NASPI WHITE PAPER

Data Mining Techniques and Tools for Synchrophasor Data

Prepared by NASPI Engineering Analysis Task Team (EATT)

January 2019

Disclaimer

Acknowledgments

Editors:
Evangelos Farantatos (EPRI) – NASPI EATT Lead
Brett Amidan (Pacific Northwest National Lab) – White Paper Lead

Contributing Authors (Alphabetical Order):
Reza Arghandeh (Florida State University)
Daniel Bienstock (Columbia University)
Pavel Etingov (Pacific Northwest National Lab)
Sean Murphy (PingThings)
Femi Omitaomu (Oak Ridge National Lab)
Matthew Rhodes (Salt River Project)
Tom Rizy (Oak Ridge National Lab)
Anurag K Srivastava (Washington State University)
Kai Sun (University Tennessee Knoxville)
Xiangyang Zhou (American Transmission Company)

1 Introduction 6
1.1 Synchrophasor Technology Background Information 6
1.2 Data Mining Background Information 8
1.2.1 Definition and Advantages of Data Mining 8
1.2.2 How Data Mining Has Been Used in Other Domains 8
1.3 An Introduction to Big Data 9
1.3.1 Characteristics of Big Data in the Utility Industry 9
1.3.1.1 Data Volume 9
1.3.1.2 Data Velocity 10
1.3.1.3 Data Variety 10
1.3.2 How Big Data Architecture Is and Could Be Used in the Power Grid 11
2 Data Mining Techniques 11
2.1 Feature Extraction 12
2.1.1 Principal Component Analysis 12
2.1.2 Manifold Learning 14
2.2 Clustering (Unsupervised Learning) 15
2.2.1 K-means 15
2.2.2 Hierarchical Clustering 16
2.2.3 Fuzzy Clustering 16
2.2.4 DBSCAN 17
2.3 Classification (Supervised Learning) 17
2.3.1 Linear and Quadratic Classifiers 17
2.3.2 Kernel Estimation 18
2.3.3 Decision Trees 19
2.3.4 Support Vector Machines 20
2.3.5 Neural Networks 20
3 Software Tools and Big Data Platforms for Data Mining 21
3.1 Data Mining Tools 21
3.1.1 Open Source Languages 21
3.1.1.1 R 21
3.1.1.2 Python 22
3.1.2 Open Source Data Mining Software 23
3.1.2.1 RapidMiner 23

3.1.2.2 Weka 23
3.1.2.3 Orange 23
3.1.3 Commercial Languages 23
3.1.3.1 MATLAB 23
3.1.3.2 SAS 24
3.1.4 Commercial Data Mining Software 24
3.1.4.1 SAS Enterprise Miner 24
3.1.4.2 IBM Intelligent Miner 24
3.1.5 Data Stream Processing Software 24
3.1.5.1 Stream Analytics 24
3.1.5.2 IBM Streaming Analytics 24
3.2 Big Data Platforms 24
3.2.1 Overview 24
3.2.1.1 Key Principles of Big Data Platforms 25
3.2.1.2 First Generation [2003 – 2010] 26
3.2.1.3 Second Generation [2010 – 2017] 26
3.2.1.4 Third Generation [2017 - present] 26
3.2.2 Example Platforms 27
3.2.2.1 PNNL's Cloud Based Analytical Framework for Synchrophasor Data Analysis 27
3.2.2.2 PingThings' PredictiveGridTM – Universal Sensor Analytics Platform 28
4 Application of Data Mining Techniques with Synchrophasor Data – Use Cases 29
4.1 Event and Anomaly Detection 30
4.2 Data Integrity Situational Awareness Tool 30
4.3 A Systematic Approach for Dynamic Security Assessment and the Corresponding Preventive Control Scheme Based on Decision Trees 31
4.4 Synchrophasor-based Data Mining for Power System Fault and Failures Analysis 33
4.5 Using Phasor Data for Visualization and Data Mining in Smart-Grid Applications 33
4.6 Synchrophasor Data Baselining and Mining for Online Monitoring of Dynamic Security Limits 34
4.7 Power System Data Management and Analysis Using Synchrophasor Data 35
4.8 Online Dynamic Security Assessment with Missing PMU Measurements: A Data Mining Approach 36

4.9 Online Calibration of Phasor Measurement Unit Using Density-based Spatial Clustering 37
4.10 SRP/ASU PMU-Based Online Monitoring of Critical Power System Assets 38
4.11 PMU-Based Load Monitoring with anomaly detection 39
5 Conclusions 40
6 References 41

1 Introduction

Data mining is the process of turning raw data into useful information. Data mining has been employed in many different data-rich industries, including banking, healthcare, manufacturing, and telecommunications. With the additions of thousands of PMUs to the nation's power grid, the power systems industry has the data necessary to take advantage of data mining techniques and gain actionable insights.

This white paper discusses the following topics related to applying data mining to the power systems industry:

- provide a high level overview of data mining,
- review how data mining has been used in various industries,
- present common big data architecture and software languages and tools that facilitate data mining, and
- provide use cases that show how data mining has been applied in the power grid community.

1.1 Synchrophasor Technology Background Information

Synchrophasor technology and systems use PMUs to monitor electrical quantities (e.g. voltage, current phasors, and frequency) at specific locations on an electric power system. PMUs estimate phasor values [1] (typically 30 or more measurements per second) using the measured voltage/currents and time-stamp these phasor values (synchrophasor data) using the Global Positioning System (GPS) signal as a reference clock for time alignment (see Figure 1-1). The resulting information gives transmission grid planners and operators a high-resolution view of power system states throughout the grid in real time and provides data for post-analysis of various types of disturbances such as generator trips, transmission line outages, and especially cascading blackouts.

Time synchronization of key field measurements for the purpose of tracking voltage, current flows, local system frequency, and rate of change of frequency in electric power systems began in 1983 [2]. At that time, measurement time synchronization was challenging especially for parts of the power system which are far apart (e.g. 100 miles or more). Time synchronized wide area monitoring was lacking fast data transfer capability, high speed computations, and high resolution sensors. The motivation for such a monitoring system at that time was primarily to improve the performance of protection systems by better understanding power system events and disturbances.

Field demonstrations of small numbers of Phasor Measurement Units (PMUs), especially in the western United States, were conducted in the 1980s and 1990s to determine the usefulness of the technology [3]. In the early 2000s, several utilities in the eastern United States deployed PMUs and provided their data to Tennessee Valley Authority (TVA), where the Super Phasor Data Concentrator was funded, developed and first deployed. For many years, TVA aggregated and

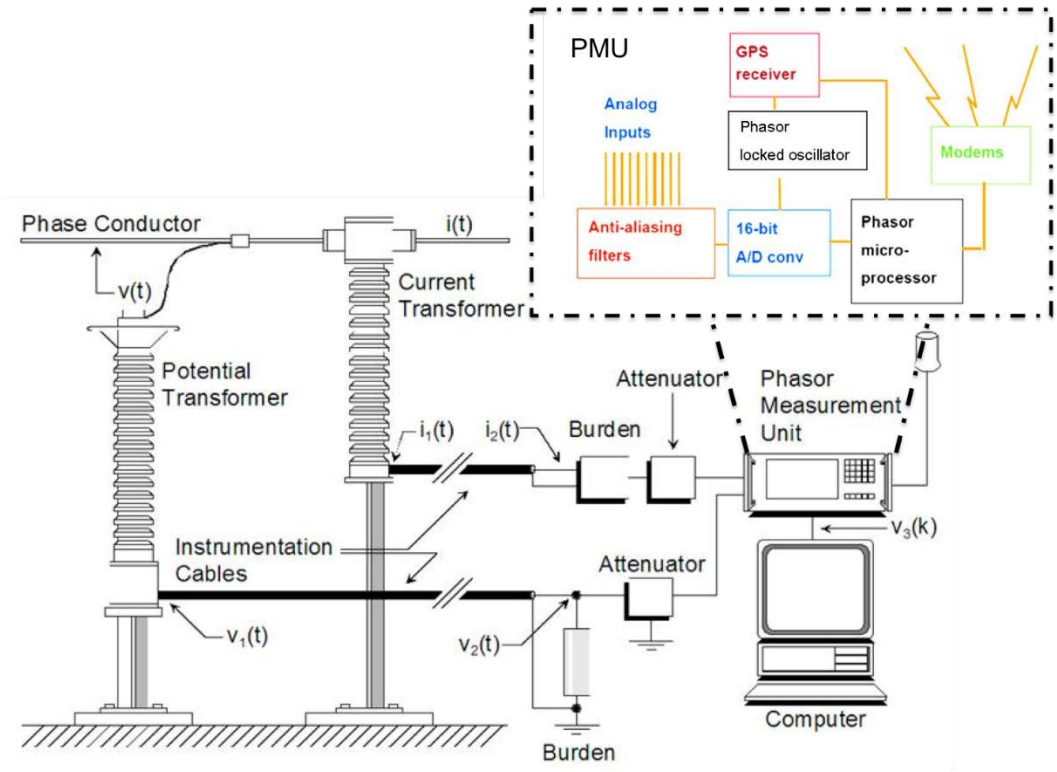

Figure 1-1 - Phasor estimation using PMU connected to CT and PT [4]

time-aligned the PMU data and sent a composite signal back out to all the utilities contributing data. This marked the early stage of shared wide-area situational awareness.

PMU usage increased in 2004 when the U.S.-Canada investigation report of the 2003 blackout was released [5]. The blackout report recognized that many of North America's major blackouts have been caused by inadequate situational awareness for grid operators and recommended the use of synchrophasor technology to provide real-time wide-area grid visibility. The report concluded that the August 2003 Northeast blackout could have been prevented and that immediate actions needed to be taken in both the United States and Canada to ensure the reliability of the North American electric system. One of the key findings was that new technology and investment was needed to provide adequate situational awareness and the ability to react to such a major disturbance. The North American Electric Reliability Corporation's (NERC) Real-Time Tools Best Practices Task Force in 2008 recommended that real-time operational tools should have high speed capabilities, both in accessing and processing power system data, to ensure the reliability of electric power systems.

The American Recovery and Reinvestment Act (ARRA) of 2009, which co-funded 14 projects (for 13 recipients) to deploy PMUs, associated communications and data management systems, and advanced synchrophasor applications greatly increased the number of standalone PMUs and

devices with PMU capabilities (such as relays) on the three major interconnection of North America. This effort greater increased the quantity of PMU data available. These grant recipients included Independent System Operators (ISOs), Regional Transmission Organizations (RTOs), and large and small utilities.

In recent years, synchrophasor technology is being applied for distribution networks for better situational awareness in the grid. The advent of distribution level PMUs prepares the grid operators and planners for new paradigms in distribution including the growth of distributed energy resources, controllable grid edge devices, and electric vehicles [6].

PMUs in transmission and distribution networks bring new opportunities for more active, intelligent, and secure control from precise and time-synchronized measurements that make the grid behavior comparable between different locations.

The high resolution of synchrophasor data (typically 30 or 60 samples per second) and the availability of huge volumes of data, enable application of data mining and machine learning techniques in both operations and planning environment. Real time applications such as situational awareness, event detection, real time load parameter tracking, stability monitoring etc., can leverage streaming synchrophasor data as well as historical data and apply data mining techniques to support grid operators in assessing the operating condition of the grid and provide guidance to operators for potential mitigation actions in case of system security threats. In the planning environment massive amount of recorded and stored synchrophasor data can be used for offline applications such as system baselining.

1.2 Data Mining Background Information

This subsection defines data mining in general and what it can accomplish. It also lists a few real-world examples of how data mining has been used in other domains.

1.2.1 Definition and Advantages of Data Mining

Simply put, data mining is the examination of raw data, usually large amounts of data, in order to gain new insight or information. Often these insights or patterns in the data are hidden because of the complexity and size of the data. Statistical and mathematical algorithms are employed to help pull useful and actionable information from the original data. This can be done more effectively with a particular hypothesis in mind that is being investigated, but value can still often be gained without prior guidance into what is of interest.

1.2.2 How Data Mining Has Been Used in Other Domains

Service providers and retail businesses are areas in which data mining has been successfully deployed and valuable insight gained. These businesses create prediction models that help them understand when customers may be considering leaving or be interested in buying based on past data. They then can follow up with specific marketing offers to help entice them into staying or purchasing. Online businesses often use data mining to help recommend certain products to

interested purchasers, without the need of a purchaser expressing previous interest in those products.

Surveillance related businesses also use data mining. Credit card and other financial businesses use it to detect when possible fraud is occurring. Intrusion detection fields, including cyber-security applications, help analysts distinguish between common everyday activity and potentially harmful intrusions. Crime prevention agencies have detected trends in crime and developed models to know where and when to deploy policing manpower.

1.3 An Introduction to Big Data

While the term "big data" has been heavily abused by marketing and sales, its meaning was captured by one of the first documented uses of the term in the 1997 paper, "Application-controlled demand paging for out-of-core visualization" from NASA. The authors were interested in scientific data visualization, which "provides an interesting challenge for computer systems: data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk. We call this the problem of big data. When data sets do not fit in main memory (in core), or when they do not fit even on local disk, the most common solution is to acquire more resources." Simply put, big data are data that exceeds the capabilities of a single computer. Integral to this definition is that it is relative; over time the amount of data considered "big" changes as computing hardware evolves – today's big data won't be considered big in five years.

