Ensemble based Algorithm for Synchrophasor Data Anomaly Detection

M. Zhou, *Student Member, IEEE,* Y. Wang, *Student Member, IEEE,*
A. K. Srivastava, *Senior Member, IEEE,* Y. Wu, *Member, IEEE,* P. Banerjee, *Member, IEEE*

***Abstract*—With the advent of Phasor Measurement Units (PMUs), high resolution synchronized phasor measurements enables real time system monitoring and control. PMUs transmit data to local controllers in substations and Phasor Data Concentrators for system wide monitoring and control application in the control center. They provide real-time phasor data for critical power system applications such as remedial action schemes, oscillation detection and state estimation.**

The quality of phasor data from PMUs is critical for smart grid applications. Several methods are developed to detect anomalies in time series data, tailored for PMU data analysis. Nevertheless, applying stand alone methods (with fixed parameters) takes great tuning effort, and does not always achieve high accuracy. In this paper, we adopt an unsupervised ensemble learning approach to develop fast, scalable bad data/event detection for PMU data. The ensemble method invokes a set of base detectors to generate anomaly scores of the PMU data, and makes decisions by aggregating the scores from each detector. We develop two algorithms: 1) a learning algorithm that trains the ensemble model, and 2) an online algorithm that infers the anomaly scores with the ensemble model over PMU data streams. The proposed method provides flag for data anomalies and triggers further analysis to differentiate between events and bad data. Using both simulated and real-world PMU data, we show that our ensemble model can be efficiently trained, can achieve high accuracy in detecting diversified errors/events, and outperforms its counterparts that use single standalone detection method.

I. Introduction

IDentifying anomalies in PMU data is critical for reliable power system operations. Phasor data from PMUs are typically used for power grid monitoring, limited control and enhanced analysis [1]. Bad data detection in linear state estimator or hybrid state estimation is still evolving for centralized application [2], and may require data pre-processing. The local or decentralized application requires bad data detection without the need of specific system models. Traditional bad data detection in PMU is based on status flag reported by the PMUs based on IEEE C37.118 data format and limited statistical techniques as discussed later. The anomalies in PMU data can be modeled as outliers in time series data (i.e., a sequence of data points). Several outliers detection techniques have been developed [3]–[5], which can be applied for the anomaly detection in time series PMU data. In [6], Phasor Data Conditioning Application (PDCA) is proposed to filter out low quality measurements. In [7], a phasor-measurements based state estimation approach is introduced to correct systematic phase angle biases and current scaling errors. A Kalman filter based data conditioning algorithm is developed to clean field synchrophasor data in [8]. Classification of faults and events on historical data has been studied in [9] and the results are correlated with protective relay reports.

Mengze Zhou, Yuhui Wang, Anurag Srivastava, Yinghui Wu and Paramarshi Banerjee are with the School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA 99163 USA. (E-mail: asrivast@eecs.wsu.edu)

This work is partially supported by the Power System Engineering Research center (PSERC), Electric Power Research Institute (EPRI), NSF IIS-1633629 and Huawei Innovation Research Program (HIRP). The views and opinions of authors expressed herein do not necessarily state or reflect any funding agency thereof.

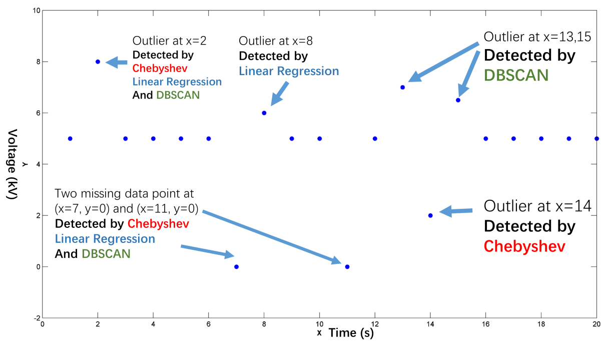


Fig. 1: Outliers identified by Linear Regression-based, Chebyshev-based and DBSCAN-based outlier detection method, respectively; "No single winner".

Anomaly detection in PMU data, nevertheless, is nontrivial. (1) There is often "no single winner" among standalone outlier detection methods that can identify all types of outliers with high accuracy, due to the heterogeneity of PMU anomalies (e.g., bad data, events, missing data). (2) To use each standalone method, great manual effort has to be made to tune the parameters to achieve reasonable accuracy. (3) Not all the methods are feasible for detecting outliers over real-life PMU data, especially under quality and response time constraints. Moreover, anomaly detection in PMU data often lacks the ground truth. Indeed, it is infeasible to require the true labels over the fast PMU data streams.

Example. Consider a segment of real-world PMU data depicted in Figure 1. To detect the data anomalies, three outlier detection methods are implemented, notably, Linear Regression (marked in blue), DBSCAN [11] (marked in green) and Chebyshev [12] (marked in red). The parameters for all the base detectors are tuned carefully to demonstrate their best performance over the sample PMU data. Seven data points that are validated as outliers are marked.

We observe the following.(1) No single method can detect all the validated outliers. For example, Linear Regression-based detector only identifies the outlier at timestamps x= $\{2, 7, 8, 11\}$ with a linear model, but missed the one at x= $\{13, 14, 15\}$, due to the fact that the model no longer fits the data. Similarly, Chebyshev-based detectors can only detect the anomaly data at x=$\{2, 7, 11, 14\}$. (2) The performance of a single detector is sensitive to the setting of its parameters. For example, outliers at x=$\{13, 15\}$ can be detected by DBSCAN-based detector with a proper set of model parameters (*e.g.,* the density of the neighborhood to be counted for a cluster), but can no longer be reported by the same detector with different parameters. Moreover, it takes a great effort for even domain experts to tune many parameters to get the best results.

These challenges motivate the need of fast and effective anomaly detection method for PMU data. In particular, *(1) How to capture different types of anomalies in PMU data? (2) How to deploy the method over PMU data streams for online anomaly detection?*

An "Ensemble detector" can automatically selects and combines the results of a set of "base detectors". Each base detector implements a single outlier detection method. The ensemble detector may yield more accurate results by learning the importance of different based detectors. Moreover, it reduces the manual effort for parameter tuning without losing detection accuracy, and allow more based detectors to be easily plugged in to further improve the detection accuracy. As verified by our experimental study, the ensemble detector outperforms its counterpartsthat use base detectors individually, and achieves good accuracy with little effort for parameter tuning, which is required by each base detector.

