

A Novel PMU Fog based Early Anomaly Detection for an Efficient Wide Area PMU Network

Zekun Yang, Ning Chen, Yu Chen, Ning Zhou

Dept. of Electrical and Computing Engineering, Binghamton University, SUNY, Binghamton, NY 13902

Abstract—Based on phasor measurement units (PMUs), a synchrophasor system is widely recognized as a promising smart grid measurement system. It is able to provide high-frequency, high-accuracy phasor measurements sampling for Wide Area Monitoring and Control (WAMC) applications. However, the high sampling frequency of measurement data under strict latency constraints introduces new challenges for real time communication. It would be very helpful if the collected data can be prioritized according to its importance such that the existing quality of service (QoS) mechanisms in the communication networks can be leveraged. To achieve this goal, certain anomaly detection functions should be conducted by the PMUs. Inspired by the recent emerging edge-fog-cloud computing hierarchical architecture, which allows computing tasks to be conducted at the network edge, a novel PMU fog is proposed in this paper. Two anomaly detection approaches, Singular Spectrum Analysis (SSA) and K-Nearest Neighbors (KNN), are evaluated in the PMU fog using the IEEE 16-machine 68-bus system. The simulation experiments based on Riverbed Modeler demonstrate that the proposed PMU fog can effectively reduce the data flow end-to-end (ETE) delay without sacrificing data completeness.

Keywords—Phasor Measurement Unit (PMU), Communication Network, Fog Computing, Anomaly Detection, Smart Grid.

I. INTRODUCTION

Nowadays, the quick growing demand and high dependence on electricity, from government, commercial and household users, are necessitating a high quality power supply. The operators are expected to have a global view of the health condition of the power grid. As a result, there is an increasing interest for wide area monitoring and control (WAMC) in the power grid [7], [8]. The synchrophasor system based on phasor measurement unit (PMU) is considered as the most promising measurement technology to support WAMC applications in the future [7]. The synchrophasor system, also named as the synchronized phasor measurement system, aims to enhance the wide-area monitoring, protection and control system of power grids by providing high accuracy, high sampling rate and synchronized system measurement data.

The sample rate of the widely used Supervisory Control and Data Acquisition (SCADA) system is one sample every 2-4 seconds. In contrast, the data acquisition devices, PMUs, measure and report the instantaneous voltage phasor, current phasor, and frequency with the speed of 30 to 120 observations per second [20]. Meanwhile, the monitoring and assessment applications based on PMUs, including power system fault detection and identification, are designed to exploit real time measurements and have a strict latency requirement [12]. The delay in the communication networks is a critical factor to meet the real-time delivery requirement for the large amount of data.

Considering the volume of monitoring data generated by PMUs, it is non-trivial to provide high quality of service (QoS) in such a big data transfer service. One observation is, however, not all the data is of same importance. When the power grid functions smoothly, longer delays are tolerable. What the operators really need is the immediate delivery of data when anomalies happen. Therefore, the capability of prioritizing the monitoring data will enable us to leverage the QoS features of the communication networks, such that the important data receives high QoS including the guaranteed ceiling of delay, average and peak data rate, limited jitters, etc. Now, it is a chicken-egg problem: the operator needs the data reflecting anomalous behaviors as early as possible, but before being processed we cannot tell which data is the useful one.

Thanks to the fast growth of computing technologies, edge-fog-cloud computing paradigm provides us a key to break the chicken-egg dilemma. Edge computing and fog computing extend the realm of cloud computing by allowing more computing intensive tasks at the edge of the network [23]. In the edge-fog-cloud hierarchical architecture, fog computing is attracting more and more attention in a variety of applications [2]. On one hand, different from remote cloud centers, fog computing consists of heterogeneous smart devices with close proximity to end users. On the other hand, fog layer possesses more computing resources than the embedded Internet of Things (IoTs) devices at the edge. In addition, as tons of monitoring data are processed continuously at the fog, workload placed on communication networks is reduced significantly.

These advantages make fog computing an ideal solution for the delay sensitive synchrophasor system to improve the efficiency of the PMU communication network and provide high quality data to the WAMC applications. The PMU devices, working with sensing and phasor extraction functions at the substation domain, are scalable and capable for extra computing task. Although it is difficult to make sophisticated controlling decisions with local information, PMUs are able to conduct some data pre-processing work on-site, such as data compressing, feature extraction and so on. In this paper, a novel PMU fog architecture is proposed, which leverages the computing resource in PMU devices to detect and mark anomalous data on-site. The marked data will be transmitted with higher priority in the communication network. Such that the critical data can be received by the controller in a timely manner without sacrificing data quality.