However, this definition is not quite sufficient to understand the impact that big data has had on numerous industries. If an organization occasionally encounters or requires the use of data sets that don't fit on a single computer, it will find an ad-hoc solution that doesn't significantly alter existing workflows to achieve short-term objectives. The real transformation occurs when the successful operation of the organization depends upon the continued and consistent use of data sets at this scale. This requires the adoption of software and technology substantially different than the existing solutions that have supported the industry to that point.

1.3.1 Characteristics of Big Data in the Utility Industry

Big data often has three defining characteristics, also known as the three "V's" of big data; these include (1) volume, (2) velocity, and (3) variety.

1.3.1.1 Data Volume

The number of sensors deployed on the grid has exponentially increased through specific utility programs, standardization, and industry market forces. Competition among vendors has driven product differentiation through the addition of sensing capabilities (i.e. multi-function relays that can also capture time synchronized phasor measurements) that do not significantly increase the overall cost. This can be seen across asset types and vendors. The heat map in Figure 1-2 below shows the annual volume of data generated by various types of utility sensors demonstrating that even with sensors deployed today, data volumes are entering the petabyte and even exabyte regime.

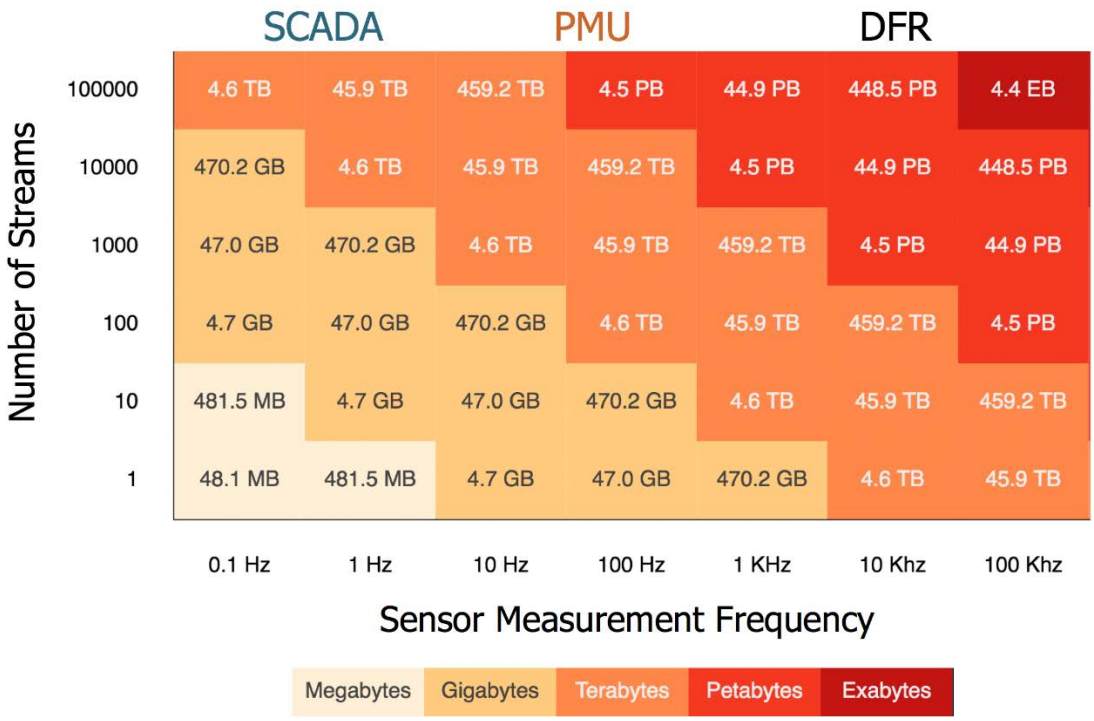

Figure 1-2 - Annual data volume generated as a function of sensor type and number of data streams.

1.3.1.2 Data Velocity

The physical processes defining the behavior of the grid are continuous in nature. Higher sampling rates allows sensors to capture faster grid dynamics, yielding previously unavailable information about the physical processes in question. This is in stark contrast to the industries in which data science arose in organizations where measurement data captured discrete events—a tweet or an email that arrive at a particular point in time—and additional useful information cannot be captured between arrivals.

For utilities, there is a real need to capture higher frequency data from physical processes such as the voltage and current waveforms on the grid. PMUs, whose technology is now several decades old, offer continuous monitoring of both the transmission and distribution grid at 30Hz up to 240Hz. Further, point-on-wave or continuous waveform monitoring was a much-discussed topic at the 2018 IEEE PES General Meeting in Portland, OR and sensors are capturing continuous, streaming measurements at 32 to 1024 samples per cycle (1920 to 61,440Hz). The exciting aspect of continuous waveform monitoring is that virtually every other measured quantity can be derived from point-on-wave data; for example, a PMU can be thought of as an edge computer that transforms point-on-wave into time synchronized phasor measurements.

1.3.1.3 Data Variety

Sensors have been deployed across the grid with pragmatism and financial restraint and were not intended to capture data for what-if scenarios or analyses that might be useful someday. Instead,

utility sensors have been designed, built, and deployed to address known, impactful issues. To exacerbate the situation, the hardware companies who design and build sensors often developed the associated software, intentionally written only to handle data from a particular sensor type and, perhaps, a single vendor. Thus, data from utility sensors, despite measuring different characteristics of the same grid, often reside within a fragmented, siloed environment that cannot provide a cohesive or integrated view of the system being measured.

This is in strong contrast to the “tech” companies that started and advanced data science such as Google, Facebook, and Amazon. During the operation of these digital-by-default businesses, the fundamental act of delivering online products and services created data, without explicit sensors, regardless of intention to address a known problem. This “data exhaust” was often captured because the core competencies of these contemporary tech giants were handling data.

1.3.2 How Big Data Architecture Is and Could Be Used in the Power Grid

Utilities already generate "big data" in their day to day operations and the volume, velocity, and variety of this data relevant to the reliable, resilient, and optimal operation of the electric grid will only increase. All too often the data from the various types of sensors describing different aspects of the grid's behavior is siloed, stored in incompatible systems and thus cannot be combined easily. Bringing together all sensor data into one cohesive picture of the grid at varying resolutions and time scales and also bringing in external data will fundamentally change how the grid is operated. But bringing this data into one place is only the start. The data must be easily accessed, visualized, used, analyzed, and consumed by downstream applications that may get tested out and thrown away in the span of a weekend or used for the next decade.

The philosophy underlying the big data revolution is that data are valuable and more of it is even more valuable. Unfortunately, we have also seen many examples within utilities where data are down-sampled or compressed in a lossy fashion, losing information describing the grid's behavior forever. Thus, intentionally throwing away data in an age where a terabyte of storage can be less than $25 is a questionable decision at best. Retaining all the data increases the probability that the historical archives will contain examples of rare events or system states. Further, what appears to be noise in higher frequency data may be the information that helps a machine learning algorithm predict the impending failure of an asset. Finally, as big data are retained and become more accessible and usable, new ideas of previously unimagined applications will be built, a statement virtually guaranteed by the numerous examples from across industries.

2 Data Mining Techniques

This section reviews several common data mining techniques and discusses why those techniques are used.

2.1 Feature Extraction

Feature extraction can be thought of as a preprocessing tool. It is commonly used when data sets are very large and redundant information is expected within the data. The features are generally variables derived from the raw data. They may be simple summaries, like means or standard deviations, or they may be more complex, like linear or nonlinear combinations of variables. These features are usually much smaller than the raw data and used as inputs for different analytical algorithms. Multiple feature extraction methods are discussed below.

2.1.1 Principal Component Analysis

The main purpose of principal component analysis (PCA) is dimensionality reduction. Assuming the data are in a matrix format, with the rows containing the observations and the columns the features, the common steps to perform PCA are enumerated below.

1. Normalize each column (zero mean unit variance). This is not a required step, but it is recommended, especially if the features are in different orders of magnitude (i.e. one feature has values around 10, while another has values around 1000). Without normalizing the columns, the features with higher magnitude values will receive more weight.
2. Compute the covariance matrix of the data matrix (normalized data matrix, if normalization was done).
3. Find the eigenvalues and eigenvectors of the covariance matrix. The sum of all eigenvalues is the total variance of the sample data. Choose the first *n* largest eigenvalues that account for a predetermined percentage of total variance (e.g. 80%, 90%, 95%). The corresponding eigenvectors make up the linear combination weights that when applied to the original data, creates the principal components (the new reduced dataset).

Figure 2-1 is a simple example using a 17-dimensional data set consisting of the average consumption of 17 types of food (gm) per person per week for the four countries in UK [7].

	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175

Figure 2-1. A simple example using 17-dimensional data set for the four countries in UK.

Applying PCA to the original data in the table above reduces the dimensionality from the 17 original variables to just two. These two new variables, referred to as pc1 and pc2 in this example, are linear combinations of the original variables and do not represent the same physical properties of the original data. The observations in {pc1, pc2} can be visualized in Figure 2-2.

Figure 2-2. Visualization of Two Observations {pc1, pc2}.

As can be seen in this plot, the two variables capture the essence of the 17 variables and show that N Ireland is most different from the others, especially in the first dimension (pc1). Looking at the values in the first eigenvector will show which of the original variables contributed most to pc1.

2.1.2 Manifold Learning

PCA is a good technique to transform high-dimensional data to low-dimensional data using a linear approach. The principal components are projections of the original data to eigenvectors, which are lines in 2-D and planes in 3-D. PCA might be not accurate if linearity is not present in the original data. For example, if PCA is applied to data on a Swiss roll, the structure of the roll will be lost from the reduced data, as show in Figure 2-3 [8].

Figure 2-3. Data Structure Loss when Applying PCA to Data on a Swiss Roll.

PCA finds a plane, in this case the first two principal components, to capture the most variance in the original data and project it onto the plane. The structure of the roll is lost in the projection. The red points are furthest to the blue points on the roll, but that is not the case in the reduced projection. Nonlinear manifold learning will make a better choice to unroll the roll and reveal the original structure in the reduced data.

Manifold is a term used in geometry topology studies. It generally refers to a nonlinear (geometry) structure that exhibits local linear structure (Euclidean distance can be defined locally). Earth (globe) is a manifold, although it's a nonlinear shape, but at a local scale it consists of countless small planes. Swiss roll is another example, locally it is just a plane.

Points on a manifold can be charted onto the local linear structure using a few available techniques. This is how nonlinear high dimensional data are reduced to low dimensional data in manifold learning. Available manifold methods include IsoMap, LLE, t-SNE etc. with different pros and cons. In Manifold learning, it is the local linear structure that the method is trying to preserve, as opposed to variance as in PCA. The available techniques differ in the choice of local structure to preserve and will give different results. This is an active research field and there's no consensus within researchers about which algorithm makes a better choice than others (it'll probably depend on your data).

Figure 2-4 is obtained by applying IsoMap on a sequence of pictures of hands with varying degrees of wrist rotation and finger extension. The technique is very effective in identifying the two variables in the original data. It is very clear that all pictures can be described by two variables, which, by prior knowledge about the pictures, should be finger extension and wrist rotation.

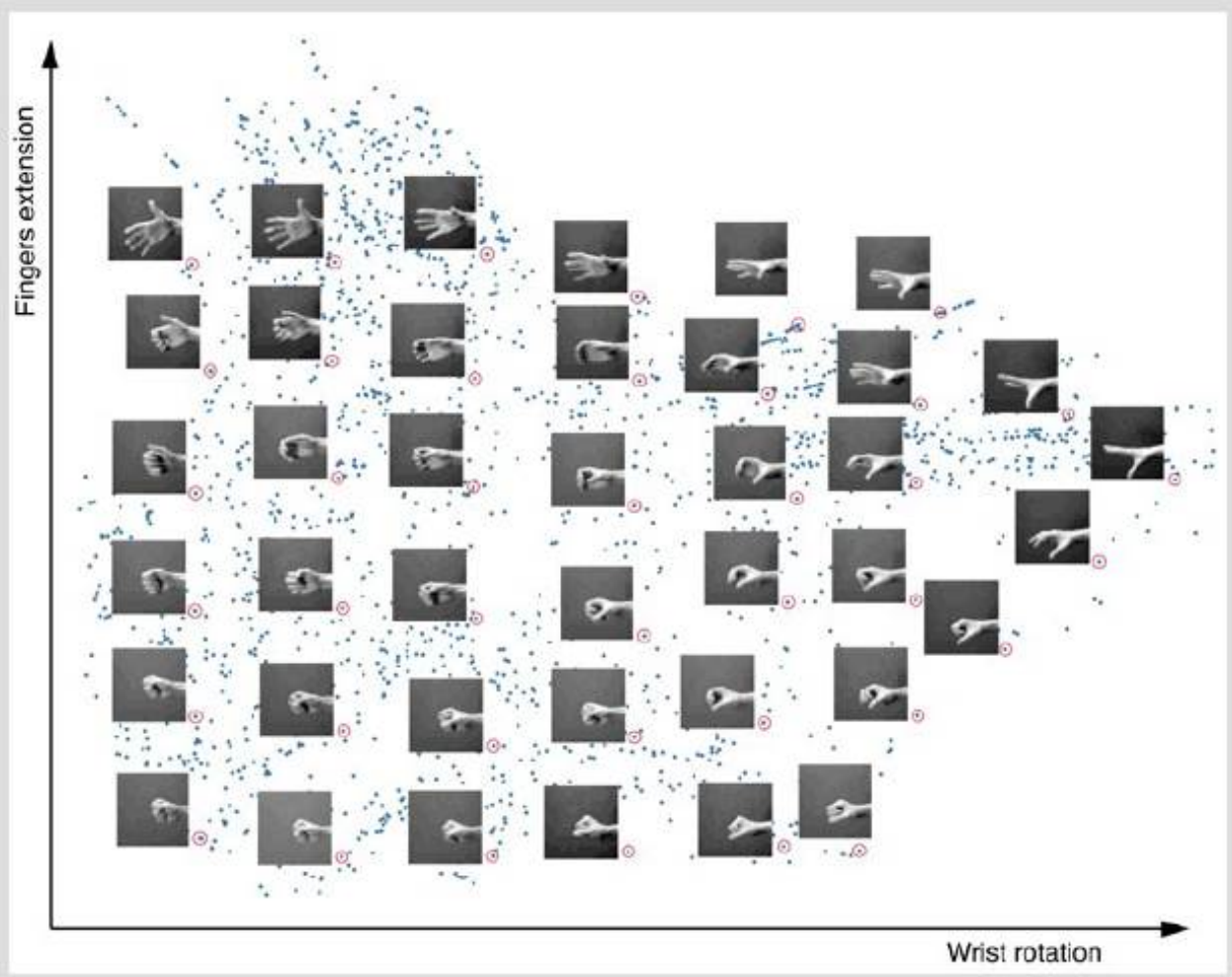

Figure 2-4. Applying IsoMap on a Sequence of Pictures of Hands with Varying Degrees of Wrist Rotation and Finger Extension.

