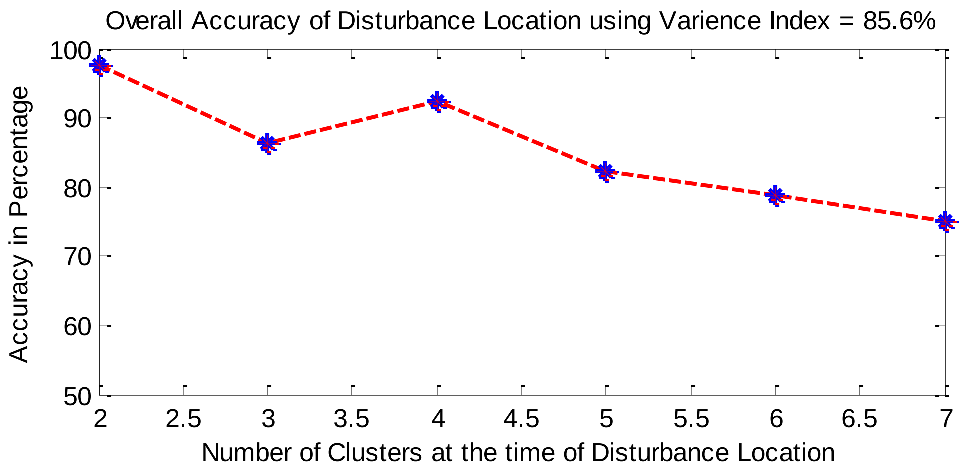


Figure 30: Accuracy of Cluster Location Index with change in Number of Clusters in Online Stage. (Accuracy=Number of Instances where attack location origin is correctly identified / Overall number of time the predictions are made). Overall Accuracy is the average of the obtained observations

The observations in Fig.30 are collected for the 10 minute simulation where different attacks and events occurring in the system for that duration and the cumulative average of accuracy of predictions occurred is considered with respect to the number of units in the each cluster while the decision about the disturbance location is made. The accuracy of the prediction location decreases with increase in the number of cluster divisions.

5.4.2 Real Time Attack Detection

As stated in section IV, the process of attack detection initiates after the location of the disturbance is identified. The inputs for this step are the discretized and N-gram represented real time responses of all the measuring units in the disturbance cluster and of all the cluster representatives. Selection of the k value is also an important step in this process as with different values of k, the prediction accuracy varies tremendously. The k value is also selected by a 10x10 cross validation on the offline database $\bar{C}_{\mu 1}^{N}$ with different values of k.

The performance of the kNN algorithm in detecting the remote trip injection attacks with different similarity measures is given in Table 7 where the results are presented in the form of accuracy, precision and recall of the algorithm for that duration. On observing the results, one can come to the conclusion that kNN with DTW as distance measure performed very well in terms of all accuracy, precision and recall measures but this algorithm is computationally more tiresome (took comparably more time) when compared to the other two measures. The kNN with Euclidian similarity has very less results of accuracy, precision and recall. This is solely because of the fact that Euclidean considers just point to point distance between two strings. Hence even if two similar strings that are shifted slightly on time axis will appear as two different string but this will not happen in case of DTW similarity measure.

Distance Measure	Accuracy	Precision	Recall	Out of Bag Error	Average time for each prediction (s)
DTW	94.87	91.13	97.5	0.0812	2.15e-2
Edit	93.41	85.2.	96.4	0.1854	12e-3
Euclidean	83.19	87.23	81.7	0.6241	1.7e-3

Table 7: Performance of Different Distance Measures in Detecting Attacks using kNN.

The accuracy of the attack prediction also varied with the number of measuring units in the disturbance cluster. This is due to the fact that with increase in attributes or measuring units that are to be tested, the conflicts or ties such as equal number of the measuring units predicting that there is an attack in the system while other measuring units behaving as if there is 'no attack' in the system increases. These conflicts or ties between two output predictions decrease the accuracy of the classifier. This decrease in the accuracy with the increase in the number of measuring units in the disturbance cluster can be seen in Fig.31.