While desirable, the training of the ensemble detector is very challenging. (1) Unlike supervised classifier assembling, it is very hard to obtain labeled training examples for PMU data. This requires us to develop unsupervised approaches. (2) Different detectors provide results based on their own standards. It is hard to combine these scores without proper normalization. (3) The proper combination of the results from different base detectors requires an effective learning algorithm. Moreover, the learning and inference process must be fast to support online decision making for PMU data.

Contribution. In this work, we developed robust and efficient ensemble-based anomaly data detection techniques for PMU bad data detection.

(1) Given a set of base detectors and PMU dataset to be analyzed, an ensemble detector first chooses a set of base detectors with reasonable performance, and then learns a weight for each detector against a given set of training examples. It finally combines the weighted results from each detector to determine the final outlier score.

(2) We introduce effective learning and inference algorithms for the ensemble-based detector. (a) The learning method normalizes the anomaly scores from different base detectors, and trains the ensemble detector with a maximum likelihood estimation (MLE) model, which is further approximated by expectation maximization procedure (EM). A sliding window-based algorithm is developed to support online learning. (b) An efficient parallel inference algorithm is developed over PMU data streams, to perform online anomaly detection.

(3) As a proof of concept, an ensemble detector is built on a set of well-established outlier detectors to detect bad data in PMU data. Three classes of base detectors, notably, Chebyshev-based, DBSCAN-based, and Regression-based detectors are selected in this work. The performance of the proposed method is evaluated based in terms of time cost and learning accuracy.

(4) Using simulated data and real-world data, the effectiveness of the ensemble detector is experimentally verified. It is demonstrated that the detector has four attractive features. (a) **Effectiveness**. Multi-model based ensemble learning is more robust (achieving higher precision/recall) than all (or most) single-model counterparts; (b) **Feasibility**. The ensemble detector can be trained with affordable time. (c) **Scalability**. The ensemble detector achieves reasonable results within limited time and memory bound. (d) **Usability**. It does not require expensive manual effort for parameter tuning. Moreover, additional detectors can be easily plugged in to improve performance.

The proposed ensemble based anomaly detection algorithms can be readily generalized to other high-resolution data. The ensemble methods are particularly needed and useful for PMU data anomaly detection for several cases.

(a) State estimation runs less frequently as compared to the application time step requirement in the control center (e.g. oscillations monitoring using PMU data running every 30 second while state estimation runs every 2 minutes). Additionally, PMU data may not be enough to run complete linear state estimation at the control center.

(b) Distributed state estimation is running less frequently compared to the application requirements at the substation (e.g. response based RAS running every second while distributed state estimation runs every 30 seconds).

(c) The control center application uses the raw PMU data in the absence of state estimation. Any applications using raw PMU data may not perform well in the presence of anomalies.

(d) The third zone of distance protection may result in occasional mal-operation causing unintended blackouts. Several literature proposed to disable the third zone protection, however, such propositions are still not adopted in the industry [16]. The typical operating delays for third zone relay operation are around 0.8 second [10], which are much longer than the processing time of the proposed method provided in Section V-C. In such scenarios, the proposed method can reduce the chances of third zone relay mal-operation resulting due to data anomalies. Also, one should be carful in applying PMU data for protection as performance of PMU may be in issue immediately following the fault. Third zone operation with zone blocking and coordinated protection with longer timeline will be more suitable to use PMU data and developed anomaly detection compared to zone 1 or zone 2 operation.

(e) The Remedial Action Schemes are critical protection infrastructure, installed in the transmission system [14] for protecting the system from large scale blackouts. The protection of transmission line from thermal overload is carried out by thermal overload RAS [21], which dynamically estimate the line ratings using metrology data and also phasor measurements from PMUs. The response time of thermal overload RAS varies from 10 seconds to a few minutes, which provides sufficient time for applying anomaly detection of phasor data using the proposed method.

(f) The need to filter out bad PMU data for critical applications (e.g. PMU based distributed and coordinated control).

(g) In the presence of bad data, pre-processing PMU data using linear state estimator yields inferior performance.

(h) It is observed that the performances of conventional general-purpose outlier detectors are quite sensitive to PMU data due to diversity, noise and random variations, thus requires great effort to be tuned to fit the anomaly detectors to work with PMU data; and this might be applicable to other high-resolution data.

(i) The proposed model in this work can be easily integrated with the external standard anomaly detection method.

Related Work. We categorize related work below.

Bad data detection. Data mining and machine learning techniques are applied to clean the PMU data before more advanced analytics are conducted. Preprocessing of PMU data is needed to detect missing data and outliers, as the bad data leads to significant negative influences on PMU based analysis. A number of bad data detection methods have been applied to clean PMU data. A moving median algorithm is used in [5], [13] [15] introduce a density based online detection algorithm. These methods serve as single stand alone detectors, and can be "plugged in" to the proposed ensemble detector.

Outlier mining. There have been a host of work on detecting outliers over time series data and event sequences [11], [12]. An outlier detector takes as input a time series, and identify the data points that are significantly different from others, by returning either a ranked list (the higher the outlier score is, the more likely the data point is an outlier), or a list of scores.

Ensemble outlier detection. While traditional ensemble method for supervised classifiers has been well studied (e.g., AdaBoost) [17], developing ensemble techniques for unsupervised learners is relatively new and much more challenging. Ensemble method has been reported for classification of power system security in [18]. As observed in [13], it is not straightforward to apply the existing classifier ensemble strategy over outlier detection. As remarked earlier, the challenges include detector selection, score normalization and combination. Another challenge for PMU data outlier detection is that the ensemble needs to respond in real-time.

In this work, the problem of base detector selection to the ensemble learning is addressed. It is also envisioned that this work will provide a scalable and effective tool for large-scale PMU data analysis.

II. Outlier detection: base detectors

We start with PMU data model and outliers, followed by three standalone methods for outlier detection. Our ensemble-learning based method supports any other applicable outlier detection methods as "plug-in"s.

PMU data. The PMU dataset is a sequence of data point $X = d_1, \ldots, d_N$ over time window with length N. Each data point $d_i \in X$ is associated with a time stamp t_i. In this work, voltage magnitude is selected as the input to test the ensemble method.

PMU outliers. The outliers in a data set refer to the data points that show significant diversion from the expected behavior. One of the most widely employed criteria for determining outliers is based on the number of neighboring data points within a fixed distance measure against a fixed threshold. PMU bad data and anomalies can be modeled as outliers in the PMU data as time series data.