2.2 Clustering (Unsupervised Learning)

Statistical clustering is the process of grouping similar observations or objects into groups based on measured variables or characteristics. This is a type of unsupervised learning because the groups were unknown beforehand, meaning that there is no guidance or ground truth as to what constitutes each group. This white paper describes parameter dependent clustering algorithms, which rely upon user defined inputs like number of groups or some other cutoff criteria, and parameter independent clustering algorithms.

2.2.1 K-means

K-means clustering is a popular parameter dependent clustering algorithm. K-means groups observations by determining which group center (mean) the observation is closest to. This algorithm is dependent on observations that start the algorithm, because the group means are a function of the previous observations that were already clustered into the groups. Usually k-

means is run with many different random starts to help the methodology to optimize. The user has to define either the number of groups that are expected to exist, or the actual means of the groups.

2.2.2 Hierarchical Clustering

Another common clustering algorithm is hierarchical clustering. This method builds a hierarchy of clusters by either starting from the bottom up—observations begin by themselves and then pair with the most similar points—or top down—observations begin in one group and then split out by being most different. Hierarchical results are often displayed in a dendrogram, shown in Figure 2-5. From the dendrogram, the user can decide where to cut the tree, resulting in some number of groups. In this plot, this dendrogram is cut at around 16 (on the y-axis), resulting in 6 groups, defined by the six colors. For example, the green group contains observations 34, 45, 2, & 10. This method requires the user to decide where to make the cut, to determine how many groups exist.

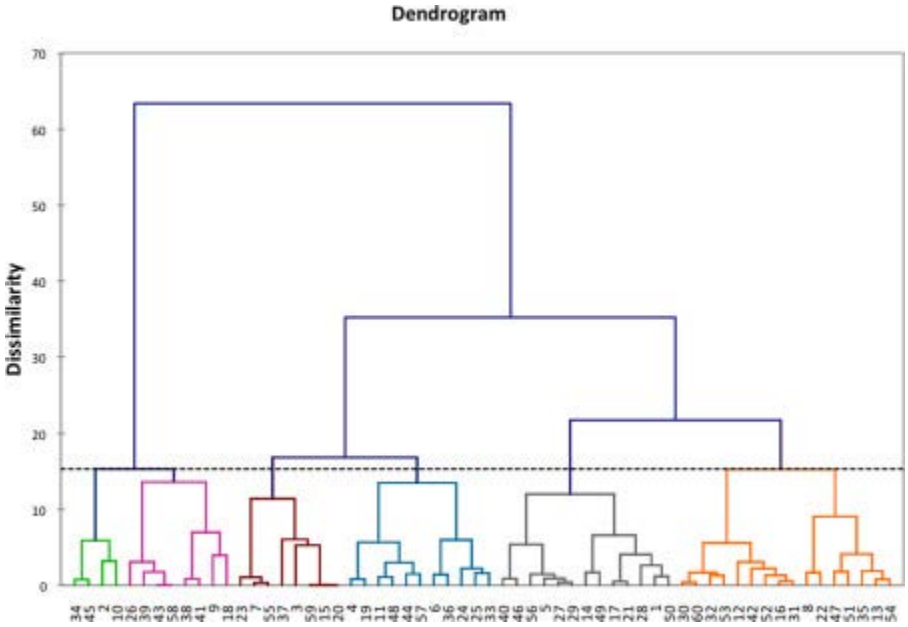

Figure 2-5. Dendrogram Display for Hierarchical Clustering.

2.2.3 Fuzzy Clustering

Hierarchical and k-means clustering are examples of hard clustering, where each observation is assigned to a group. Fuzzy clustering is soft clustering, where each observation could belong to more than one group. Fuzzy c-means clustering algorithm is a widely used fuzzy method. It is similar to k-means, except that there is a term called the fuzzifier which is calculated. This term helps in the determination of an observation belonging to multiple groups.

2.2.4 DBSCAN

The most common parameter independent clustering algorithm is DBSCAN. DBSCAN is a density-based clustering method that groups together observations that are packed together. This method marks observations as outliers that exist in low-density areas. This method defines core points as those where at least MinPts (user defined value of points) are within distance *Eps* (user defined value). Figure 2-6 shows a simple two-dimensional example of how DBSCAN works [9]. This reference also provides more information about the algorithm.

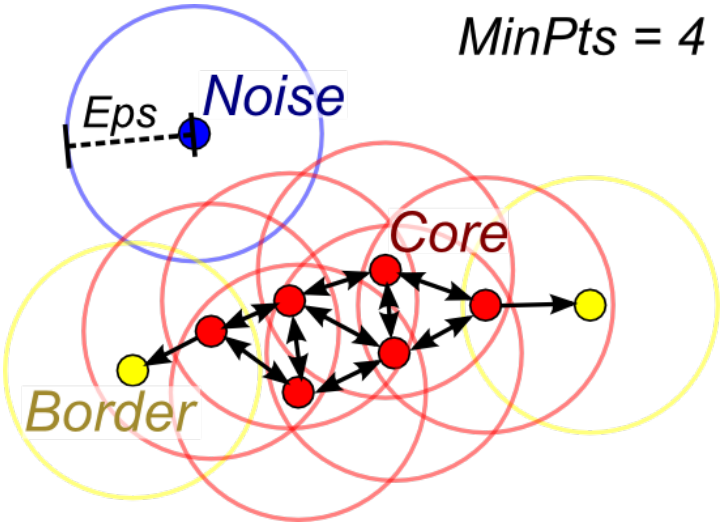

Figure 2-6. DBSCAN Applied to a Simple Two Dimensional Example [9].

2.3 Classification (Supervised Learning)

Statistical classification is the process of determining the characteristics that define a sub-population or group and then using that information to identify future observations that possibly belong to that given sub-population or group. Classification is called supervised learning because there are a set of observations with known group labels, which are used to determine the rules that define each group.

There are many different classification algorithms. These algorithms include: linear and quadratic classifiers, kernel estimation, decision trees, support vector machines, and neural networks. Each of these algorithms is discussed in further detail below.

2.3.1 Linear and Quadratic Classifiers

Linear and quadratic classifiers rely upon linear (or quadratic) combinations of the various characteristics to create the rules that define each group. LDA (linear discriminant analysis) and QDA (quadratic discriminant analysis) are closely related to regression analysis and are common classifiers. Figure 2-7 shows graphically how LDA rules use linear cuts to define different groups. QDA is similar, except that the lines can be quadratic curvatures.

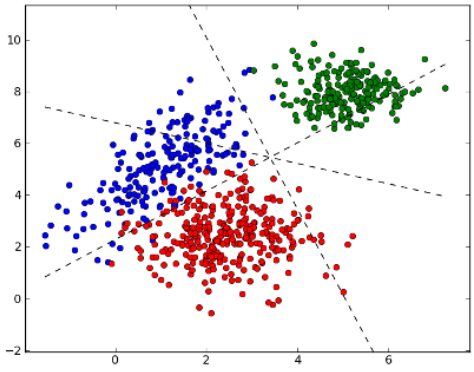

Figure 2-7. Using LDA Linear Cuts to Define Different Groups.

Logistic regression is a common classification approach that is used when the outcome is binary (part of the group, or not). Multinomial logistic regression can be used when the response variable has more than two outcomes and ordinal logistic regression can be used when the outcomes are ordered categories. Logistic regression is analogous to linear regression, except that it is based on a Bernoulli distribution, instead of a Gaussian (normal) distribution. Logistic regression can provide probabilities calculated for each outcome.

2.3.2 Kernel Estimation

k-nearest neighbors is a common kernel estimation approach to classification. It is a non-parametric method in which membership for an observation is determined by a majority vote of its neighbors. “k” is the number of closest neighbors to consider for membership. Figure 2-8 gives a good example of k-nearest neighbors [10]. The point of interest (the star) would be classified into Class B, if k = 3, but Class A if k = 6. The researcher will have to determine the optimal k value.

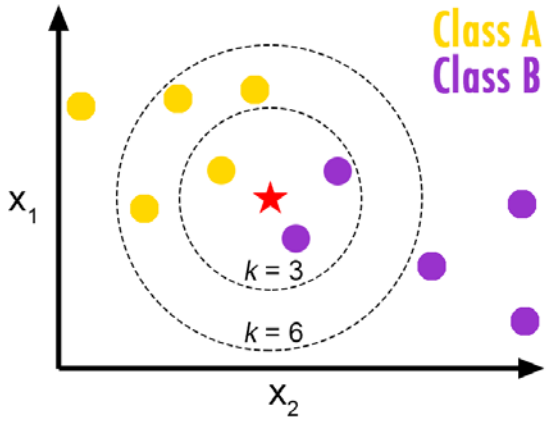

Figure 2-8. An Example of K-nearest Neighbors [10].

2.3.3 Decision Trees

Decision trees use tree-like diagrams to model decisions and consequences for each observation. Decision rules are created from training data, where the true label is known for each observation, and then future observations are placed into a group based on the rules. The characteristics (measurements) from the observations can be numeric or categorical. The algorithm will determine the best place to make splits based on those measurements. Figure 2-9 is an example of decision trees [11]. The goal is to determine a person's credit risk from three variables: income, credit history (good, bad, unknown), and debt (high, low). From this tree an individual can follow the tree diagram, using the rules on the tree, to determine which credit risk the individual belongs.

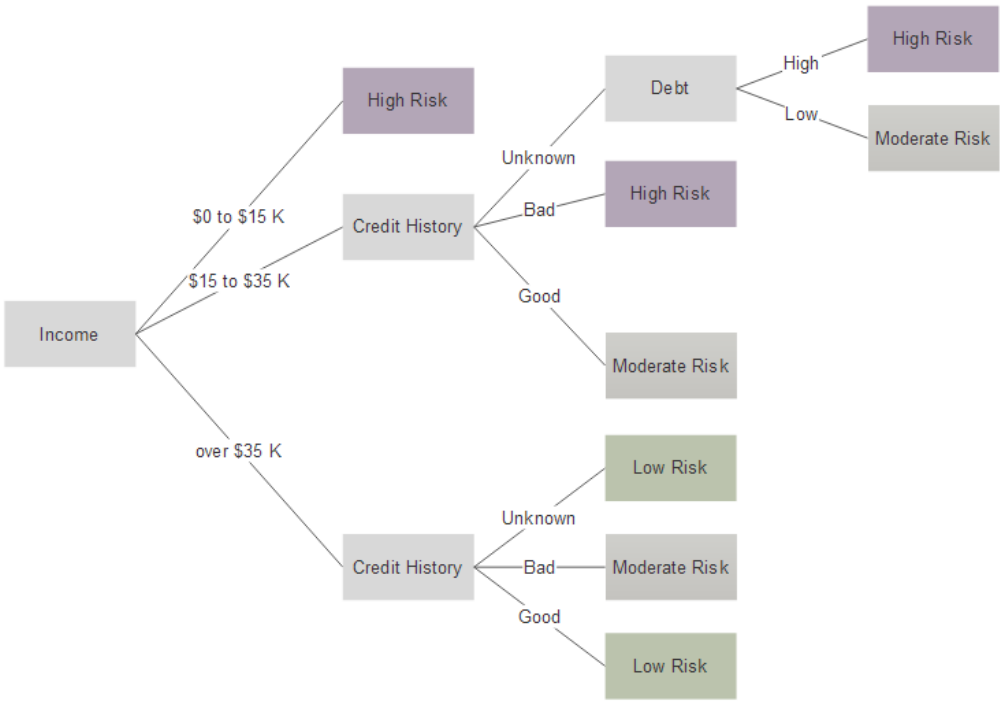

Figure 2-9. An Example of Decision Trees [11].