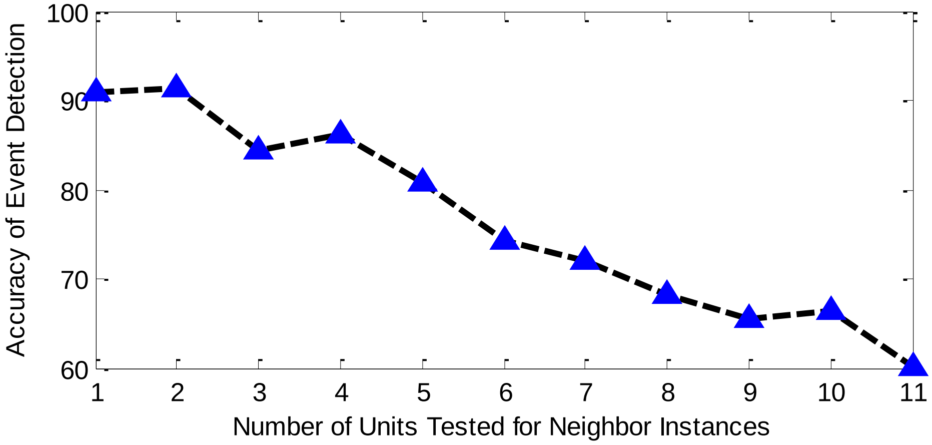


Figure 31: Accuracy of kNN-DTW based Attack Detection with varying number of units searched for Neighbor Instances.

5.4.3 Malicious Data Attack Detection

As mentioned earlier, the proposed Malicious data Detection is not performed in real time. Here, three attack cases are considered with variable number of units compromised. Compromised units contain the malicious responses of an attack in the system where the other units in the system showing their genuine normal responses. Naïve Bayes classifier is used to predict the responses of the cluster representative attributes in the disturbance cluster. The prediction algorithm starts when the ξ of the prediction comes to be lower than the threshold considered. The ξ selected here is such that, at least equal number or majority of representative units belonging to other clusters should also predict that there is an attack in the system for current instant responses. If ξ falls below that, the particular attack discretized subsamples are sent in this process. Fig.32, shows an attack scenario where the responses of units at buses 21 and unit 22 are spoofed to that of an attack on line 21-22. The Random Forest Classifier detected a disturbance in the system for these responses and identified the probable disturbance location as a cluster which includes buses 21, 22 and ξ as 0.2. The responses of these two units are predicted using other measuring units as attributes. The DTW distance between the predicted and observed is also shown in Fig. 32. The λ upper bound variable is selected by rigorous analysis, performed on training database with different sets of spoofed measurements. λ is selected such that if at least 65% of the predicted discretized samples are different to that of the actual ones, we can state that the observed responses are from the compromised.