The communication channel for transmitting PMU data from the field may drop, delay or corrupt data packets. These lead to different types of anomalies. The dropping of data packets results in missing data (considered in our simulated PMU data). The delay of data packets results with multiple data having same timestamps, which is eliminated by the phasor data concentrator (PDC) during time aligning. The data packets may be corrupted by the communication channel or inaccurate measurements by the acquisition sensors inside the PMU, which results in error in the PMU measurements. Such errors are considered in the simulated PMU data by injecting $\pm$ 20% to 50% error of the true value. In this way, most of the possible anomalies and noise is considered in the simulated PMU data for evaluating the performance of the proposed method.

Base Detectors. A base detector $D(X, \mathcal{K}, \theta)$ is a binary classifier that takes as input a PMU dataset X, a vector of model parameters $\mathcal{K}$ (to characterize the neighborhood and distance measures), and a similarity threshold θ (to distinguish normal data points and outliers), and returns a set of outliers in X with a binary label (*e.g.,* "1" if outlier, "0" otherwise).

Each base detector is supported by an established standalone outlider detection method. To adapt the methods to PMU time series data, a sliding window with (tunable) length N is adopted. The base detectors report the "seen" batch of data points cached in the window at any time, and process the PMU data in batches.

A. Regression-based detector

Linear regression fits a straight line through the set of N points in such a way that the sum of squared residuals of the model is minimized. The model aims to minimize the total vertical distances between the points of the data set and the fitted line (Figure 2, see [19] for more details).

The regression line is modeled as

$$regression = \beta x + \alpha \tag{1}$$

Where x is the data set within a window. Taking the mean value of the distances between y values and the regression line, the deviation is computed as

$$Dev = \frac{1}{N} \sum_{i=1}^{N} abs(x(i) - regression(i)) \tag{2}$$

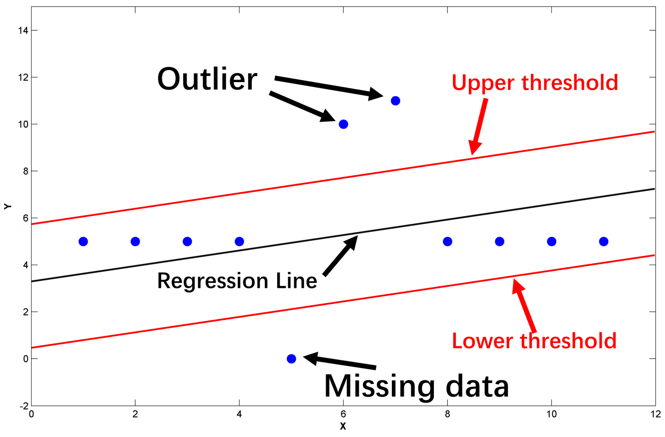


Fig. 2: Linear regression-based outlier detection.

where N is the number of data points. In addition, an upper threshold H and a lower threshold L are derived, where

$$H = \beta x + \alpha + k \times Dev \tag{3}$$

$$L = \beta x - \alpha + k \times Dev \tag{4}$$

Over a stream of PMU data, a regression-based detector D generates a regression line (with β the slope and α the y axis intercept) over the PMU data points X cached in a current window. The parameter k is a preset number that decides the high threshold H and low threshold L. The PMU data points beyond the range constrained by the thresholds are determined as outlier/bad data.

B. Chebyshev-based detector

Chebyshev's theorem determines a lower bound of the percentage of data that exists within k number of standard deviations from the mean. In the case of normal distribution, 95% of the data fall within two standard deviations from the mean. This means that 5% of the data is expected to be outside two standard deviations from the mean.

Chebyshev-based detectors use Chebyshev's inequality to identify outliers as follows (see the details in [12]).

$$P(\mid X - \mu \mid \leq k\sigma) \geq (1 - \frac{1}{k^2}) \tag{5}$$

Where X represents the input PMU data, μ is the mean value of the data within the window, σ is the standard deviation of the data within the window, and k represents the number of standard deviations from the mean. The parameter k is optimized, for PMU outlier detection and the corresponding upper and lower thresholds H and L are given as

$$H = \mu + k * \sigma \tag{6}$$

$$L = \mu - k * \sigma \tag{7}$$

The outliers are then identified as the data points out of the range determined by the thresholds H and L.

C. Density-based clustering

Density-based spatial clustering is applied to detect outliers and missing data with noise. This method is best at finding anomalies among clusters with "irregular" shape, which may not be captured by Linear regression and Chebyshev-based detectors. In particular, we consider DBSCAN [11], a well-adopted model due to its simplicity and effectiveness.

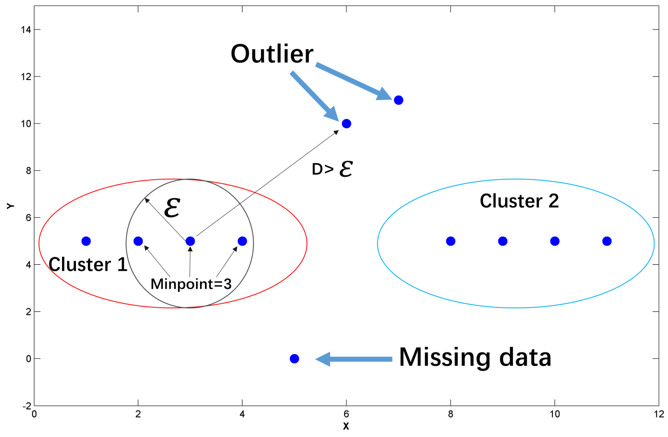


Fig. 3: DBSCAN-based outlier detection.

The DBSCAN algorithm requires two parameters in model parameter set $\mathcal{K}$: ϵ (radius) and Min_point (the minimum number of points within the distance of ϵ) to characterize core points (blue points in Fig. 3). These points have at least Min_point data points in their neighbors within distance ϵ. Two core points are said to be connected if they are in each other's dense neighborhood. The clusters are identified as connected sequences of core nodes. The outliers are then captured as the clusters that contain few node. See [11] for more detail description of DBSCAN algorithm.

Given a PMU data set X, a density-based outlier detector identifies a densely connected area as clusters, and selects the data points that are significantly far away from clusters or contained in clusters of few data points as outliers.