The rest of the paper is organized as follows. The Section II introduces the background and related work on synchrophasor system and fog computing paradigm. Section III analyzes the PMU communication network model and provides an efficient solution based on fog computing. Then Section IV illustrates

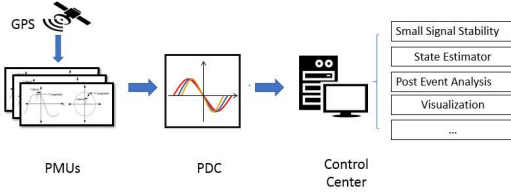


Fig. 1. Synchrophasor system.

two anomaly detection methods fit for edge, including a detailed report of simulating experiments and performance comparison of the two selected methods in subsection IV-C. After the demonstration of the anomaly detection at edge, in Section V a simulation of PMU communication network based Reiverbed Modeler will be built to evaluate the benefit in terms of network efficiency brought by the proposed PMU fog. Finally, Section VI concludes the paper.

II. BACKGROUND AND RELATED WORK

A. Synchrophasor System and Application Requirements

As shown by Fig. 1, a synchrophasor system consists of four types of basic components, the data acquisition device PMU, the phasor data concentrator (PDC), the server to operating WAMC applications and the communication networks [7].

PMUs measures the instantaneous voltage and current at specific location with the sampling rate of 2400 Hz or higher, and then convert them into phasors of 30 Hz or more. Each PMU measurement is time-stamped by the global positioning system (GPS) clock as such a PMU measurement is also called as synchrophasor. The synchrophasors measured by PMU are sent to the remote peer PDC by dedicated communications. The PDC aggregate and time-align synchrophasor data from multiple sources to create a system-wide visibility to the servers for processing in applications software.

The synchrophasor system has brought benefits to the power grid with a large number of actual and potential wide area synchrophasor applications. According to the North American Electric Reliability Corporation (NERC) [20], the use of synchrophasor data can be classified into two general categories, real-time operations applications and planning applications (off-line applications). The former category includes Wide-Area Monitoring and Visualization (WAMV), Oscillation Detection, Frequency Stability Monitoring, Voltage Stability Monitoring, Disturbance Detection and Alarming Studies, State Estimation and so on. The later category includes Post-Event Analysis, Model Validation and so on. The North American SynchroPhasor Initiative (NASPI) provides detailed applications requirements and classification, as shown in Table I. There are several applications that have a strict requirement on the latency while they are not tolerant to the loss of data accuracy. For example, the application of transient stability analysis requires the latency less than 100ms. And the total latency includes the sensor processing time, the data end-to-end (ETE) delay in network, the computing time for decision making, the transmission time for control signal, and the operating time of local device. Therefore, there is a big challenge to keep a low data ETE delay in the network.

B. Fog Computing

As more and more smart mobile devices are ubiquitously deployed, it is natural to utilize the abundant computing power at the edge of network. Compared with cloud computing, the key advantage of fog computing is the on-site processing that eliminates the communication time [25]. The attractive features such as low latency, geo-distribution and real-time interaction make fog computing an ideal platform to deliver instant services in connected sites. A three-level architecture was presented for healthcare infrastructure in which the second level is a Fog-Computing-informed-paradigm [24]. Another suggested application of fog computing is augmented reality and real-time video analysis [26]. Augmented reality applications overlay information on the real world where customers are sensitive to the latency even at the level of tens of milliseconds. In traffic surveillance and target tracking applications, cameras and sensors are intensively deployed in urban areas. Fog computing can efficiently utilize the computing and storage resource to meet the tight latency requirements [1].

A number of research efforts have been reported in literature where the challenges and research directions are identified [5], [15]. As to low-latency processing and resource allocation efficiency, fog computing facilitates mission critical applications at the network edge. A smart urban traffic surveillance system has been implemented with fog computing [1], [3]. It shows that big urban data can be processed in real-time, which is essentially significant in decision making. Fog-to-cloud architectures for smart cities are explored and benefits are evaluated [22]. The utilization of communication and computing resources enables efficient real-time services.

Researchers also explored the potential of fog computing in transmission, storage and processing of big data in a real-time manner in a smart grid monitoring system. A data aggregation approach has been applied on devices, where fog computing enabled routers to improve the throughput for power line communication (PLC) for Smart Meters [19]. Vehicular Delay-Tolerant Networks (VDTN) has been used to prevent data dissemination of various devices by the mobile devices such as vehicles located at the edge of the network [13]. Both of the two approaches depend on the distributed storage capability of the edge devices.