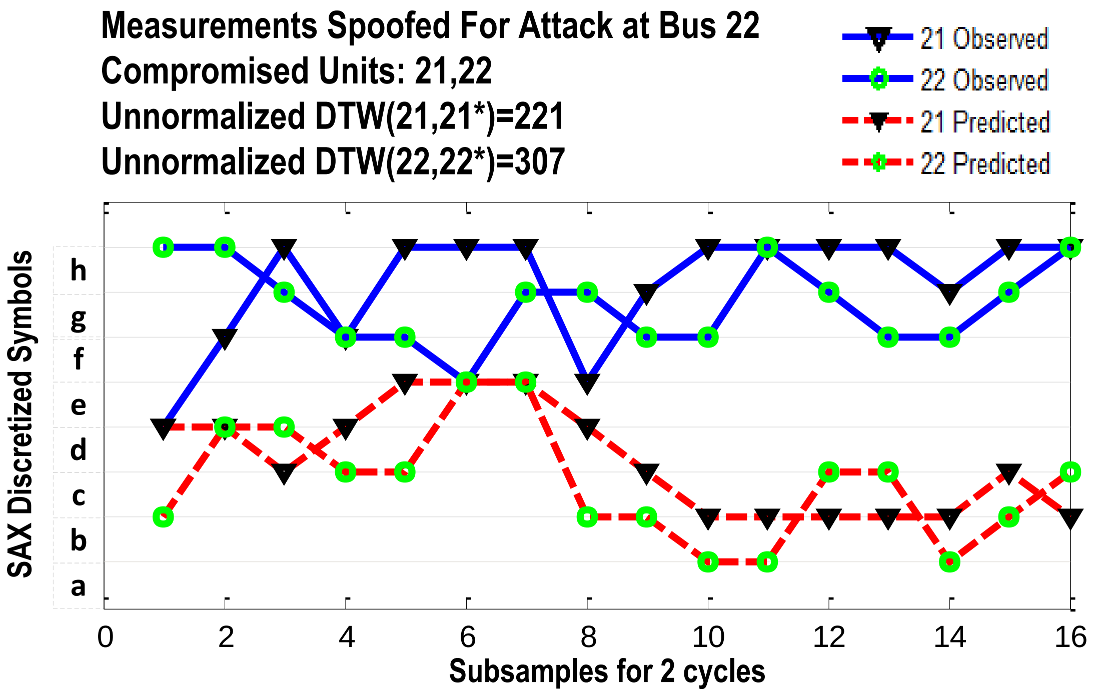


Figure 32: Malicious data attack at bus 22 with two units compromised. Predicted (red) and Observed (blue) responses with their DTW distance.

In Fig.32 we can notice that the DTW distance between the predicted and actual is very high and also the difference in discretized symbols of predicted and observed for units 21 and 22 exceeds the upper bound specified. Hence the responses of these two units are considered as malicious data.

Now a new scenario is considered for the same example in Fig.32. The only difference between this scenario and the previous one is that, along with units 21, 22 a third unit is also considered giving malicious or spoofed responses that represent an attack at Bus 22. In this case, the units compromised are Buses 18, 21 and 22. The same approach shown in Fig 5-12. is performed again and Fig.33 in the next page shows the predicted responses of units at buses 18, 21 and 22 by considering other cluster representative units as attributes.

From Fig.33 we can observe that with the increase in the number of units compromised, the DTW distance between observed and predicted decreased compared to the predictions in Fig.32. This is because with the increased number of units stating that there is an attack in the system, the probability of predicting attack responses by other units also increases. However, in this case, the malicious data attack is detected as the difference between discretized symbols for observed and predicted for all the three units exceeds the threshold selected.