"No single winner." If tested separately over PMU data, each base detector reports precision, recall and false positive very sensitive against different cases of bad data and anomalies as defined later.

As observed in Section V, there is no single method that is capable of detecting most outliers or missing data for various kinds of PMU data. Moreover, (1) it is difficult to set proper parameters for base detectors, because they are so sensitive, and (2) even for well-tuned method, it still cannot perform well for all the cases. As remarked earlier, an ensemble approach is needed to automatically compute a weighted score by assembling the results from each base detector.

D. Ensemble-based outlier detection for PMU data

Given a PMU data set X, and a set of m baseline detectors $D_1, \ldots D_m$, we develop PMU_Ensemble, an ensemble outlier detector that automatically combines the results of the base detectors and predicts outliers in X.

This is a nontrivial problem. Most of the outlier detection models are learned in a supervised manner. Such methods require sufficient amount of training data with correct labels. . However, there is usually little or no labeled data available for high speed PMU data. We next develop unsupervised learning algorithm to train PMU_Ensemble, and develop an online inference algorithm to detect outliers over PMU data stream, in Section III and Section IV, respectively.

III. Ensemble Detector

A. Model Learning

Given a sequence of observed PMU data X= $x_1, \ldots, x_n$ and a set of base detectors $D_1, \ldots D_m$, the learning algorithm, denoted as PMU_Train, has the following three steps.

(1) **Initialization**. PMU_Train first invokes each base detector D_i $(i \in [1, m])$ over X and computes a vector of outlier scores F_i, where $F_i[j]$ denotes the outlier score of data point x_j reported by detector D_i.

(2) **Normalization**. PMU_Train next normalizes the outlier scores to their probabilistic interpretation. For each base detector D_i and the outlier scores F_i, it transforms F_i to a normalized vector F'_i, where each score $F_i[j] \in F_i$ is normalized to a corresponding score $F'_i[j] \in F'_i$, such that $F'_i[j] \in [0, 1]$.

(3) **Model Learning**. PMU_Train then invokes an unsupervised learning algorithm to learn a function, which assigns weight to the outlier scores from each base detector. The function is used in the inference process of PMU_Ensemble to predict the outliers over newly arrived PMU data.

The details of step (2) and (3) is introduced as follows.

Step (2): Normalization. Recall that each detector D_i reports a sequence of outlier scores F_i over PMU data points X. Different base detectors may compute the anomaly scores in various ranges. This makes it difficult to interpret and integrate the outlier scores for the ensemble.

Our goal is to normalize each outlier score vector F_i $(i \in [1, m])$ to a sequence of probability scores F'_i. Each score $F'_i[j]$ (in $[0, 1]$) denotes an estimated probability $P(O \mid F_i[j])$, *i.e.,* the probability that the j-th PMU data point x_j is in an outlier class O rather than from a normal class N, given the outlier score $F_i[j]$. To this end, we adapt a general normalization process in [20] for PMU data. Following [20], the normalization is essentially a parameter learning task that aims to maximize a Bernoulli distribution function.

Consider a score vector F_i and data point $x_j \in X$. By Bayes's theorem, the term $P(O \mid F_i[j])$ can be estimated as

$$\begin{aligned} P(O \mid F_i[j]) &= \frac{p(F_i[j] \mid O)P(O)}{p(F_i[j] \mid O)P(O) + p(F_i[j] \mid N)P(N)} \\ &= \frac{1}{1 + exp(-a_i)} \end{aligned} \tag{8}$$

where

$$a_i = log\frac{p(F_i[j] \mid O)P(O)}{p(F_i[j] \mid N)P(N)} \tag{9}$$

Following [20], we assume that a_i is a discriminant function that classifies x_j to either an outlier or a normal point under the Gaussian distribution with equal covariance matrices. This simplifies a_i as a linear function a_i = $A * F_i[j]$ + B [22]. Therefore,

$$P(O \mid F_i[j]) = \frac{1}{1 + exp(-A * F_i[j] - B)} \tag{10}$$

An Expectation Maximization (EM) algorithm [20] is then invoked to learn parameters A and B. Once the parameters are computed, PMU_Train computes $P(O \mid F_i[j])$ as the normalized score reported by base detector D_i for x_j.

Step (3) Model learning. Once the anomaly scores in step (2) are normalized, the combining strategy for predictors in [23] is used for PMU base detectors, and a Maximum Likelihood Estimation (MLE) algorithm, denoted as MLE-ensemble is used to learn the ensemble detector. The MLE is approximated by an EM algorithm [25] to learn the weights to be linked with each detector. Background knowledge from actual PMU data and physics rules are applied in the model to make it more accurate (and getting a reasonable bias). The details of the MLE-ensemble is presented below.

MLE-ensemble. The MLE-ensemble is constructed on two practical assumptions. (1) All base detectors make independent prediction. (2) Assuming no ground truth for real PMU data, the result obtained by majority voting of the base detectors as the "ground truth" is used to cold start the training process of PMU_Ensemble model.

For each normalized outlier score $F_i[j]$, a binary label f_{ij} is derived, by setting a threshold between $[0, 1]$. We choose threshold 0.95: if $F_i[j] \geq 0.95$, then f_{ij} = 1, which indicates an outlier reported by D_i; otherwise, it indicates a normal data point. The MLE-ensemble learning takes as input the PMU data X of length n, m base detectors (e.g., Chebyshev, linear regression, DBSCAN), and their corresponding outlier vectors with binary labels f_i $(i \in [1, m])$. Over each observed PMU data points $x \in X$, the likelihood of a label y for x, determined by all the base detectors that makes independent predictions, can be written as $\Pi_{i \in [1,n]} Pr(f_i(x)|y)$.

With the assumption that each detector makes independent decision, the likelihood can be represented as the product of the likelihood of f_i as follows [23]:

$$\hat{y}^{MLE} = \mathbf{argmax}_y \Pi_{i \in [1,n]} Pr(f_i(x)|y) \tag{11}$$

which can be further approximated as a weighted sum of the binary values of each f_i with weights determined by sensitivity and specificity of each base detector; that is

$$\hat{y}^{MLE} = sign \sum_{i=1}^{k} (f_i(x) log\alpha_i + log\beta_i) \tag{12}$$

where sign($\cdot$) is the sign function that returns the signs of elements of a numeric vector (class labels that indicate outliers or not). The parameters α_i and β_i are the weights to be trained for PMU_Ensemble, and are defined by the accuracy of base detector D_i as

$$\alpha_i = \frac{\psi_i \eta_i}{(1 - \psi_i)(1 - \eta_i)} \tag{13}$$

$$\beta_i = \frac{\psi_i (1 - \psi_i)}{\eta_i (1 - \eta_i)} \tag{14}$$

Here (a) the *sensitivity* ψ is the fraction of correctly identified outliers, and (b) the *Specificity* η refers to the fraction of correctly identified non-outliers. Importantly, EM algorithm requires an initial guess of the true label in order to update sensitivity and specificity. A common choice is to use the majority rule of the base detectors (Assumption (2) as remarked earlier).