The two most common decision tree algorithms are CART (classification and regression trees) and random forest. CART can be used to classify an observation into a group or calculate a predicted numeric value (hence "regression" in the name). Random forest is a specific type of decision tree that is based on an ensemble approach using bootstrapping and aggregating (bagging). Random forest is usually better at not overfitting as much to the training data. Because it is an ensemble of trees, it cannot be displayed in tree form, like in the graphic above.

2.3.4 Support Vector Machines

The support vector machines (SVM) technique is based on statistical learning theory and it is used for learning classification and regression rules from data. When used for classification problems, the algorithm is usually called support vector classification (SVC) and when used for regression problems, the algorithm is support vector regression (SVR). Unlike other predictive model, the SVM technique attempts to minimize the upper bound on the generalization error based on the principle of structural risk minimization (SRM) rather than minimizing the training error. This approach has been found to be superior to the empirical risk minimization (ERM) principle employed in artificial neural network [12]. In addition, the SRM principle incorporates capacity control that prevents overfitting of the input data [13]. The SVM technique has been widely used in various real-world applications [14].

The SVM technique continues to gain popularity for prediction because of its several outstanding properties [15]. Some of these properties include:

- the use of a kernel function that makes the technique applicable to both linear and nonlinear approximations,
- good generalization performance as a result of the use of only the so-called support vectors for prediction,
- the absence of local minima because of the convexity property of the objective function and its constraints,
- and the fact that it is based on structural risk minimization that seeks to minimize the upper bound of the generalization error rather than the training error.

The kernel technique is one of main building blocks of SVM. Choosing a suitable kernel function in SVM is equivalent to choosing an architecture for a neural network application. The use of kernels can overcome the curse of dimensionality in both computation and generalization. The fact that simply mapping data into another space can greatly simplify the classification or regression task has been known for a long time [16].

In application to classification problems, SVM can produce models with different kinds of decision borders - it depends on the parameters used (especially on the kernel type). The border can be linear or highly nonlinear. Note that the complexity of the borders does not mean poor generalization, because margin optimization takes care of the proper placement of the border. SVM minimize the empirical risk function with soft margin loss function for classification problems. The construction of the optimal hyperplane is the fundamental idea of SVM. The optimal hyperplane separates different classes with maximal margin (the distance between the hyperplane and the closest training data point). The construction of an optimal hyperplane is impossible if the data set transformed by kernel is not linearly separable. To solve this problem, the soft margin hyperplane technique using slack variables was introduced [12]. [17], [18], [19] describe other extensions of SVM.

2.3.5 Neural Networks

Artificial neural networks (ANN) represents a very broad class of different algorithms designed for classification, regression, signal processing, time series prediction, clustering, etc. Neural networks are built from neurons which are grouped in layers. Neurons may be connected in a

number of ways. The standard artificial neuron is a processing element whose output is calculated by multiplying its inputs by a weight vector, summing the results, and applying an activation function to the sum. The activation function could be one of many types. An output of a linear activation function is simply equal to its input. However, it is not simple for a nonlinear activation function. There are several types of nonlinear activation functions. Differentiable nonlinear activation functions can be used in networks trained with backpropagation. Some of the most common activation functions are the logistic function (output range between 0 and 1) and the hyperbolic tangent function (output range between -1 and 1). Non-differentiable nonlinear activation functions are usually used for perceptrons and competitive networks. The two common types are the threshold function (output is either a 0 or 1) and the signum function (output is either a -1 or 1).

Neurons are grouped into layers, and layers are grouped into networks to form highly interconnected processing structures. An input layer does no processing, it simply sends the inputs, modified by a weight, to each of the neurons in the next layer. This next layer can be hidden layer or the output layer in a single layer design. A network with no hidden layer can separate linearly separable inputs but it will fail if the inputs are not linearly separable. Linearly non-separable patterns can be separated with multilayer networks.

Neural networks with one or more hidden layers are called multilayer neural networks or multilayer perceptrons. Normally, each hidden layer of a network uses the same type of activation function. The output activation function is either sigmoidal or linear. In order to be a universal approximation, the hidden layer of a multilayer network is usually a sigmoidal neuron. A linear hidden layer is rarely used because any two linear transformations can be represented as one linear transformation. Some good references on ANN include [20], [21], [22].

3 Software Tools and Big Data Platforms for Data Mining

This section discusses many of the software tools and computer languages that employ data mining algorithms. It also reviews some of the most common architectures used to handle big data.

3.1 Data Mining Tools

This section reviews some of the more common data mining tools. This discussion separates the tools into coding languages versus OTS (off the shelf) software and open source versus commercial.

3.1.1 Open Source Languages

3.1.1.1 R

R is an open source language for statistical computing. The R language is commonly used by statisticians and data miners for developing analytical and graphical processes and is often the language of choice for the implementation of the absolutely newest statistical techniques and algorithms. Information pertaining to R can be found on the CRAN (Comprehensive R Archive

Network) website [23]. Thousands of user developed packages can be accessed on CRAN. These packages contain functions that perform many different algorithmic, mathematical, and graphical tasks.

There are many data mining related packages accessible in R. A partial list of these packages includes the following:

- nnet (neural networks)
- rpart (decision trees)
- randomForest (random forest)
- gbm (boosting and gradient descent),
- e1071 (support vector machines),
- MASS (qda, lda, mca).

Further information concerning machine learning packages for R language can be found in [24].

R is a very powerful statistical software package, but it comes with a steep learning curve. There are no point and click or automated processes. Coding in the R language is a requirement. R interfaces well with C++, Python, and other programming languages.

3.1.1.2 Python

Python has become the de facto programming language for data analytics and machine learning, both for research purposes and also for operational deployment in large scale production systems. It is a scripting language that can be used interactively and doesn't require compilation of the source code into an executable to run, making it easy to port a Python program between computers and Operating Systems. Python supports procedural and object oriented programming as well as offers some support for functional programming methodologies but not as robustly as purely functional languages such as Lisp and Haskell. Python can call external C or C++ programs, and can be embedded in other languages to implement the scripting capability.

One of Python's greatest assets is that it has an incredibly large ecosystem of open source libraries that extend the language's capabilities, much like MATLAB's toolboxes. NumPy offers a high-performance numerical computing library with highly optimized data structures for vectors and matrices and common mathematical functions; this library's performance serves as the foundation for many other Python libraries. SciPy offers a wealth of scientific and engineering functions for many different disciplines and Matplotlib is a visualization library that was designed to mimic and replicate the functionality offered by MATLAB. However, since its creation, many newer data visualization libraries have been developed that offer more modern approaches to this problem including Seaborn, ggplot, Altair, plotly, and Bokeh. Pandas is a commonly used data analysis library for Python, providing a robust implementation of a data frame and an incredible amount of related capability.

Scikit-learn is the Machine Learning library for Python (although there are others). It has become the "reference" library for machine learning algorithms meaning that once a published approach or algorithm has enough citations, it will almost always get implemented and added to scikit-learn. It is a simple and efficient tool for data mining and data analysis, accessible to everybody,

and reusable in various contexts. The module is built on top of NumPy/SciPy and matplotlib. Also, scikit-learn is an open source, and commercially usable under the BSD license [25].

Machine Learning Python (MLPY) is another Python module for Machine Learning built on top of NumPy/SciPy and the GNU Scientific Libraries. It provides a wide range of state-of-the-art machine learning methods for supervised and unsupervised problems and it is aimed at finding a reasonable compromise among modularity, maintainability, reproducibility, usability and efficiency. MLPY is multiplatform, it works with Python 2 and 3 and it is Open Source, distributed under the GNU General Public License version 3 [26].

With respect to data mining related packages, Google released its neural network library Tensorflow under open source license in 2015. Although the library is also available to languages like C and Java, the full functionality is only made available to Python.

3.1.2 Open Source Data Mining Software

3.1.2.1 RapidMiner

RapidMiner is a point and click, GUI interfaced data mining tool. It is free when analyzing less than 10,000 rows of data (less than 3 minutes of 60 Hz PMU data). The unlimited version is $10,000 a year. R and Python can be incorporated into RapidMiner processes. RapidMiner interfaces with Hadoop with their Radoop software. It has good and easy to use graphical capabilities. More information about RapidMiner can be found in [27].

3.1.2.2 Weka

Weka is a machine learning based software tool with a workbench of machine learning techniques and a GUI interface. It also has the capability for developers to create their own machine learning algorithms. Weka was founded out of New Zealand and is not a commonly used system. It requires data files stored in the uncommon ARFF format, although it will read in CSV files with some issues. More information on Weka can be found [28].

3.1.2.3 Orange

Orange is a widget based open source software package with a GUI for data mining and machine learning methods. It allows for performing of simple data manipulation and visualizations. Desired processes are added in a workflow. Interacts with Python. More information on Orange can be found in [29].

3.1.3 Commercial Languages

3.1.3.1 MATLAB

MATLAB (MATrix LABoratory) is a proprietary scripting language developed by MathWorks. It has many built in functions that allow for matrix manipulations, plotting, and analyses. Optional toolboxes can be purchased. Tall arrays allow the user to apply statistical and machine learning algorithms to data that cannot fit into memory. Further information about MATLAB and how it deals with big data can be found at https://www.mathworks.com/solutions/big-data-matlab.html.

3.1.3.2 SAS

SAS is a software suite that will manage, retrieve, mine, and analyze data. It uses a scripting language or a graphical point and click user interface, depending on user preference. Optional SAS components can be purchased to perform a variety of analyses of data. SAS/INSIGHT contains data mining tools and SAS/STAT contains statistical algorithms. Enterprise Miner is an additional component that is further discussed in the next subsection. Further information can be found at https://www.sas.com/en_us/home.html.

3.1.4 Commercial Data Mining Software

3.1.4.1 SAS Enterprise Miner

SAS Enterprise Miner is a business enterprise solution from one of the leading statistical software companies, SAS. It has an easy to use GUI and can run batch jobs. It professes to have sophisticated data preparation, summarization, and exploration, as well as advanced predictive and descriptive modeling. It also provides open source integration with R. More information can be found in [30].

3.1.4.2 IBM Intelligent Miner

IBM DB2 Intelligent Miner for Data provides many of the data mining functions discussed in this paper. It allows for building and applying mining models from databases or flat files. More information can be found in [31].

3.1.5 Data Stream Processing Software

3.1.5.1 Stream Analytics

Stream Analytics is an event processing engine that specifically works on streaming data. This Microsoft produced software extracts information from streaming data, by identifying patterns, trends, and relationships. More information can be found in [32].

3.1.5.2 IBM Streaming Analytics

IBM Streaming Analytics provides fast streaming analytics that allows developers the ability to use existing Python code. More information can be found in [33].

There are many other commercial and open source data streaming software solutions. For a listing and high level details, see [34].

3.2 Big Data Platforms

3.2.1 Overview

Two technology pathways exist to handle data sets that consistently exceed the capabilities of a single computer. The traditional solution was to build a bigger computer, with more memory, storage, and processors than a “standard” computer. These “enterprise servers” and even super computers are incredibly expensive. There is a large price markup for server-class components

(ECC memory, server-class processors, enterprise hard drives, etc.) due to the willingness of large enterprises to accept these premiums and the fact that economies of scale are drastically reduced. As the system become larger, an increasing percentage of the server hardware becomes custom and the software tends to become increasingly proprietary, both causing a non-linear increase in price.

The alternative to this approach is a solution that relies primarily on software to leverage commodity, off-the-shelf (COTS) computers at enormous scale. While COTS hardware may not be as durable as "enterprise-grade" computing equipment, failure concerns are irrelevant as the software is designed to handle such events gracefully. When working with hundreds or thousands of computers, the question is not if hardware will fail but which hardware is failing at the present moment. This approach to big data, using software to distribute the "big data" problem over lots of COTS computers has been shown to be the most cost-effective approach and forms the ideological basis for big data platforms today.

3.2.1.1 Key Principles of Big Data Platforms

Big data platforms are complex software systems designed to operate across multiple computers and are often composed of numerous layers of functionality. Thus, the most well-known to date, such as Apache Hadoop and Apache Spark are open source. Using open source software is a necessity as it lowers the total cost of platform development by leveraging the efforts of literally thousands of software engineers at almost no cost. Further, these teams of programmers will continue to improve and evolve each software component over time.

A fundamental design principle with big data systems is that you move the computation to the data instead of moving the data to the code. The simple reason for this is that the network bandwidth connecting the computers in the platform is a fixed quantity and also orders of magnitude smaller than the bandwidth between process and main memory within each node. Therefore, it is much more efficient to move the source code that is orders of magnitude smaller in size than the big data. This has many implications for how the platform is architected including that the analytics and machine learning must be core components or "first-class citizens" of the platform. This is one of the fundamental reasons why it is impossible to add or "bolt on" real time analytics, let alone machine learning capabilities, to legacy systems with non-scalable architectures.

This concept has also been extended somewhat to fog and edge computing, where the idea is that computations are done locally at the sensor due to communication channel bandwidth constraints. As with all things, this architecture has tradeoffs. Pushing calculations to the edge is often done to reduce the volume of data streamed to a central location and decrease the response time for control applications. However, this also tends to result in a complete lack of transparency as the calculations done on the edge are vendor specific black boxes.