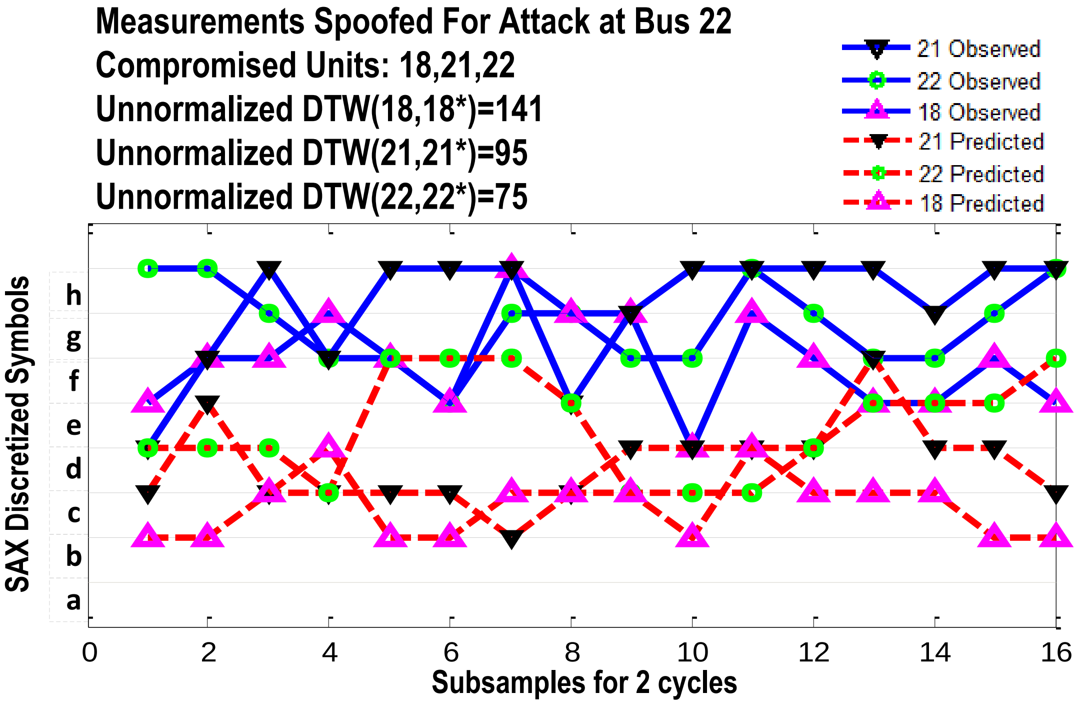


Figure 33: Malicious data attack at bus 22 with three units compromised. Predicted (red) and Observed (blue) responses with their DTW distance

Chapter 6: CONCLUSION

6.1 Discussion

6.1.1 Resolving Data Mining Ties

Ties often occur in data analysis and mining algorithms when a majority selection based prediction is employed. In the proposed algorithm, the location identification and real time attack detection stages makes a prediction based on the output of the majority of the (attributes) measuring units. Detailed explanation is as follows; the location identification stage computes the change in variance in the current instant responses of all the units in rep and concludes that the cluster with highest variance value is the disturbance cluster. This approach sounds good when there is a large dispersion in the values of variances for all the units in rep. What if; two cluster representative units belonging to different clusters have equal maximum variance value? In such case selecting a disturbance cluster out of those two clusters will be very difficult as a wrong selection may deter the prediction accuracy of this stage. Also the attack prediction stage requires the responses of units in disturbance cluster as inputs, hence wrong disturbance cluster selection can result in wrong attack predictions.

In order to deal with this issue, intuitive tie resolving approaches are proposed in [74]. Nonetheless, the proposed research uses a simple 'Next Best Approach' to resolve ties by selecting the next highest value or next best prediction given by majority of the units in the system. This can be performed as follows;

1) Tie in Location Identification Stage: As stated earlier, if the maximum variance is equal for two cluster representative units belonging to different clusters, these two clusters are selected and the second maximum variance value in these two selected clusters is chosen. The cluster with this next largest variance value is selected as the location of the disturbance.

2) Tie in Attack Prediction Stage: Ties in attack prediction stage occurs when equal number of measuring units in the disturbance cluster including the cluster representative measuring units predict two different outputs (<attack, no attack>) for the current instant responses. These ties are resolved by choosing the prediction made by the units which has the minimum value of the aggregate similarity measure. If there is

a tie again between these two minimum values which is not seen during our experimental evaluation, the prediction about attack in the system could not be made and the responses seen by the measuring units are to further examined which is one of the limitation of the proposed approach.

6.1.2 Strengths and Limitations of the Proposed Real Time Attack Detection Methodology

The online clustering employed in this algorithm continuously monitors the existing clusters in the system and the usability of the proposed approach is very high as the clustering used is hierarchical and does not need any predefined number of target clusters. Hence continuous monitoring of the measuring units is performed easily and more appropriately which is not the case in [10] where clustering is performed on stationary data.

The severity and criticality assessment schemes improve the situational awareness of the operator when an attack is detected. SAX reduces the dimensionality and size of the data generated and hence it increases the computation efficiency of the attack prediction stage. Strong examination is also performed by using tedious cross validation based trial and error approaches to select each and every parameter or variable used in this thesis instead of just assuming some random values.

Finally the major asset of the proposed algorithm is its applicability to adapt to any kind of fast streaming data based detection or prediction problem. The same algorithm with different set of instances in the training database can be used to detect different disturbances such as voltage flicker, low frequency harmonics, sags, swells and lot more. However with all these strengths, the biggest limitation of this approach is its data centric nature. With newly seen data samples coming from the system for new topologies and system operational parameters, the efficiency of the proposed approach decreases tremendously or this approach may not even work. The storage requirements is also one of the major limitation of the proposed algorithm as large number of discretized samples for different system conditions needs to be stored or collected in the training database to make the predictions in real time. These limitations of adapting to the new data with limited storage requirements are our goals that are to be achieved in the future.