Now PMU_Train just needs to learn α_i and β_i to maximize the likelihood $\hat{y}^{MLE}$. This learning process consists of the following steps. (1) It uses majority voting to make an "initial guess" of ground truth. (2) Estimate ψ and η for each base detectors using the ground truth, and initialize α and β. (3) Apply an EM algorithm to iteratively update α_i and β_i and improve the likelihood estimation until convergence. Finally, it returns PMU_Ensemble as the learned function $\hat{y}^{MLE}$.

We remark that when no training data is available, majority voting is usually considered to "cold start" the learning process [23] and suffice for our test cases. This is based on the intuition that the chance most base detectors make mistakes at the same time is usually small. On the other hand, when there is more labeled data from domain experts, such data can be readily fed to PMU_Ensemble to improve the model accuracy.

B. Inference algorithm

Once PMU_Ensemble is learned with corresponding weights for each base detector, the inference algorithm PMU_Infer simply uses PMU_Ensemble to predict the anomaly score of new PMU data points. Upon receiving a sequence of new PMU data, it (1) invokes the baseline detectors to obtain the outlier score vectors f_i, and (2) directly computes the ensemble outlier score as y^{MLE} with the learned weights for each base detector.

IV. Online Ensembling

The next challange is to extend our model to PMU data streams. A naive method is to cache all PMU data, and learn the MLE ensemble in a "batch" method. Clearly, this method cannot satisfy the need of online outlier detection. Instead, a stream learning algorithm is developed, by pipelining the learning and prediction process in consecutive sliding windows. The algorithm uses a sliding window of length N over the PMU data stream.

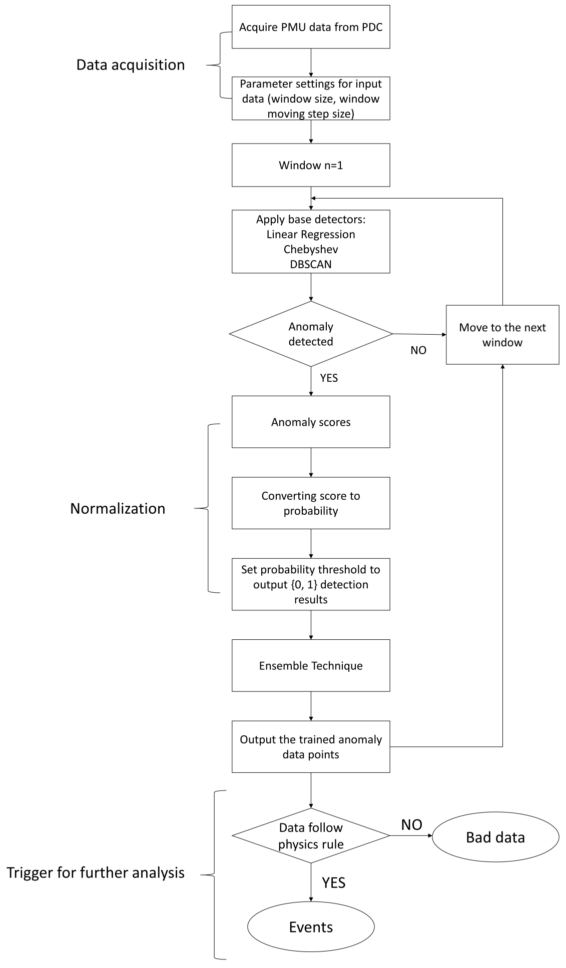
In a nutshell, the algorithm contains two components: an "eager" decision making component, and a "lazy" MLE learner. The eager decision making actively invokes the current PMU_Ensemble model to detect the outliers for new updates in the sliding window; and the lazy learner only does necessary updates to the PMU_Ensemble model, which takes a relatively long training time.

(1) At timestamp 1, the sliding window buffers N data points as a single batch of data points, and invokes the learning algorithm PMU_Train to train the PMU_Ensemble detector.

(2) At timestamp t, PMU_Ensemble trained with data points in the $t-1$th batch (of size L) is invoked to report a set of outliers for the t-th batch of the data points. These outliers are added to the training dataset, and are fed to the lazy learner.

(3) Upon receiving new training data, the lazy learner monitors the aggregated changes of the parameters (*e.g.,* α and β). It remembers the original values of these parameters, and compute new values using majority voting as ground truth. It computes the difference of the old and new values of the parameters, and re-trains PMU_Ensemble only when the changes to the parameters are "significant", determined by a predefined threshold. Typically, the smaller the threshold, the more frequently the model is retrained. For online ensemble, updating PMU_Ensemble too often leads to more training cost, but may achieve higher average accuracy. Therefore, a balance between learning cost and the accuracy of online ensemble should be made. Our experimental study uses a threshold of 15% for *e.g.,* α and β. That is, the lazy learner triggers retraining of PMU_Ensemble only when the values of these parameters increase or decrease by more than 15%.

The stream implementation of PMU_Ensemble makes it adaptive to the possible "drift" of the outlier characterizations and quality change of the different detectors as time goes by, hence remains to be robust over varying PMU data.



Data acquisition
Acquire PMU data from PDC
Parameter settings for input data (window size, window moving step size)
Window n=1
Apply base detectors: Linear Regression Chebyshev DBSCAN
Anomaly detected
NO
Move to the next window
YES
Normalization
Anomaly scores
Converting score to probability
Set probability threshold to output {0, 1} detection results
Ensemble Technique
Output the trained anomaly data points
Trigger for further analysis
Data follow physics rule
NO
Bad data
YES
Events

Fig. 4: MLE_Ensemble based anomaly detection: algorithm

Figure. 4 illustrates the flow chart of online PMU data anomaly detection, using the proposed Ensemble detector.