The idea of "scaling" is critical to Big Data systems for many reasons. As such systems were designed to run in a distributed fashion, the ability to horizontally scale across computers while offering as close to a linear speed increase as possible is key. Further, many big data workloads

are “bursty,” requiring dramatically more compute resources than the baseline for short periods of time for full system analyses or for machine learning or deep learning algorithm training. This is also why big data systems are so often hosted within cloud infrastructures, either internal or external to the organization. Cloud-based systems offer the ability to scale out temporarily for large compute tasks and then immediately scale back to baseline, conserving resources.

3.2.1.2 First Generation [2003 – 2010]

Hadoop is the basis for first generation big data systems but is itself an open source implementation of Google’s distributed file system [35] and the MapReduce data processing model [36], first publicly described in 2003 and 2004 respectively. Hadoop’s distributed file system allowed hard drives across a large number of machines to appear as a single file system to the developer and map reduce provided a simplified programming model that allowed this distributed data to be processed and analyzed, often for traditional business intelligence purposes. This approach took advantage of the relatively inexpensive hard drives of the day and the vast bandwidth available when reading and writing data in parallel across a number of machines and hard drives. Further, Google understood that the limiting reagent was often the software engineer and this approach attempted to simplify the traditionally difficult task of distributed computing programming to increase the productivity and effectiveness of software engineers. Hadoop was designed for batch analytics, with large but finite data sets, and not streaming data sets that grow continuously over time.

3.2.1.3 Second Generation [2010 – 2017]

The rise of machine learning spurred the development of a second generation of general-purpose big data systems best exemplified by Apache Spark[37]. Machine learning algorithms not only require large volumes of data for successful training, but the process is an iterative one, requiring the application of a function repeatedly to this data for the optimization of a set of parameters, often through gradient descent approaches. Spark accelerates this process by keeping data in memory, whereas Hadoop writes the results of each mapper and reducer to disk. Where Hadoop leveraged the relative low cost of hard drives, Spark took advantage of the relatively low cost and, therefore, plentiful RAM or main memory available at the time the platform was architected. During this generation, the need for systems capable of handling streaming data arose, with data sets continuously growing without bound and requiring immediate processing to produce actionable output.

3.2.1.4 Third Generation [2017 - present]

Each generation of big data platforms was built considering the economics of solving the problems relevant in its day given the constraints of contemporary computing hardware. The state of the art has moved beyond both first generation—Hadoop’s general-purpose batch processing—and second generation—general purpose big datastores and processing frameworks such as Cassandra and Spark—big data platforms to third generation systems. Third generation platforms are purpose-built with specialized data structures and architectures optimized for a particular type of data, specific analytic use cases, and tailored to the eccentricities of particular industries. Not only are they far more efficient at processing their specific type of data and computing relevant

analytics, they demand far less effort by the analysts and engineers faced with those highly specialized problems.

For example, take the problem of hammering in a nail. While a Swiss Army knife may have a suitable attachment, a standalone hammer would be more efficient at the task because the knife is weighed down by its other capabilities. If the original problem is hammering in one thousand nails, the better tool would be a nail gun: a tool more complex than the simple hammer but purpose built for this scale of task to accelerate the worker's capabilities. For this reason, general purpose big-data platforms provide lots of flexibility with little optimization for specific use cases at scale. The focus of a 3rd generation big data platform is to solve a particular industry's unique problems *in a highly cost-effective fashion*.

3.2.2 Example Platforms

The following big data platforms were designed specifically with the electric industry in mind. Note that each leverages a large amount of open source software.

3.2.2.1 PNNL's Cloud Based Analytical Framework for Synchrophasor Data Analysis

PNNL has been working with BPA to develop a cloud-based framework for PMU Big Data analysis. The framework is based on the PNNL Institutional Cloud Computing OpenStack installation. The Hadoop Distributed File System (HDFS) is used to store the raw PMU information and Apache Spark is used for data analysis and ML. The aim of this work is to develop technologies and techniques that improve power system situational awareness and reliability. The computer cluster consists of 20 nodes including one master head node. Each node is equipped with eight core processors, 32 GB of RAM, and 100 GB of disk storage space.

The main functional components are diagrammed in Figure 3-1. PNNL receives the synchrophasor measurements as a real-time data stream from BPA, storing it at the PNNL's Electricity Infrastructure Operations Center (EIOC) as a set of PDAT-formatted files. The PDAT format, developed by BPA, is based on the IEEE Standard C37.118.2-2011 data frames and used by the utility company to capture PMU measurements from multiple devices in binary files. Each file

contains one minute of PMU data, collected at the 60 samples per second rate. More details on the PMU Big Data framework design can be found in [38][39].

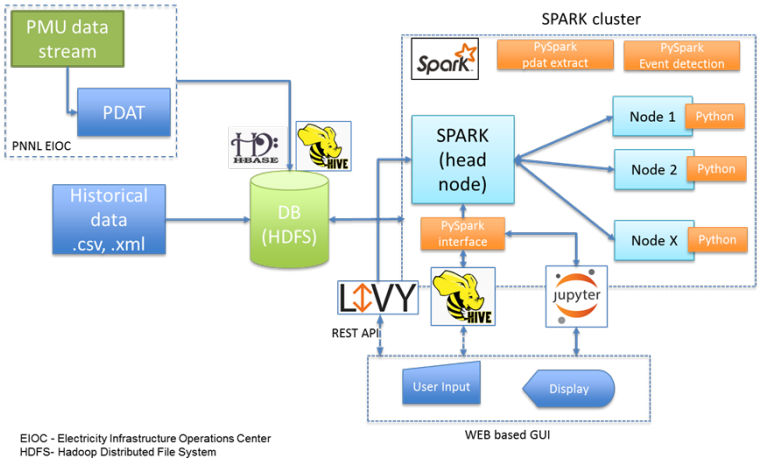

Figure 3-1. Cloud Based Framework [39].

3.2.2.2 PingThings' PredictiveGridTM – Universal Sensor Analytics Platform

PingThings' PredictiveGrid platform is a universal sensor analytics and AI platform designed for utilities. It is universal in that it was designed for any and all timestamped measurements that inform grid operators, planners, and designers, ranging from digital fault recorders sampling the voltage and current waveforms at 100KHz to residential smart meters reporting measurements every 15 minutes. To accomplish this goal, the platform is horizontally scalable and architected to ingest, store, access, visualize, analyze, and learn from (train machine learning and deep learning algorithms with) data captured by an arbitrary number and type of sensors measuring the grid with nanosecond temporal resolution. Early platform benchmarks demonstrated a throughput of over 50 million inserted values per second and 120 million queried values per second on a small, four-node cluster. More recent testing has shown that the platform can support over 100,000 simultaneous PMUs, each providing at least 20 streams of 60Hz data. This platform's system diagram is detailed in figure below.

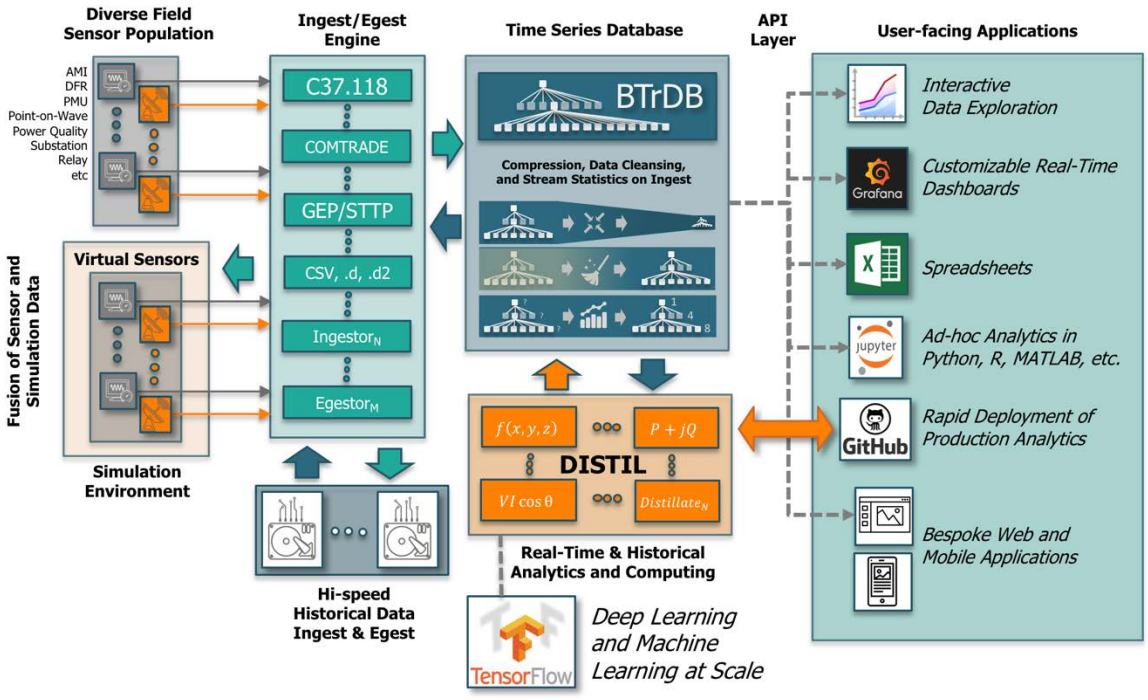

Figure 3-2 – A third generation big data platform architected specifically for sensor data and for the utility industry.

The platform can be decomposed into several functional areas. Moving from left to right in the diagram in the figure above, it supports the ingestion of both streaming and historical data archives in a wide range of formats at scale via the ingest engine. Data are ingested into a database specifically designed for dense time series sensor data, the Berkeley Tree Database (BTrDB)[40] whose development was funded by the ARPA-E Micro Synchrophasors for the Distribution System project. Further, the platform contains a distributed analytics and computational framework designed to operate across time series in parallel, executing significantly faster than real time to handle both real time and historical analyses and the training of machine learning and deep learning algorithms[41]. The platform provides numerous APIs that provide not only a direct connection for web applications including a data explorer, dashboards, and Jupyter Notebooks for ad-hoc analytics but also to utility planning and operations software, allowing for the seamless integration of highly novel algorithms with the real world.

4 Application of Data Mining Techniques with Synchrophasor Data – Use Cases

This section contains use cases that discuss methods and results from actual applications of data mining tools with power grid data, and especially synchrophasor data. There are simple cases, which rely on queries or previous knowledge of events or anomalies to make interesting discoveries. There are also more complex cases that use deep learning or more complex algorithms that allow the data to reveal interesting patterns or events.

Additional use cases and challenges are discussed by experts at data analytics workshops available at [42][43]. Additional use cases, not discussed in this report include: [44], [45], [46], [47], [48], [49], and [50].

PMU measurement data provide observations which were not possible before without the higher-resolution, time-synchronization, and phase inclusive angle measurements provided by PMUs. Respectively, data mining and data-driven applications using PMU data are evolving to utilize the embedded information in the measurements. The high volume, velocity, and variety of PMU measurement data make it possible to take advantage of new machine learning and statistical inference in applications such as short-time events and faults detections. This section reviews some of the recent applications of data mining techniques using synchrophasor data. These activities are still in the R&D maturity level, indicating the growing interest of the power systems community in the application of these techniques on synchrophasor data. Short summaries of the activities are listed next with the corresponding references available so that the reader can access more details if desired.

4.1 Event and Anomaly Detection

PNNL has developed several statistical and ML methods and applied them to large synchrophasor datasets to detect different types of events (e.g., frequency or voltage) and abnormalities. The first, relatively simple "engineering" approach is based on user-specified thresholds for signal values and duration. This method is commonly used by electrical utilities to detect system events [38]. The second approach that is based on the multi resolution wavelet analysis which separates one-dimensional signals into two-dimensional components that overlap in the time-frequency domain [51]. Wavelet-based multi-resolution analysis (MRA) uses wavelet function and scaling function to decompose and construct signals at various resolution levels, such that the anomaly phenomena can be detected and localized at each resolution level. The cluster analysis and Principal Component Analysis (PCA) for identifying similarities between the events were also developed. Clustering groups multiple objects by putting similar objects in the same group [51].

Two other related methods are: 1) An anomaly detection method based on dynamic regression models and applying a Kalman filter [52]; and 2) An ensemble based approach using statistical, clustering and regression approach as discussed in [53][54].

4.2 Data Integrity Situational Awareness Tool

PNNL has collaborated with BPA to develop a tool named "Data Integrity Situational Awareness (DISAT)". The tool aims to find atypical moments in streaming PMU data. The tool is based on a data driven, multi-variate statistical approach and relies on multi-processing (cluster computing) and big data techniques. It builds a baseline of typical behavior using past data, and then compares current data to that baseline. Those moments in time that are different from the typical behavior are flagged as atypical (unusual) moments. The tool allows for further drilling down of the data, to help determine what unusual behavior was found. This tool was completely developed using the statistical programming language R.

Figure 4-1 shows an example atypicality during the month of September using many prior months of PMU data to develop the baseline. The top plot shows that two atypicalities occurred during that month (atypicality score exceeded the cutoff value). The bottom plot shows a drill down plot of a voltage magnitude that was unusual. The gray/dark background in the plot shows typical behavior for this voltage magnitude, while the orange plot shows its values at the atypical moment of interest.