6.2 Conclusions

In this thesis, a novel data mining based attack detection methodology is proposed to identify and locate data injection attacks at different locations in the system. A severity and criticality assessment scheme is formulated to analyze the severity of the attack detected and identify the post attack critical elements in the system that are to be protected to restrict the detected attack in propagating to a large failure. The online clustering methodology employed here improves the real time performance of the proposed methodology by allowing it to monitor only few selected units, out of all the units deployed in the system. Finally a prediction based malicious data attack detection technique is proposed to differentiate malicious data attacks with real attacks in the system. Listed real time evaluation results show the credibility and real time applicability of the proposed approach. Finally the limitations of this approach are also presented to give an insight for interested readers who are willing to extend the presented work.

Chapter 7: References

1. Xiao-yun Zhang, Ying Wu, “Power System Voltage Stability Analysis Based on Data Mining”, In Proceedings of Advances in Intelligent and Soft Computing Volume 62, 2009, pp 1685-1691.
2. CE Cheng, “Power System Online Stability Assessment using Synchrophasor Data Mining,” PhD Dissertation, Texas A&M University, 2014.
3. M. Kezunovic, A. Abur "Merging the temporal and spatial aspects of data and information for improved power system monitoring applications," IEEE Transactions, Vol. 9, Issue 11, pp. 1909-1919, 2005.
4. C. Zheng, Y. Dong, O. Gonen, M. Kezunovic "Data integration used in new applications and control center visualization tools," In Proceedings of IEEE Power and Energy Society General Meeting, Minneapolis, USA, July 2010.
5. Kejun Mei “Dynamic event identification and wide-area response based control in large systems,” PhD Dissertation, Purdue University, 2009.
6. N.G Bretas and L.F.C Alberto, “Coherency on electrical power systems,” In Proceedings of PowerConn 2014.
7. A. Srivastava, T. Morris, T. Ernster, C. Vellaithurai, Shengyi Pan, U.Adhikari “Modeling Cyber-Physical Vulnerability of the Smart Grid With Incomplete Information,” IEEE Transactions, Vol. 4, Issue 9, 2013.
8. Yao Liu, Peng Ning, Michael K. Reiter “False Data Injection Attacks against State Estimation in Electric Power Grids,” Proceedings of 16th ACM Conference, 2009
9. L. Chuang, C. Wang, H. Lee, M.-Y. Liu, Y.T. Hsiao, J.-A. Jiang, “An adaptive routing algorithm over packet switching networks for operation monitoring of power transmission systems,” IEEE Transactions on Power Delivery 25 (2010) 882–890.
10. "What Caused the Power Blackout To Spread So Widely and So Fast?". Genscape. August 15, 2003. Retrieved July 13, 2011.
11. Sandip Patel, Jigish Zaveri, “A Risk-Assessment Model for Cyber Attacks on Information Systems,” Journal on Computers, Vol.5, Issue 3, 2010
12. Rob Sullivan, “ Introduction to Data Mining for Life Sciences,” Chapter 12, 2012
13. http://www2.cs.uregina.ca/~dbd/cs831/notes/kdd/1_kdd.html
14. Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, “ From Data Mining to Knowledge Discovery in Databases,” Article Published in American Association for Artificial Intelligence, 1996