Pipelining. Pipelining strategy is used to further improve the efficiency of online learning and training, using multiple processors. (1) Each base detector D_i is assigned a processor P_i that computes the outlier scores for the batches of buffered data points received by D_i. Over the data stream, each processor invokes a base detector to detect the outliers in batches, without the need to wait for other processors to complete the detection. (2) Two additional processors are used, one for lazy learning, and the other for eager decision making. The training and inference processes are pipelined, without the need to wait for each other to complete.

V. Evaluation

This section introduces detailed experimental evaluation of our methods, using both simulated PMU data and real-world datasets. Three sets of tests are conducted to evaluate: (1) the accuracy of PMU_Ensemble; (2) the efficiency and learning cost; and (3) a case study to validate the reported outliers for PMU analysis.

A. Experimental settings

Datasets. We use the following datasets.

1.5-hour PMU data with missing data and outliers. A set of 1.5 hours of PMU data generated by Real Time Digital Simulator (RTDS) is used to test the performance of base detectors. The sampling rate is 60 data points per second, and the total number of data points in the test data set is 327023. This work focuses on the outlier and missing data detection instead of an event detection.

Noise is injected that simulates missing data and outliers. A fraction (5%) of the 1.5-hour clean PMU data is randomly selected, and is "polluted" with bad data (missing data or outliers) using a random function that follows Gaussian distribution. The range of changes applied to the cleaned data values is $\pm$ 20% to 50%. For example, if a selected data point has the voltage of 1 p.u, a random function that follows normal distribution updates it to a value in the range of $0.5 - 0.8$, $1.2 - 1.5$, or zero (missing data). Note that although outliers and bad data are simulated, we do not assume that we have the ground truth labels when training PMU_Ensemble over simulated data. As remarked earlier, our model immediately benefits from more validated ground truth (Section III-A).

Real-world PMU data. A 16-minutes real-world PMU data provided by an industry collaborator is used for performance evaluation. The data is verified to contain missing data, outliers and event data by domain experts.

Quality measures. Given a PMU detector D and PMU data X, denote the actual bad data set as B_T, and the bad data/outliers reported by D as B_D, the performance of D is evaluated using three metrics as follows.

Precision. Precision measures the fraction of true bad data in the reported ones from D, defined as

$$Precision = \frac{|B_D \cap B_T|}{|B_D|} \tag{15}$$

TABLE I: Accuracy: w/o event, parameter set 1

	Chebyshev	Regression	DBSCAN	PMU_Ensemble
Parameters	$k = 0.45$	$k = 5$	$\epsilon = 0.05$, Minpts=3	–
Recall	0.8512	0.9254	0.8632	**0.9254**
Precision	0.8851	0.8565	0.8654	**0.8762**
FP	0.1149	0.1435	0.1346	**0.1238**

TABLE II: Accuracy: w/o event, parameter set 2

	Chebyshev	Regression	DBSCAN	PMU_Ensemble
Parameters	$k = 2.2$	$k = 4.5$	$\epsilon = 0.02$, Minpts=4	–
Recall	0.9225	0.9532	0.5632	**0.9652**
Precision	0.7563	0.8021	0.7214	**0.7642**
FP	0.2437	0.1979	0.2786	**0.2358**

Recall. Recall measures the ability of D in finding all outliers. It is defined as

$$Recall = \frac{|B_D \cap B_T|}{|B_T|} \tag{16}$$

False Positive. False positive (FP) evaluates the possibility of false bad data detection; the smaller, the better.

$$FP = 1 - \frac{|B_D \cap B_T|}{|B_D|} \tag{17}$$

Base detectors. We make use of three classes of base detectors: Regression-based, Chebyshev-based, and DBSCAN-based detectors. Each base detector is supported by a standalone method. For a fair comparison, the proper set of parameters for each base detector is computed to achieve reasonable accuracy.

B. Accuracy over simulated PMU data

Base detectors. The performance of PMU_Ensemble and three base detectors over the simulated PMU data is reported in Tables I - IV, measured by precision, recall and false positive. To evaluate the impact of key parameters, we varied the parameters for each base detectors, We fix the threshold that controls the updation of α_i,β_i as 15%, and the threshold that determines data anomaly as 95% for PMU_Ensemble. The window size 72 data points is used, and slide the window over the PMU data in the time intervals of 36 data points. Two types of simulated data are used: (1) The PMU data with injected noise and missing data (setting 1), and (2) The PMU data with noise and missing data, as well as injected event data, which are segments of outliers that encode certain events (setting 2).

Setting 1. The accuracy of the base detectors and the impact of their parameters are reported in Table I and Table II. We tune the parameters of DBSCAN with slightly more "strict" setting which requires denser and smaller neighbors for density-based clustering in Table II compared with Table I; similarly for other base detectors.

Setting 2: This setting uses PMU data with injected missing data, outliers and events (all as outliers). The simulated events in this work include: tap change, line fault and load change. The accuracy is reported in Table III and Table IV, using the

TABLE III: Accuracy: w/ event, parameter set 1

	Chebyshev	Regression	DBSCAN	PMU_Ensemble
Parameters	$k = 0.45$	$k = 5$	$\epsilon = 0.05$, Minpts=3	–
Recall	0.8254	0.8754	0.9125	**0.9250**
Precision	0.8635	0.8146	0.8922	**0.8754**
FP	0.1365	0.1854	0.1078	**0.1246**

TABLE IV: Accuracy: w/ event, parameter set 2

	Chebyshev	Regression	DBSCAN	PMU_Ensemble
Parameter setting	$k = 2.2$	$k = 4.5$	$\epsilon = 0.02$, Minpts=4	–
Recall	0.9142	0.9224	0.9548	**0.9935**
Precision	0.9510	0.9621	0.9721	**0.9962**
FP	0.0490	0.0379	0.0279	**0.0038**

same parameter setting with their counterparts in Setting 1 (Table I and Table II), respectively.

Analysis. The above results conclude following observations. (1) It verifies the "no single winner" observation. (2) The results also verify that a small change in parameter setting will have a large impact on the detection results for PMU data set. For example, the precision of Chebyshev-based detector decreases from 0.8851 to 0.7563 with a slight change of the thresholds. For DBSCAN-based detector, a slightly more "strict" setting that requires denser neighbors (from 3 to 4) within a smaller range (from 0.05 to 0.02) leads to a significant drop of both precision and recall. This is because the condition identifies more "fragmented" smaller clusters, making each a higher chance to be identified as an outlier (low recall), and misses the true outliers due to false positives (low precision). In addition, all the base detectors perform worse over more diversified outliers (with event data included), as expected.