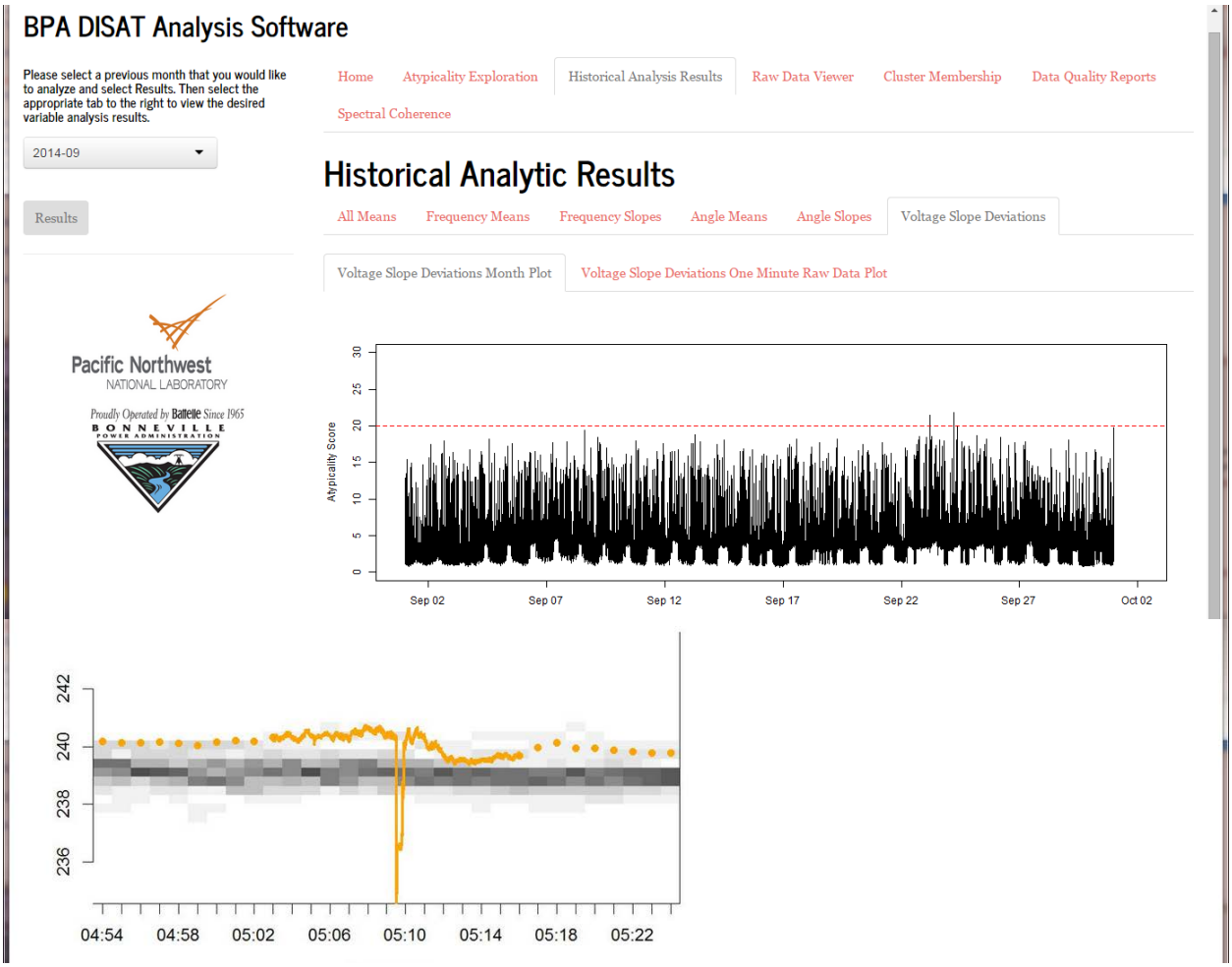

Figure 4-1. DISAT Example Results of Atypicality Score over Time and an Unusual Voltage Magnitude – PNNL [55].

4.3 A Systematic Approach for Dynamic Security Assessment and the Corresponding Preventive Control Scheme Based on Decision Trees

[56] proposes a decision tree (DT) based systematic approach for cooperative online power system dynamic security assessment (DSA) and preventive control. This approach adopts a new

methodology that trains two contingency oriented DTs on daily basis by the databases generated from power system simulations. Fed with real-time wide area measurements, one DT about measurable variables is employed for online DSA to identify potential security issues and the other DT about controllable variables provides online decision support on preventive control strategies against those issues. A cost-effective algorithm is adopted in this proposed approach to optimize the trajectory of preventive control. An importance sampling algorithm on database preparation is also proposed for efficient DT training for power systems with high penetration of wind power and distributed generation. The performance of the approach is demonstrated on a 400-bus, 200-line operational model of the western Danish power system.

Phasor measurement units (PMU) providing high resolution real-time measurements can be used in the proposed DT-based approach. The flowchart of the approach is shown in Figure 4-2 and the proposed approach is executed based on the following stages:

- Stage I: Identification of the Security Boundary.
- Stage II: Importance Sampling.
- Stage III: Offline time domain (T-D) Simulation and DT Training.
- Stage IV: Online Preventive Control.

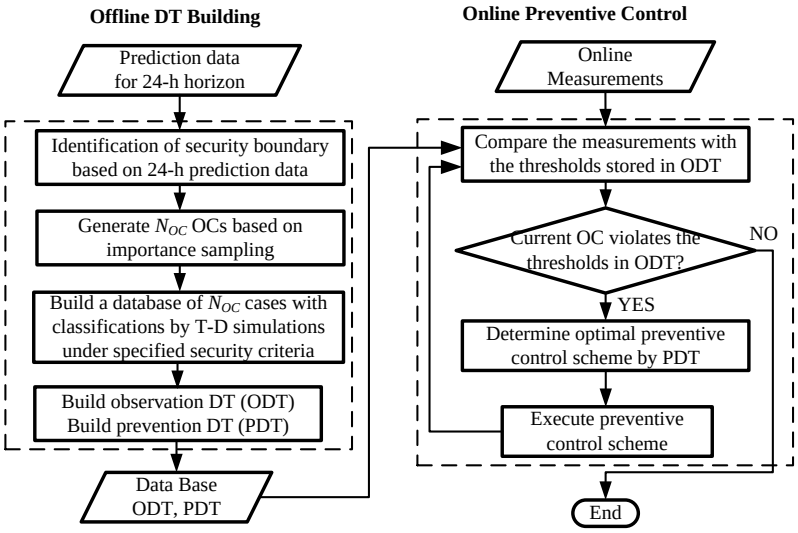

Figure 4-2. Flowchart of DT-based DSA and preventive control approach [56].

As shown in Figure 4-3, this systematic approach offline trains two paralleled Decision Trees (DTs) for each critical contingency based on 24-hour horizon system prediction data, such as load forecast, weather forecast, unit commitment-based generation plan, network topology as well as the unavailability of system elements due to scheduled maintenance, etc. Fed with online PMU and SCADA data, Observation Decision Tree (ODT) is employed for online Dynamic Security Assessment (DSA) to identify the margins of predictors against their thresholds determined from DT training. If any online measurements of predictors violate the thresholds, ODT would provide

situational awareness on insecurity if that contingency really happens. At the same time, Prevention Decision Tree (PDT) would provide system operators with preventive control schemes to drive the state to a new OC without insecurity under that contingency. Therefore, the parallel and cooperative utilization of PDT and ODT in the control schemes provides both the situational awareness and the preventive control against critical contingencies.

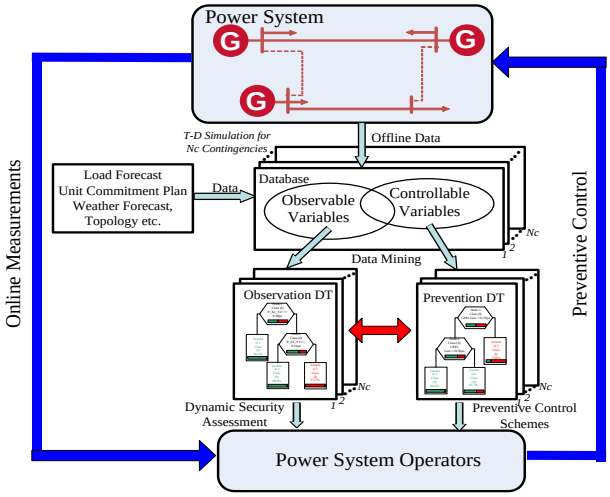

Figure 4-3. Systematic approach for DSA and preventive control scheme [56].

4.4 Synchrophasor-based Data Mining for Power System Fault and Failures Analysis

PMUs can provide high resolution and synchronized power system data, which can be effectively utilized for the implementation of data mining techniques. Data mining, based on pattern recognition algorithms can be of significant help for power system analysis, as high definition data are often complex to comprehend. Three pattern recognition algorithms are applied to perform data mining analysis in [57]. Fault data classification is first applied, then frequent faults are identified and finally the root cause of a fault is identified by clustering the parameters behind each scenario. For such classification three algorithms are chosen, k-Nearest Neighbor, Naïve Bayes and the k-means Clustering [57].

Fault analysis becomes more challenging in presence of failures in protection system. An approach for failures diagnosis in protection system with data mining approach integrated with physics based analysis was discussed in [58].

4.5 Using Phasor Data for Visualization and Data Mining in Smart-Grid Applications

[59] presents a density-based clustering (DBSCAN) technique to visualize and analyze smart-grid data including synchrophasor data. The technique is aimed to aid in detecting bad-data, various fault types, and deviation on frequency, voltage or current values for better situational awareness [59]. DBSCAN is a density based clustering algorithm. The algorithm grows regions

with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise. PMU data for various operating conditions are clustered by DBSCAN and presented in [59]. One clustering result for a line-to-ground fault-condition is shown in Figure 4-4.

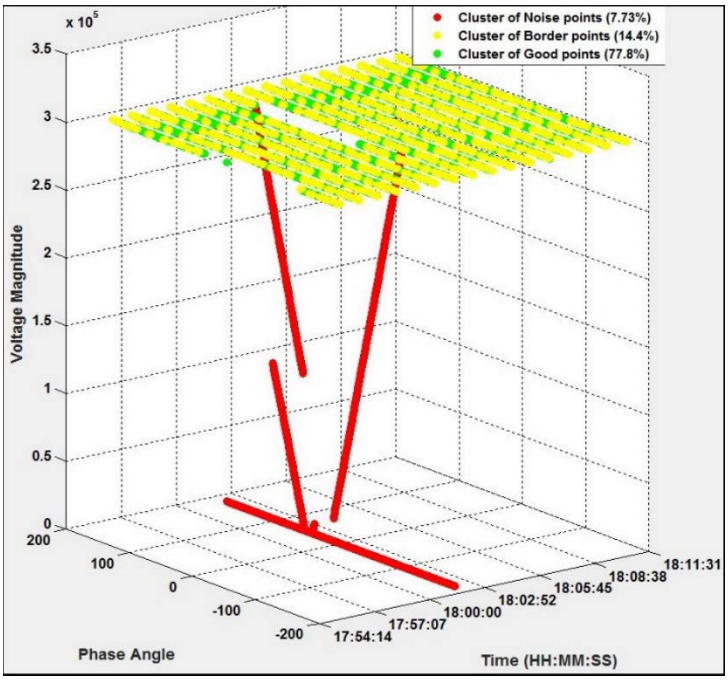

Figure 4-4. DBSCAN Clustering Results for a Line-to-Ground Fault [59].

4.6 Synchrophasor Data Baselining and Mining for Online Monitoring of Dynamic Security Limits

[60] develops a systematic approach to baseline phase-angles versus actual transfer limits across system interfaces and enable synchrophasor-based situational awareness (SBSA) [60]. Statistical methods are first used to determine seasonal exceedance levels of angle shifts that can allow real-time scoring and detection of atypical conditions. Next, key buses suitable for SBSA are identified using correlation and partitioning around medoid (PAM) clustering. It is shown that angle shifts of this subset of 15% of the network backbone buses can be effectively used as features in ensemble decision tree-based forecasting of seasonal security margins across critical interfaces [60]. The main steps for the proposed baselining study are shown in Figure 4-5 as follows:

- Stage I: Data Filtering and Decimation.
- Stage II: PAM Clustering.
- Stage III: Baselining using the Exceedance Levels.
- Stage IV: Predictive Baselining using Random Forest (RF) Models

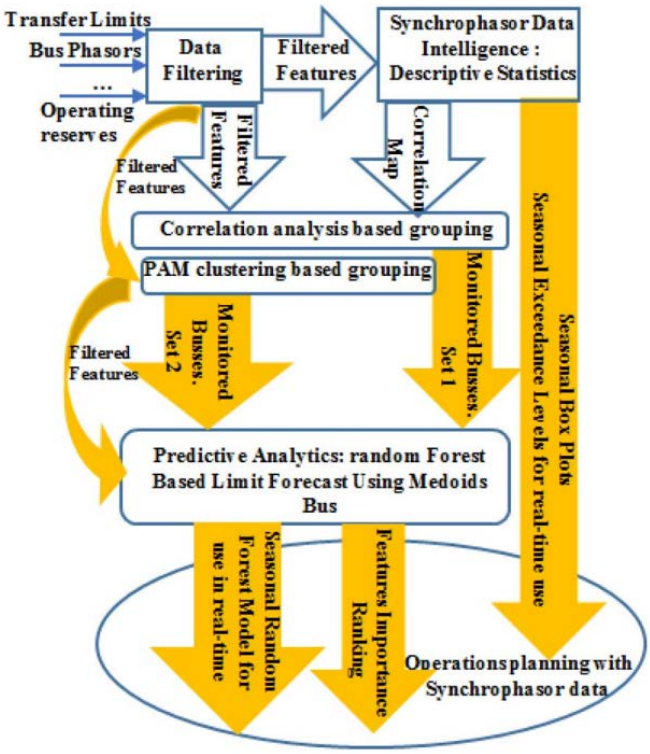

Figure 4-5. Proposed Framework for Synchrophasor Data Baselining Study [60].