15. http://www.rithme.eu/?m=resources&p=kdprocess&lang=en
16. http://en.wikipedia.org/wiki/Data_reduction
17. Joannes Vermorel, “Overfitting: when accuracy measure goes wrong,” Published in 2009
18. James Dougherty, Ron Kohavi, Mehran Sahami, “Supervised and Unsupervised Discretization of Continuous Features,” Proceedings of the 25 International Machine Learning Conference, 1995
19. Rajashree Dash, Rajib Lochan Paramguru, Rasmita Dash, “Comparative Analysis of Supervised and Unsupervised Discretization Techniques,” International Journal of Advances in Science and Technology, Vol. 2, No. 3, 2011.
20. Elena S. Dimitrova, John J. McGee, Reinhard C. Laubenbacher, “Discretization of Time Series Data,” Published in Journel of Micobiology, 2010
21. Almahdi M. Ahmed, Azuraliza Abu Bakar, Abdul Razak Hamdan, “A Harmony Search Algorithm with Multi-pitch Adjustment Rate for Symbolic Time Series Data Representation,” Published in I.J. Modern Education and Computer Science Journal, 2014
22. P. Motamarri, M. R. Nowak, K. Leiter, J. Knap, V. Gavini, “Higher-order adaptive finite-element methods for Kohn-Sham density functional theory,” http://arxiv.org/pdf/1207.0167.pdf.
23. Carlo Cattani, Armando Ciancio, “Wavelet Clustering in Time Series Analysis,” Balkan Journal of Geometry and Its Applications, Vol.10, No.2, 2005
24. J. Lin, E. Keogh, S. Lonardi, J.P. Lankford and D.M. Nystrom, “Visually Mining and Monitoring Massive Time Series,” In proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, WA, Aug 22-25, 2004.
25. http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf
26. Zhan Yong, Cheng Haozhong, Ding Yifeng, “S-Transform based classification of power quality disturbance signals by support vector machines,” Proceedings of the CSEE. (2005) 51-56.
27. Preheim SP, Perrotta AR, Martin-Platero AM, Gupta A, Alm EJ, “Distribution-based clustering: using ecology to refine the operational taxonomic units,” Journal of Microbiology, 2013
28. B.J. Hand, W.E Henley, “Statistical Classification Methods in Consumer Credit Score: a Review,” Journal in Statistics, 1997
29. Ian Dent, Tony Craig, Uwe Aickelin, Tom Rodden, “Variability of Behavior in Electricity Load Profile Clustering; Who Does Things at the Same Time Each Day?,” 2014 http://arxiv.org/pdf/1409.1043.pdf
30. S. Hemamalini, Sishaj P Simon, “Dynamic Economic Dispatch with Valve-Point Effect Using Maclaurin Series Based Lagrangian Method,” International Journal of Computer Applications, Vol 1, No. 17, 2010

31. Wei-Yin, Loh, "Overview of Classification and Regression Trees," http://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf.
32. Ruisheng Diao, Kai Sun, Vijay Vittal, Robert J.O'Keefe, Michael Richardson, Navin Bhatt, Dwayne Stradford, Sanjoy K.Sarawgi, "Decision Tree Based Online Voltage Security Assessment using PMU Measurements", IEEE Transactions on Power Systems, vol.24, no 2, May 2009.
33. T. Amaree, S. Ranjbar, "Transient Instability Prediction Using Decision Tree Technique," IEEE Transactions on Power Systems, Vol. 8, Issue.3, 2013.
34. Chengxi Liu, Kai Sun, Rather.Z.H., Zhe Chen, Bak. C.L., Thogersen, P, Lund P, "A Systematic Approach for Dynamic Security Assessment and the Corresponding Preventive Control Scheme Based on Decision Trees," IEEE Transactions on Power Systems, Vol. 29, Issue.2, 2013.
35. Miao He, Junshan Zhang, Vijjay Vittal, "A Data Mining framework for Online Dynamic Security Assessment: Decision Trees, Boosting and Complexity Analysis," in Proc. 2012 IEEE Innovative Smart Grid Technologies Conf.pp.1-6.
36. S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, "Decision trees for real-time transient stability prediction," IEEE Trans. Power Syst., vol.9, no. 3, pp. 1417–1426, Aug. 1994.
37. Jeu-Min Lin, Tainan, Taiwan, Shyh-Jier Huang, Kuang-Rong Shih "Application of sliding surface-enhanced fuzzy control for dynamic state estimation of a power system," IEEE Transactions on Power Systems, Vol. 18, Issue.2, 2003.
38. S. K. Tso, J. K. Lin, H. K. Ho, C. M. Mak, K. M. Yung, Y. K. Ho, "Data Mining for Detection of Sensitive Buses and Influential Buses in a Power System Subjected to Disturbances," IEEE Transactions on Power Systems, Vol 19, no.1, 2014
39. Hassan Nouri, Mohsen Mohammadi Alamuti, "Comprehensive Distribution Network Fault Location Using the Distributed Parameter Model," IEEE Transactions on Power Delivery, Vol 26, no.4, 2011
40. http://www.smartgrids.eu/mission
41. Konstantinos Pelechrinis, Marios Iliofotou, Srikanth V. Krishnamurthy, "Denial of Service Attacks in Wireless Networks: The Case of Jammers," IEEE Transactions on Communication Surveys and Tutorials, Vol 13, no.2,2011
42. Yao Liu, Peng King, Micheal.K. Reiter, "False Data Injection Attacks against State Estimation in Electric Power Grids," ACM Journal, 2011.
43. Ting Liu, Yun Gu, Dai Wang, Yuhong Gui, Xiaohong Guan, "A Novel Method to Detect Bad Data Injection Attack in Smart Grid," In Proceedings of IEEE INFOCOM Workshop on Communications and Control for Smart Energy Systems, 2013.