A decentralized voltage stability monitoring method was proposed that is based on a fault-tolerant distributed computing architecture (DCBlocks) [14]. A state estimation scheme was suggested that tried to take advantages of both cloud and fog computing paradigm [17]. A distributed state estimation method was presented that is placed with 5G mobile cellular networks [6]. The three approaches addressed a particular WAMC application at the edge of the network and reduced the communication burden compared with centralized application. However, it is risky to make a decision without global information. Meanwhile, as the edge devices possess limited computing and storage resources, focusing on one specific application may sacrifice the performance on other applications. For example, it may be a trade-off to suffer longer sensor processing time.

A recently proposed hierarchical anomaly detection architecture shares the same insight with us, however, it did not tackle the delay under the umbrella of communication

TABLE I. APPLICATION DATA REQUIREMENT FOR WAMC

Class	Applications based on it	Latency	Sampling Rate	Time Window	Accuracy
State Estimation		1-2 Minutes	1 samples/sec.	5-10 Minutes	Somewhat Important
Transient Stability	Load Trip, Generation Trip, Islanding	100 ms	30 samples/sec.	Few Minutes	Critically Important
Small Signal Stability	Modes, Modes shape, Damping, Online update of PSS, Decreasing tie-line flows	1-5 Seconds	10 samples/sec.	10-15 Minutes	Critically Important
Voltage Stability		Few Seconds	30 samples/sec.	1 hour	Critically Important
Post-Event Analysis	Disturbance Analysis Compliance, Frequency Response Analysis, Model Validation	NA	30 samples/sec.	Few minutes	Critically Important
Visualization	Wide-Area Monitoring and Visualization	1-5 Seconds	1 sample/sec.	Snapshot	Not very important

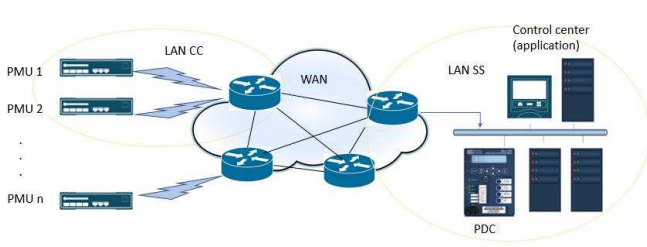


Fig. 2. Synchrophasor system communication network.

network, neither was the advanced edge-fog-cloud computing paradigm leveraged [11]. In contrast, the proposed PMU fog focuses on prioritizing the data such that the existing QoS in communication networks can be leveraged to enable higher QoS for delay sensitive, mission critical applications. The most attractive features of the PMU fog are in two folds: (1) low cost of computational and storage resource of the edge device; and (2) no modification is required on the data packages.

III. PMU FOG ARCHITECTURE FOR EFFICIENT WAMC

A. PMU Fog based PMU Network

Figure 2 presents a typical communication network of the synchrophasor system [4]. PMUs placed in specific substations are connected to the router of the substation via a local area network (LAN), represented by *LAN CC*. There may exist multiple tiers of PDCs considering multiple factors, such as the long distance, data volume, and administration issues. In this model, a simplified architecture is applied, in which it assumes a single layered PDC residing at the control center. Substation routers are able to communicate with the control center router through a wide area network (WAN). In the domain of control center, data transmission depends on another LAN, represented by *LAN SS*, where the PMU measurement data are transmitted to PDC for sorting and synchronization first, and then sent upstream to multiple applications and operations.

Figure 3 shows the WAMC network architecture with the PMU fog proposed in this paper. To reduce the ETE delay without sacrificing data completeness, the PMU fog conducts anomaly detection function at the network edge. Once an anomalous activity is detected, the data packets are assigned a higher priority and then sent into the communication network.

According to the WAMC application requirements shown in Table I, the latency sensitive applications are all transient stability related. It is reasonable to assume that the sensed data related to grid failures are more valuable and should be received by the controller timely. Therefore, the proposed PMU fog assigns the data packets related to grid anomalies a higher priority of transmission. To achieve this goal, the PMUs are

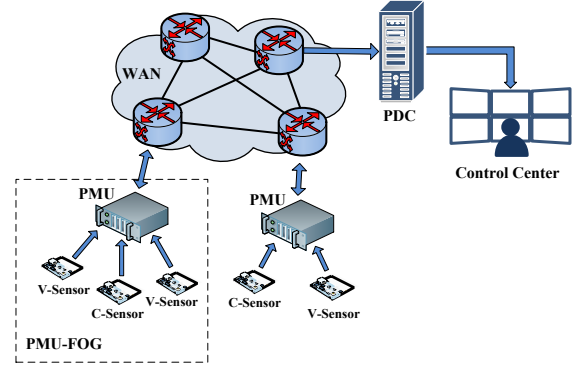


Fig. 3. PMU Fog Architecture.

required to detect the anomalies on site. Once an abnormal situation is detected by a PMU, it marks the current data with a higher QoS level immediately. As long as the network supports the QoS protocols, the marked data will be transmitted with higher priority. Then the PDC at control center receives the data of interest and starts processing with less time delay.