On the other hand, PMU_Ensemble outperforms most (if not all) of the base detectors in precision, recall and false positive. For example, in TABLE I, the recall of PMU_Ensemble is equal to the best base detector 'Regression' in this case without sacrificing too much precision. The precision result is acceptable as it is just marginally lower than 'Chebyshev', which has the highest precision in this case, but with only 0.8512 in Recall. Therefore, the proposed algorithm demonstrates its ability to deliver reliable precision and recall at the same time. In addition, results of PMU_Ensemble are also less sensitive to the insertion of event data, compared with its base detector counterparts. This demonstrates the robustness of PMU_Ensemble against various parameters of the base detectors, and the diversity of outliers.

Performance of PMU_Ensemble. The performance of PMU_Ensemble is reported and evaluate the impact of the ratio and range of the injected noise and bad data to its accuracy. The result is presented in Table V, and compared with the base detectors.

As shown in Table V, (1) with more amount of bad data injected, the accuracy of the detectors decreases, as expected; (2) PMU_Ensemble achieves reasonable accuracy: the recall is 89.54%, even when 5% of the data is selected as bad data, and (3) PMU_Ensemble is constantly better or equal to the best base detector in various cases and proved to be effective in

TABLE V: Comparison of Recall

Ratio/range	Recall	
$1.0\%/[-10\%, 10\%]$	Regression	0.9235
	Chebyshev	0.9015
	DBSCAN	0.8991
	PMU_Ensemble	**0.9355**
$2.5\%/[-5\%, 5\%]$	Regression	0.8575
	Chebyshev	0.8585
	DBSCAN	0.9015
	PMU_Ensemble	**0.9015**
$5.0\%/[-15\%, 15\%]$	Regression	0.8545
	Chebyshev	0.8784
	DBSCAN	0.8954
	PMU_Ensemble	**0.8954**
$1\%[-7.5\%, 7.5\%]$	Regression	0.9213
	Chebyshev	0.9241
	DBSCAN	0.9868
	PMU_Ensemble	**0.9953**

TABLE VI: Processing time per window (in seconds)

Window size (# of data points)	processing time
0.5s (30)	0.0044 s
1.0s (60)	0.0093 s
2.0s (120)	0.0224 s
4.0s (240)	0.0589 s

identifying PMU anomaly data. The result over the precision and false positive are consistent with the result over recall.

C. Processing time

Table VI reports the processing time (seconds per window); including learning and inference computational cost of PMU_Ensemble over the simulated 1.5 hours data, under different window sizes. Hence, it is observed that PMU_Ensemble is quite feasible over PMU data streams. It takes on average 0.0044 second, 0.0093 second, 0.0224 second and 0.0589 second for windows with 30, 60, 120 and 240 data points, respectively. As expected, the larger the window size, the longer PMU_Ensemble takes. Indeed, larger windows contain more data points for each base detectors and the learning algorithm to process.

On the other hand, PMU_Ensemble always completes the processing of a window before the new window arrives for all cases. Moreover, it is observed that the model can be trained once and remains to be accurate for outlier detection for a large batch of following windows before need to be retrained. These verify the effectiveness of PMU_Ensemble over PMU data streams.

D. Case Study using Real-world PMU Data

In this set of tests, the effectiveness of PMU_Ensemble over real-world PMU data is evaluated. As there is no ground truth, advice from domain experts are sought. Three types of anomalous data are recognized, which are as follows. (1) Missing data. As the voltage can not be zero under normal operation, any data point with zero voltage magnitude will be treated as anomaly data. (2) Data during events. All the data points detected during an event will be determined as anomaly data since the voltage magnitude will be largely affected by events. Event data detection can be processed using another algorithm presented in [24]. (3) Any data point which is beyond 5% of the mean value of its previous and following points. This is reasonable for this industry data set because the outlier points are all isolated and not in sequence.

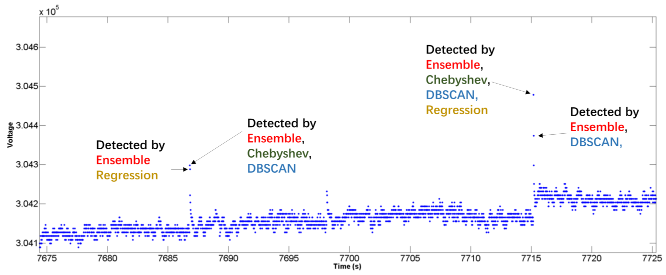


Fig. 5: Real world anomaly PMU data (Voltage Magnitude) detected by PMU_Ensemble

Figure. 5 shows a snapshot of the voltage magnitude of industry PMU data. It is shown that PMU_Ensemble is able to detect data anomalies that cannot be detected by all the base detectors in any cases. In contrast, the base detectors often fail to identify the true outliers and reports considerable amount of false positive cases in our case study. This demonstrates the potential of the PMU_Ensemble in the application of practical PMU analysis in real-world data.

Summary. An ensemble-based outlier detector for PMU data is developed by making use of three different data mining approaches and automatically combining the results with unknown reliability. As verified in our case study, the method is more stable and less sensitive to the change of parameters, and achieves reasonable accuracy. It is observed that it takes a one-time training of the model with affordable time, cost, and the model remains to be effective in predicting newly arrived PMU data. Simulation results using simulated data and known anomalies shows the superiority of the proposed method.

VI. Conclusions

An Ensemble-based anomaly detector for PMU data is developed in this work. Learning and inference algorithm for ensemble based anomaly detector is presented, and online implementation with future parallelization strategies is developed. The ensemble method is observed to be reliable in all the test cases based on the simulated and the real world data. The case study verifies the effectiveness of the ensemble-based anomaly detector, and suggest that the proposed method can be used for different PMU data without having to spend much time on selecting detectors and tuning the corresponding parameters related to different detectors. As the proposed method is data-driven, so it does not depend on the system model or topology. Hence, it can be used as an effective pre-processing tool for PMU data for related centralized or decentralized applications.

The ensemble detector is expected to be more reliable and accurate with more base detectors and PMU data. Additionally, effective online ensemble-based algorithms can be developed that can detect events utilizing PMU data streams.

Acknowledgement

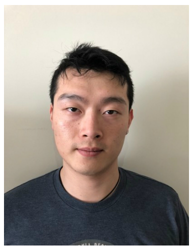
Authors are thankful to the Electric Power Research Institute (EPRI) for providing industrial PMU data.