44. Ata Arvani, Vittal S. Rao,“Detection and Protection Against Intrusions on Smart Grid Systems,” In the Proceedings International Journal of Cyber-Security and Digital Forensics (IJCSDF) 3(1): 38-48 The Society of Digital Information and Wireless Communications, 2013.

45. Gerard Vancells, Joaquim Meléndez Sergio, Herraiz Juan Prieto, Guillermo Bravo, “ Bad Data Detection and Identification in Distribution System by means of Principal Component Analysis,” In Proceedings of 21st International Conference on Electricity Distribution, 2011

46. Oliver Kosut, Liyan Jia, Robert J. Thomas, Lang Tong, “Malicious Data Attacks on Smart Grid State Estimation: Attack Strategies and Countermeasures,” ACSP, Cornell University, 2010. http://acsp.ece.cornell.edu/papers/KosutJiaThomasTong10SmartGridCom.pdf

47. Ondrej Linda, Todd Vollmer, Milos Manic, “Neural Network Based Intrusion Detection System for Critical Infrastructures,” In Proceedings of International Joint Conference on Neural Networks, 2009.

48. Robert Wagner, “Address Resolution Protocol Spoofing and Man-in-the-Middle Attacks,” 2001 http://www.sans.org/reading-room/whitepapers/threats/address-resolution-protocol-spoofing-man-in-the-middle-attacks-474

49. Adnan Anwar, Abdun Naser Mahmood, “Cyber Security of Smart Grid Infrastructure,” The State of the Art in Intrusion Prevention and Detection, CRC Press, Taylor & Francis Group, USA, January 2014, pp. 449-472.

50. R. A. Schlueter, I.-P. Hu, M.-W. Chang, J. C. Lo, and A. Costi, “Methods for determining proximity to voltage collapse,” IEEE Transactions on Power Systems, vol. 6, no. 2, pp. 258-292, Feb. 1991.

51. Tingyan Guo, J. V. Milanović, “On-line prediction of transient stability using decision tree method — Sensitivity of accuracy of prediction to different uncertainties,” In Proceedings of PowerTech, 2013 IEEE Grenoble, 2013.

52. Qin Wang, “A novel continuation-based quasi-steady-state analysis approach to mitigate long-term voltage instability” PhD Dissertation 2001.

53. Vijay Vittal, “A Tool for On-line Stability Determination and Control for Coordinated Operations between Regional Entities Using PMUs” PSERC Final Project Report.