4.7 Power System Data Management and Analysis Using Synchrophasor Data

[61] provides a synchrophasor data analysis methodology that leverages statistical correlation techniques in order to identify data inconsistencies, as well as power system contingencies [61]. This analysis includes the techniques for data management that are used to process the high-granularity and high-cardinality data gathered from PMUs. This work utilized real, archived PMU data obtained from the Western Electric Coordinating Council (WECC) in order to show that this methodology is not only feasible, but extremely useful for power systems monitoring, decision support, and planning purposes. The results presented indicate preliminary identification of PMU data issues, as well as power system instabilities [61].

The proposed correlation methodology is able to distinguish between bad PMU data and power system events. Specifically, the Pearson Product-Moment correlation method was used to determine how well data are linearly correlated. Figure 4-6 shows a sample visualization using the proposed correlation technique. Each coordinate (square) represents the correlation coefficient of the two PMUs that make up its coordinates. The color of the square represents how close the correlation is to 1 or −1, and the sign at the coordinate represents either positively correlated or inversely correlated PMU pairs. Typically, a magnitude of correlation above 0.4 −

0.5 is considered correlated. Thus any squares depicting blue shades would be considered de-correlated [61].

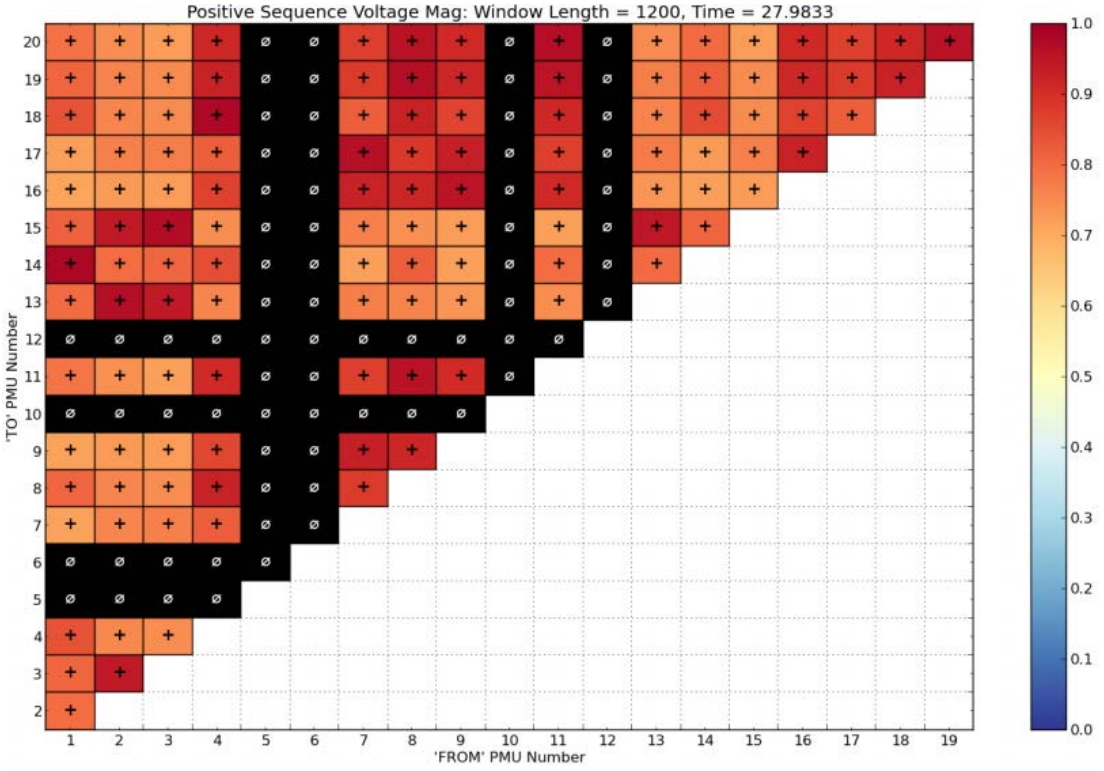

Figure 4-6. Example PMU Correlation Visualization (Topological Distance) over 1200 Data Points (20 Seconds) [61].

4.8 Online Dynamic Security Assessment with Missing PMU Measurements: A Data Mining Approach

A data mining approach using ensemble decision trees (DTs) learning is proposed in [62] for online dynamic security assessment (DSA), with the objective of mitigating the impact of possibly missing PMU data [62]. Specifically, multiple small DTs are first trained offline using a random subspace method. In particular, the developed random subspace method exploits the hierarchy of wide-area monitoring system (WAMS), the locational information of attributes, and the availability of PMU measurements, so as to improve the overall robustness of the ensemble to missing data. Then, the performance of the trained small DTs is re-checked by using new cases in near real-time. In online DSA, viable small DTs are identified in case of missing PMU data, and a boosting algorithm is employed to quantify the voting weights of viable small DTs. The security classification decision for online DSA is obtained via a weighted voting of viable small DTs. A case study using the IEEE 39-bus system demonstrates the effectiveness of the proposed approach [62]. Figure 4-7 illustrates the test results for online DSA in case of missing PMU measurements. It compares multiple approaches including DT using surrogates, random

forest with and without surrogates and the proposed approach. It is observed that the proposed approach has much better performance than others.

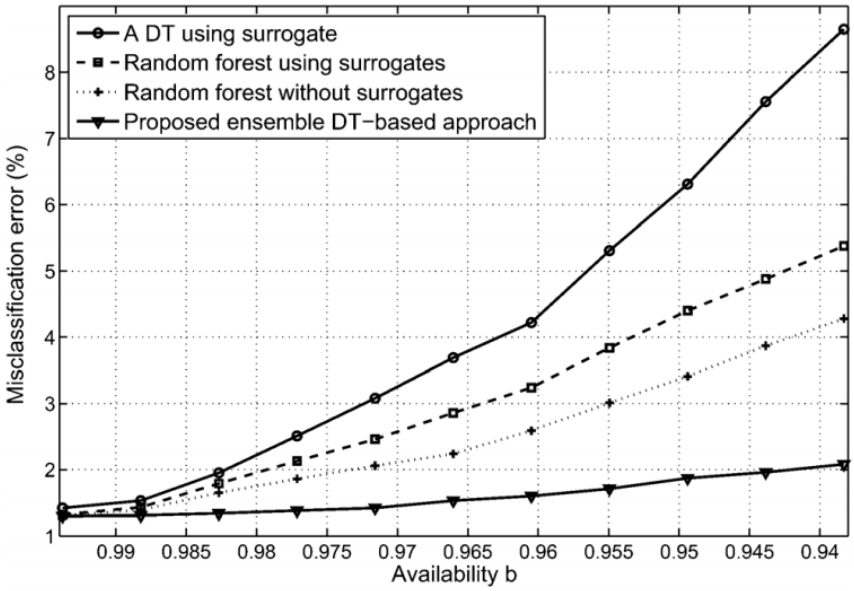

Figure 4-7. Performance of Online DSA in Case of Missing PMU Measurements [62].

4.9 Online Calibration of Phasor Measurement Unit Using Density-based Spatial Clustering

Global Energy Interconnection Research Institute North America (GEIRI North America) developed a framework named “Power System Parameter Calibration System (PSPCS)” for PMU data calibration. The proposed approach for PMU data calibration is an online calibration of the overall bias error in PMU data without knowing the exact system model. An unsupervised data mining technique named “density-based spatial clustering of applications with noise (DBSCAN)” is used in the proposed approach. PSPCS was tested by using both simulated data and real PMU data. Figure 4-8 shows a test result using real PMU data. PSPCS is now implemented at the grid dispatch center of Jiangsu Electric Power Company, China [63].

	Calculated	Calculated Error in		
	p.u. or rad ($\times 10^{-3}$)	R (%)	X (%)	B_C (%)
∂V_s	-0.0032			
∂V_r	-0.0033			
∂I_s	0.0171			
∂I_r	0.0164	-14.0	6.4	12.6
$\partial \theta'_{Vs}$	0.0003			
$\partial \theta'_{Vr}$	0.0027			
$\partial \theta'_{Is}$	-5.6238e-5			

Figure 4-8. Test Result Summary by Using Real PMU Data [63].

4.10 SRP/ASU PMU-Based Online Monitoring of Critical Power System Assets

To go beyond the readily visible values (phasor magnitudes & angles) presented by synchrophasors raw data, people also design new metrics/statistics that can be calculated out of the synchrophasors and use them to facilitate data mining.

Salt River Project (SRP) and Arizona State University (ASU) are developing a new analytics tool to predict major power system asset failure with synchrophasor data [64]. The project goes beyond the standard vision of synchrophasors – voltage magnitude, voltage angle, current magnitude, current angle, imaginary current and real current, etc., it proposes to leverage the metric called Signal-to-Noise-Ratio (SNR) as the main data signature. SNR can be calculated for each signal stream.

Progress has been made to reveal the failure of a major piece of equipment in SRPs system in June of 2016, the 500/230kV Rudd transformer. According to the SNR that is computed at different times away from the transformer failure (one year ago, one month ago and the same day), as depicted in Figure 4-9, noticeable pattern differences are observed, which can be leveraged for potential asset failure prediction.

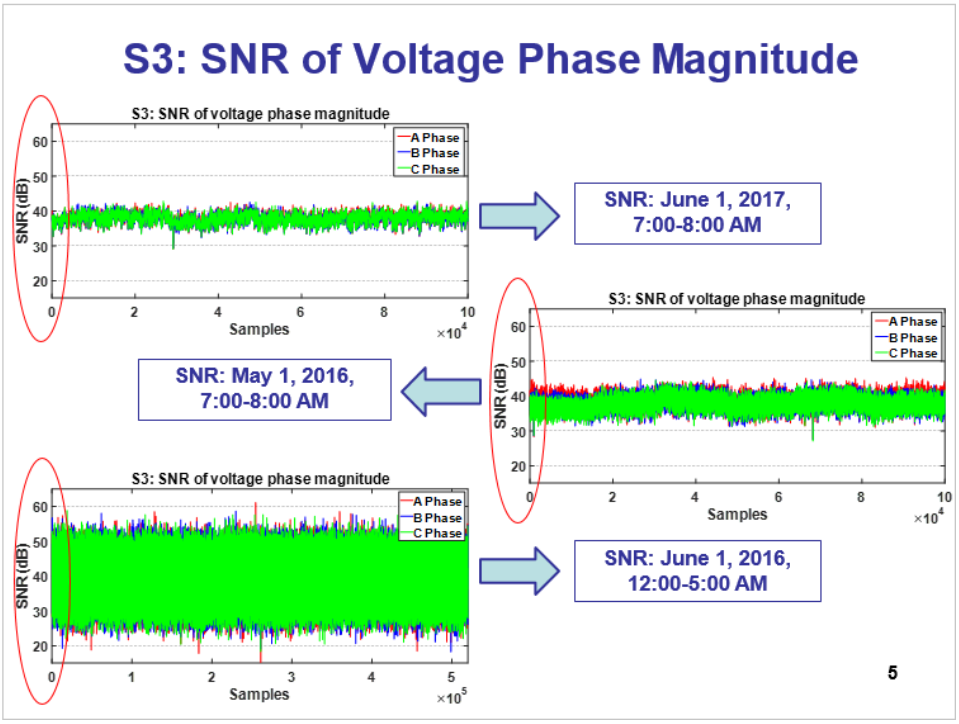

Figure 4-9 SNR of voltage phase magnitude from a neighboring substation of the Rudd transformer at 3 different time periods [64]

Failure in protection system assets were analyzed and monitored using DBSCAN approach as outlined in [58][65].

4.11 PMU-Based Load Monitoring with anomaly detection

High resolution PMU data can be used to track load parameters but data quality is challenging. A novel adaptive search-based algorithm to estimate load model parameters is presented in [66][67]. A static load model is used with the Z (constant impedance), I (constant current), and P (constant power) components of the load. Prony analysis and adaptive window based approach were developed to eliminate anomalies in the input data for accurate estimation of the load parameters.

5 Conclusions

The large amount of synchrophasor data produced in the daily operation of modern power systems has become a valuable resource for the development of advanced applications for both real-time operational, as well as offline planning environment.

An area of research that has recently attracted the interest of the power systems industry, is the application of artificial intelligence, machine learning and data mining techniques with high resolution time synchronized measurements. Application of data mining has been growing recently in various industries with many success stories.

This report summarizes the state-of-the-art in data mining and big data analytics, and documents ongoing R&D activities and use cases in the power grid using synchrophasor data. Several data mining techniques are briefly introduced in chapter 2. Chapter 3 provides an overview of the state-of-the-art data mining software and platforms. Both open source and commercial tools are covered.