References

[1] J. De La Ree, et al. ”Synchronized phasor measurement applications in power systems.” IEEE Transactions on Smart Grid, pp. 20-27, January 2010.
[2] R. F. Nuqui and A. G. Phadke. ”Hybrid linear state estimation utilizing synchronized phasor measurements.” IEEE Power Tech, Lausanne, Switzerland, 2007.
[3] M. M. Breunig, H. P. Kriegel, R. T. Ng, J. Sander, LOF: identifying density-based local outliers, in Proceedings of the ACM SIGMOD international conference on Management of data, ACM Dallas, Texas, USA, 2000.
[4] Y. Kou, C.-T. Lu, D. Chen, Spatial Weighted Outlier Detection, In Proceedings of the Sixth SIAM International Conference on Data Mining, pp. 614-618, Bethesda, Maryland, USA, 2006.
[5] A. Lazarevic, V. Kumar, Feature bagging for outlier detection, in Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 157-166, Chicago, Illinois, USA 2005.
[6] K. Martin, Synchrophasor data diagnostics: detection & resolution of data problems for operations and analysis, in Electric Power Group Webinar Series, Jan 2014.
[7] Ghiocel, Scott G., et al. ”Phasor-measurement-based state estimation for synchrophasor data quality improvement and power transfer interface monitoring.” IEEE Transactions on Power Systems 29.2 (2014): 881-888.
[8] Jones, Kevin D., Anamitra Pal, and James S. Thorp. ”Methodology for performing synchrophasor data conditioning and validation.” IEEE Transactions on Power Systems 30.3 (2015): 1121-1130.
[9] Eric McCollum et. al., Correlating Protective Relay Reports for System-Wide, Post-Event Analysis, in Proceedings of 44th Annual Western Protective Relay Conference, October 1719, 2017, in Spokane, WA.
[10] Network Protection & Automation Guide, Alstom Grid, 2011.
[11] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), August 1996, in Portland, Oregon.
[12] B. Amidan, T. A. Ferryman, and S. K. Cooley. ”Data outlier detection using the Chebyshev theorem.” IEEE Aerospace Conference, 2005.
[13] M. Wu, and L. Xie. ”Online identification of bad synchrophasor measurements via spatio-temporal correlations.” Power Systems Computation Conference (PSCC), IEEE, 2016.
[14] NERC, ”Remedial Action Scheme Definition Development”, June 2014, [online] Available: http://www.nerc.com/pa/Stand/Prjct201005_2SpclPrtctnSstmPhs2/FAQ_RAS_Definition_0604_final.pdf.
[15] Wu, Meng, and Le Xie. ”Online Detection of Low-Quality Synchrophasor Measurements: A Data-Driven Approach.” IEEE Transactions on Power Systems 32.4 (2017): 2817-2827.
[16] S. H. Horowitz and A. G. Phadke, ”Third zone revisited,” IEEE Transactions on Power Delivery, 21.1 (2006): 23-29.
[17] Y. Freund and R. E. Schapire A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, in Journal of Computer and System Sciences, Special issue: 26th annual ACM symposium on the theory of computing, 1994
[18] A.Zhukova et. al. ”Ensemble methods of classification for power systems security assessment” Applied Computing and Informatics, in Press.
[19] S. Weisberg. ”Simple linear regression.” Applied Linear Regression, Third Edition (2005): 19-46.
[20] J. Gao, and P.-N. Tan. ”Converting output scores from outlier detection algorithms into probability estimates.” Sixth International Conference on Data Mining (ICDM), IEEE, 2006.

[21] Y. Gu and L. Xie, Fast sensitivity analysis approach to assessing congestion induced wind curtailment, IEEE Transactions on Power Systems, 29.1 (2014): 101110.
[22] Bishop, Christopher M. Neural networks for pattern recognition. Oxford university press, 1995.
[23] F. Parisi, et. al. ”Ranking and combining multiple predictors without labeled data.” Proceedings of the National Academy of Sciences, pp. 1253-1258, 2014.
[24] A. Srivastava, S. Pandey, M. Zhou, P. Banerjee, Y. Wu, ”Ensemble Based Technique for Synchrophasor Data Quality and Analyzing Impact on Applications”. North American Synchrophasor Initiative (NASPI) Work Group meeting, Gaithersberg, MD, USA, 2017.
[25] A. Dempster, N. M. Laird, and D. B. Rubin. ”Maximum likelihood from incomplete data via the EM algorithm.” Journal of the royal statistical society. Series B (methodological), pp. 1-38, 1977.

Mengze Zhou received the B.E. degree in electrical engineering from Chengdu University of Information Technology, Chengdu, China, in 2012, the M.S. degree in electrical engineering from Washington State University, Pullman, WA, USA, in May 2017. He joined the Reliability Standards department, PacifiCorp, Portland, USA, in December 2017. He is now working on data mining and PMU data quality at PacifiCorp.



Yuhui Wang received the B.E degree in software engineering from Xidian University, China, in 2010 and B.S degree in Mathematics from Washington State University in 2015. He is current in 2nd year of his PhD in Computer Science, Washington State University. His research interests are in the areas of machine learning and activities recognition. His current research focuses on activities recognition using mobile sensors.

Anurag K. Srivastava is an associate professor of electric power engineering at Washington State University and the director of the Smart Grid Demonstration and Research Investigation Lab (SGDRIL) within the Energy System Innovation Center (ESIC). He received his Ph.D. degree in electrical engineering from the Illinois Institute of Technology in 2005. His research interests include data-driven algorithm for the power system operation and control. Dr. Srivastava is an editor of the IEEE Transactions on Smart Grid, IEEE Transactions on Power Systems, IET Generation, Transmission and Distribution and Elsevier Sustainable Computing. He is an IEEE distinguished lecturer, and the co-author of more than 250 technical publications.

Yinghui Wu is an assistant professor at School of EECS, Washington State University, US. His current research focuses on Big Data, databases, data mining and network science, with applications in smart grid, information network analytics, and cyber security. He receives his Ph.D. from the School of Informatics, University of Edinburgh in 2011.

P. Banerjee is an Assistant Research Professor at School of EECS, Washington State University, US. His current research mainly focuses on wide-area monitoring and control, real-time simulation and synchrophasor application in the power system. He receives his Ph.D. from Indian Institute of Technology Kanpur in 2015.