54. Noor Izzri Abdul Wahab, Azah Mohamed,“Transient Stability Assessment of Power Systems using Probabilistic Neural Network with Enhanced Feature Selection and Extraction” International Journal on Electrical Engineering and Informatics - Volume 1, Number 2, 2009

55. M. Mohammadi, G.B. Gharehpetian “Application of core vector machines for on-line voltage security assessment using a decisiontree-based feature selection algorithm” Published in IET Generation, Transmission & Distribution

56. M. J. E. Alam, K. M. Muttaqi, D. Sutanto, “A SAX-Based Advanced Computational Tool for Assessment of Clustered Rooftop Solar PV Impacts on LV and MV Networks in Smart Grid” IEEE Transactions on Smart Grid, Vol 4, no1, 2013.
57. H. Mori and Y. Umezawa, “A SAX-based method for extracting features of electricity price in power markets,” Transmission & Distribution Conference & Exposition: Asia and Pacific, 2009.
58. http://bv.com/Projects/Salt-river-project-transformer-asset-management
59. Entergy Report “Superdome Partial Power Outage”- http://entergy-neworleans.com/content/superbowl/130202_Report.pdf
60. Cynthia K. Veitch, Jordan M. Henry, Bryan T. Richardson, Derek H. Hart, “Microgrid Cyber Security Reference Architecture”, SANDIA Report
61. Emanuel E. Bernabeu, Farid Katiraei, “Aurora Vulnerability Issues & Solutions Hardware Mitigation Devices (HMDs)” July 2011
62. Nayot Poolsappasit, Rinku Dewri, Indrajit Ray, “Dynamic Security Risk Management Using Bayesian Attack Graphs”, EEE Transactions on Dependable and Secure Computing, vol.9, issue 1, 2011
63. Teodor Sommestad, Mathias Ekstedt, Lars Nordström “Modeling security of power communication systems using defense graphs and influence diagrams”.
64. Xuan Liu and Zuyi Li, “Local Load Redistribution Attacks in Power Systems With Incomplete Network Information”, IEEE Transactions on Smart Grid, Vol 5, no 4, 2011.
65. Wenyuan Li , “Risk Assessment of Power Systems: Models, Methods, and Applications”, 2nd Edition.
66. Qun Qiu , “Risk Assessment of Power System Catastrophic Failures and Hidden Failure Monitoring & Control System”, PhD Dissertation 2003.
67. Prem Alluri, Sarika Khushalani-Solanki, Jignesh Solanki, Tim Menzies, “Power System State Recognition using Data Mining Algorithms”, Proceedings of North American Power System Symposium, USA, Sept 2013.
68. P.P Rodrigues, J.Gama, J.P. Pedroso, “Hierarchieal Clustering of time series data streams,” IEEE Transactions on Knowledge and Data Engineering, vol 20, issue 5, 2008.
69. http://en.wikipedia.org/wiki/Random_forest
70. Ho, Tin Kam, “Random Decision Forest”, Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.
71. Ho, Tin Kam, "The Random Subspace Method for Constructing Decision Forests". IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998.

72. Noor Izzri Abdul Wahab, Azah Mohamed, “Area-Based COI-Referred Rotor Angle Index for Transient Stability Assessment and Control of Power Systems” Journal in Hindawi Publishing Corporation, 2012.
73. http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
74. 'Hongjun Lu, Weiguo Fan, Cheng Hian Goh, Stuart E. Madnick, David W. Cheung, “Discovering and Reconciling Semantic Conflicts: A Data Mining Perspective” 1997 Report.
75. “Dynamic Time Warping” Springer Chapter 4.
76. Arno Zinke, Dessislava Mayer, “Iterative Multi Scale Dynamic Time Warping” Computer Graphics Technical Reports-2006
77. http://en.wikipedia.org/wiki/Edit_distance
78. Oliver Kosut, Liyan Jia, Robert J. Thomas, Lang Tong, “Malicious Data Attacks on Smart Grid State Estimation: Attack Strategies and Countermeasures”, In Proceedings of First IEEE International Conference on Smart Grid Communications (SmartGridComm), 2010
79. http://www.mathworks.com/products/matlab/
80. Eibe Frank, Mark Hall, Geoffrey Holmes, Richard Kirkby, Bernhard Pfahringer, Ian H. Witten, “WEKA”, Chapter 1.
81. http://www.r-project.org/