Chapter 4 reviews several R&D use cases that adopt data mining techniques with synchrophasor data analysis. These use cases comprise both industrial R&D and academic research efforts, and serve as good demonstrations of application of data mining techniques and tools with synchrophasor data.

6 References

[1] A primer on synchrophasors and phasor values is provided in the report "Synchrophasor Technologies and their Deployment in the Recovery Act Smart Grid Programs" dated August 2013. https://www.smartgrid.gov/recovery_act/program_impacts/applications_synchrophasor_technology

[2] A.G.Phadke, J.S.Thorp, M.Adamiak, "A New Measurement Technique for Tracking Voltage Phasors, Local System Frequency, and Rate of Change of Frequency," IEEE Transactions on PAS, May 1983.

[3] D. T. Rizy, et. al., "The Future of GPS-Based Electric Power System Measurements, Operation and Control", Proceedings of the 11th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 1998), Nashville, TN, September 15 - 18, 1998.

[4] Modified from R.F. Nuqui, "State Estimation and Voltage Security Monitoring Using Synchronized Phasor Measurements", Doctorate Dissertation, Virginia Polytechnic Institute, Blacksburg, VA, July 2, 2001.

[5] "Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations," US-Canada Power System Outage Task Force.

[6] "Micro-synchrophasors for distribution systems," Alexandra Von Meier, David Culler, Alex McEachern, Reza Arghandeh, IEEE Innovative Smart Grid Technologies Conference (ISGT), 2014.

[7] Partha Sarathi Kar, "Principal Component Analysis", Linkedin Slide Share, [Online]. Available at: https://www.slideshare.net/ParthaSarathiKar3/principal-component-analysis-75693461

[8] Marina Meil˘a, "Is Manifold Learning for Toy Data Only?" [Online]. Available at: https://www.stat.washington.edu/mmp/Talks/mani-MMDS16.pdf

[9] "DBSCAN: What is a Core Point?" Cross Validated Question. [Online]. Available at: https://stats.stackexchange.com/questions/194734/dbscan-what-is-a-core-point.

[10] M. Ivan, "Classification using k-Nearest Neighbors in R", [Online]. Available at: https://en.proft.me/2017/01/22/classification-using-k-nearest-neighbors-r/.

[11] "Decision Tree Maker: Quickly and easily create decision trees and more", SmartDraw, LLC. [Online]. Available at: https://www.smartdraw.com/decision-tree/decision-tree-maker.htm.

[12] V. Vapnik, S.E. Golowich, and A. Smola. (1996). Support vector method for prediction, regression estimation, and signal processing. *Advances in Neural Information Processing Systems.* San Mateo, CA: Morgan Kaufmann Publishers.

[13] C.M, Bishop (1995). *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

[14] A.J. Smola and B. Schölkopf (2004). A tutorial on support vector regression. *Statistics and Computing*, **14**: 199-222.

[15] L. Cao, and F.E.H, Tay. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks* **14** (6): 1506-1518.

[16] V. Cherkassky, and Y. Ma. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, **17** (1): 113-126.

[17] J. Ma, T. James, and P. Simon. (2003). Accurate on-line support vector regression. *Neural Computation*, **15**: 2683-2703.

[18] O.A. Omitaomu. (2006). On-line Learning and Wavelet-Based Feature Extraction Methodology for Process Monitoring using High-Dimensional Functional Data. University of Tennessee, Knoxville, Ph.D. Dissertation.

[19] O.A. Omitaomu. (2013). Intelligent Process Monitoring and Control Using Sensor Data. Germany: Lap Lambert Academic Publishing, March.

[20] C.M. Bishop (1995). Neural Networks for Pattern Recognition. Oxford University Press, London, UK.

[21] S. Haykin (1994). Neural Networks - A Comprehensive Foundation. Maxwell MacMillian Int., New York.

[22] D. Ripley (1996). Pattern Recognition and Neural Networks. Cambridge University Press. Cambridge.

[23] "The Comprehensive R Archive Network", CRAN Project. [Online]. Available at: https://cran.r-project.org/.

[24] T. Hothorn, "CRAN Task View: Machine Learning & Statistical Learning", CRAN Project. [Online]. Available at: https://cran.r-project.org/web/views/MachineLearning.html/

[25] "scikit-learn: Machine Learning in Python", Scikit-learn Home Webpage. [Online]. Available at: http://scikit-learn.org/stable/#.

[26] D. Albanese, R. Visintainer, S. Merler, S. Riccadonna, G. Jurman, C. Furlanello. mlpy: Machine Learning Python, 2012.

[27] "Lightning Fast Data Science Platform", RapidMiner. [Online]. Available at: https://rapidminer.com/.

[28] "Weka 3: Data Mining Software in Java", Machine Learning Group at the University of Waikato. [Online]. Available at: https://www.cs.waikato.ac.nz/ml/weka/index.html.

[29] "Data Mining Fruitful and Fun". Orange. [Online]. Available at: https://orange.biolab.si/.

[30] "SAS Enterprise Miner", SAS. [Online]. Available at: https://www.sas.com/en_us/software/enterprise-miner.html.

[31] "IBM DB2 Intelligent Miner for Data", IBM Knowledge Center. [Online]. Available at: https://www.ibm.com/support/knowledgecenter/en/SSEPGG_9.5.0/com.ibm.im.overview.doc/c_ibm_db2_intelligent_miner_for_data.html.

[32] "What is Stream Analytics?", Microsoft Azure. [Online]. Available at: https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction.

[33] "Streaming Analytics: Details", IBM. [Online]. Available at: https://www.ibm.com/cloud/streaming-analytics/details.

[34] "Top 18 Open Source and Commercial Stream Analytics Platforms", Predictive Analytics. [Online]. Available at: https://www.predictiveanalyticstoday.com/top-open-source-commercial-stream-analytics-platforms/.

[35] Ghemawat, S., Gobioff, H., and Leung, S. "The Google File System", Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles - SOSP '03(2003)

[36] Dean, J. and Ghemawat, S. 2004. "MapReduce: simplified data processing on large clusters", In Proc. of the 6th conference on Symposium on Operating Systems Design & Implementation - Volume 6 (OSDI'04), Vol. 6. USENIX Association, Berkeley, CA, USA, 10-10.

[37] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2010. “Spark: cluster computing with working sets”, In Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (HotCloud'10). USENIX Association, Berkeley, CA, USA, 10-10.

[38] Etingov PV, Z Hou, H Wang, H Ren, DV Zarzhitsky, J De Chalendar, D Kosterev, AJ Faris, and S Yang. 2017. "Cloud Based Analytical Framework for Synchrophasor Data Analysis." In CIGRE US National Committee 2017 Grid of the Future Symposium, Cleveland, Ohio, October 22-25, 2017.

[39] PNNL and BPA, “PMU Big Data Analysis Based on the SPARK Machine Learning Framework”, presented at JSIS Meeting, May, 2017. [Online]. Available at: https://www.wecc.biz/Administrative/08%202017-05-23%20JSIS%20Spark%20ML-Etingov.pdf.

[40] Andersen M and Culler D, “BTrDB: Optimizing Storage System Design for Timeseries Processing”, Fast '16 14th USENIX Conference on File and Storage Technologies, Feb 2016

[41] Andersen M, Kumar S, Brooks C, von Meier A, and Culler DE. 2015. “DISTIL: Design and implementation of a scalable synchrophasor data processing system”, In Smart Grid Communications, 2015 IEEE International Conference on. IEEE, 271–277

[42] Workshop on “Data Analytics for the Smart Grid”, August 28th, 2017, Pullman, WA, Proceedings Available at: https://sgdril.eecs.wsu.edu/workshop_conferences/data-analytics-for-the-smart-grid-dasg/

[43] Workshop on “Real Time Data Analytics for the Resilient Electric Grid”, August 4-5, 2018, Portland, WA, Proceedings Available at: https://sgdril.eecs.wsu.edu/workshop_conferences/real-time-data-analytics-for-the-resilient-electric-grid/Grids, 2017

[44] Luca Schenato, Grazia Barchi, David Macii, Reza Arghandeh, Kameshwar Poolla, and Alexandra von Meier. “Bayesian linear state estimation using smart meters and pmus measurements in distribution grids.” In International Conference on Smart Grid Communications (SmartGrid-Comm), pages 572–577. IEEE, 2014.

[45] Reza Arghandeh, Martin Gahr, Alexandra von Meier, Guido Cavraro, Monika Ruh, and Goran Andersson. “Topology detection in microgrids with micro-synchrophasors”. In Power & Energy Society General Meeting. IEEE, 2015.

[46] G Cavraro, R Arghandeh, “Power Distribution Network Topology Detection with Time-Series Signature Verification Method”, IEEE Transactions on Power Systems, 2017

[47] Reza Arghandeh, Yuxun Zhou, “Big Data Application for Power Systems”, Book, Elsevier, Oxford, UK, 2017.

[48] Y. Zhou, R. Arghandeh, and C. Spanos, “Partial knowledge data-driven event detection for power distribution networks”, IEEE Transactions on Smart Grid, pp. (99), 2017.

[49] Y. Zhou, R. Arghandeh, I. C. Konstantakopoulos, S. Abdullah, A. von Meier, and C. J. Spanos, “Abnormal event detection with high resolution micro-pmu measurement”, In IEEE Power Systems Computation Conference. IEEE, 2016.

[50] Y. Zhou, R. Arghandeh, and C. J. Spanos. Online learning of contextual hidden markov models for temporal-spatial data analysis. In IEEE Conference on Decision and Control (CDC 2016), 2016.

[51] Ren H., Hou Z., Wang H., Zarzhitsky D., Etingov P. “Pattern Mining and Anomaly Detection based on Power System Synchrophasor Measurements,” in Proceedings of the 51st Hawaii International Conference on System Sciences, 2018.

[52] Ren H., Hou Z., Etingov P. “Online Anomaly Detection Using Machine Learning and HPC for Power System Synchrophasor Measurements,” PMAPS conference, June 24-28, 2018 (accepted).

[53] M. Zhou, Y. Wang, A. K. Srivastava, Y. Wu, and P. Banerjee. "Ensemble based Algorithm for Synchrophasor Data Anomaly Detection." *IEEE Transactions on Smart Grid* (2018).

[54] A. Srivastava, “SyncAD: Ensemble Based Data Mining Tool for Anomaly Detection In PMU data and Event Detection”, Joint Synchronized information Subcommittee, Oct. 11-13, 2017 Westminster, CA

[55] PNNL, “Phase Angle Monitoring Using DISAT”, presented at JSIS Meeting, May, 2017. [Online]. Available at: https://www.wecc.biz/Administrative/08%202017-05-23%20JSIS%20Phase%20Angle%20Monitoring%20Using%20DISAT-Amidan.pdf.

[56] C. Liu, K. Sun, et al, “A systematic Approach for Dynamic Security Assessment and the Corresponding Preventive Control Scheme Based on Decision Trees”, IEEE Trans. Power Systems, vol. 29, No. 2, pp 717-730, March 2014.

[57] M. Al Karim, M. Chenine, K. Zhu and L. Nordstrom (2012). Synchro-phasor based Data Mining for Power System Fault Analysis. 2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe), Berlin, p:1-8.

[58] B. Cui, A. Srivastava, and P. Banerjee. "Automated Failure Diagnosis in Transmission Network Protection System Using Synchrophasors." *IEEE Transactions on Power Delivery* (2018).

[59] A. Mukherjee, R. Vallakati, V. Lachenaud, and P. Ranganathan (2015). Using phasor data for visualization and data mining in smart grid applications. IEEE First International Conference on DC Microgrids, p:13-18.

[60] Anissa Kaci, Innocent Kamwa, Louis-A. Dessaint, and Sébastien Guillon (2014). Synchro-phasor data baselining and mining for online monitoring of dynamic security limits. IEEE transactions on power systems, Vol. 29, No. 6, p:2681-2695.

[61] Rich Meier, Ben McCamish, David Chiu and Miles Histand (2014). Power system data management and analysis using synchro-phasor data. IEEE conference on technologies for sustainability, p:225-231.

[62] Miao He, Vijay Vittal and Junshan Zhang (2013). Online Dynamic Security assessment with missing PMU Measurements: A Data Mining Approach. IEEE transactions on power systems, Vol. 28, No. 2, p: 1969-1977.

[63] Di Shi, Xinan Wang, Zhiwei Wang, Xiao Lu, ChunleiXu and ZhihongYang (2017). Online Calibration of Phasor Measurement Unit Using Density-based Spatial Clustering. NASPI group meeting.

[64] Malhar Padhee, Anamitra Pal and Matthew Rhodes, “PMU-based Online Monitoring of Critical Power System Assets” [Online] Available at: https://www.wecc.biz/Administrative/13_PMU-Based%20Online%20Monitoring%20of%20Critical%20Power%20System%20Assets_JSIS-May%202018.pdf

[65] A. Srivastava, “Failure Diagnosis and Cyber Intrusion Detection in Transmission Protection System Assets using Synchrophasor Data”, NASPI meeting, March 2017

[66] Tushar, S. Pandey, A. Srivastava, P. Markham, and M. Patel, “Online Estimation of Steady-State Load Models Considering Data Anomalies”, IEEE Transactions on Industry Applications, 2017

[67] Tushar, H. Lee, P. Banerjee, and A. K. Srivastava, “Synchrophasor Applications for Load Estimation and Stability Analysis”, IET Power and Energy Series, Synchronized Phasor Measurements for Smart