Leveraging the high priority transmission, the data reflecting the abnormal situation of the grid will receive a guaranteed bandwidth and stricter limit on allowed jitters. Consequently, a shorter ETE transmission delay and less fluctuation in latency are achieved, which means a low average T_{WAN} and less opportunity of extremely long delay. Therefore, the timeout T_{TO} can be set to a relatively small value with lower data loss rate. The proposed PMU fog will benefit the power grid by addressing two major factors, the ETE delay and data completeness, which are discussed in the following subsections.

B. End to End (ETE) Delay

The ETE delay counts the time from when a specific packet is created to when it arrives at the destination. The delays incurred by the transducer, phasor generation and abnormal detection will be discussed in the latter section. This section is focused on the analysis of the delay afterward.

For a specific PMU i , ($i = 1, 2, \dots, N$), its ETE transmission delay T_i is,

$$T_i = T_{CCi} + T_{WANi} + 2T_{SSi} + T_{PDC} \quad (1)$$

where T_{CCi} , T_{WANi} and T_{SSi} are the transmission time on *LAN CC*, *WAN*, and *LAN SS*, and T_{PDC} is the data processing time on PDC. It is reasonable to assume that $T_{CCi} \simeq T_{SSi} \ll T_{WANi}$, then:

$$T_i = T_{WANi} + T_{PDC} \quad (2)$$

Therefore the ETE delay largely depends on the transmission time on the WAN and the speed of PDC. T_{WANi} is determined by the physical distance and network condition. More analysis of T_{PDC} is as the following.

Besides functions like data quality check and data storage, the basic function of PDC is data alignment with time-stamps, data synchronization. To align data according to their time stamps, the PDC will gather the PMU measurements with the same time stamp into one set. Each PDC assigns a timeout T_{TO} for each time stamp and this timeout would be the amount of time that the buffer waiting for the PMU measurements with this particular time stamp arriving at PDC. When the set is full or the timeout expires, the PDC will forward the data in the buffer to applications and be ready for the data with next time stamp. The timeout T_{TO} is typically user configurable and set according to the expected worst-case network communications delay. Hence, the PDC processing time T_{PDC} is determined by the timeout T_{TO} and the transmission time in WAN of each PMU T_{WANi} :

$$T_{\theta} = \max(T_{WANi}) \quad i = 1, 2, \dots, N \quad (3)$$

$$T_{PDCi} = \begin{cases} T_{\theta} - T_{WANi}, & \text{if } T_{\theta} < T_{TO}, \\ T_{TO} - T_{WANi}, & \text{if } T_{\theta} \geq T_{TO}. \end{cases} \quad (4)$$

C. Data Incompleteness

Data incompleteness is another critical issue in PMU communication networks. It directly influences the data quality in upper layer applications and further application accuracy. A timeout T_{TO} is assigned to make sure that maximum amount of synchronized data are received. However, when a packet is lost or an extremely long transmission delay occurs, the corresponding measurement cannot arrive PDC in time, and the data incompleteness is inevitable during the data synchronization. Regardless of the packet loss, increasing the T_{TO} can effectively reduce the data incompleteness resulted from long transmission delay. However, longer T_{TO} directly lead to a longer ETE delay. Obviously, it is a trade-off between the shorter ETE delay and the lower data incompleteness rate.

IV. ANOMALY DETECTION BY PMU FOG

As discussed above, the capability of conducting anomaly detection tasks at the edge is the key. In this work two anomaly detection approaches, Singular Spectrum Analysis (SSA) [21] and K-nearest neighbors (KNN) [10], are investigated for the feasibility to be implemented on PMU fog. It is challenging to migrate computing intensive tasks to the fog due to constraints at the network edge, such as limited computing resource and lack of global information.

A. SSA based Anomaly Detection

SSA is a model-free and structure-less approach. It separates the sampled time series into multiple components, and then analyzes a linear recurrence relation (LRR) for interpretable components. By selecting different categories of components as research objects, SSA has been recognized in a wide field of time series processing applications, such as finding data structure, extracting periodic pattern and complex trends, smoothing and change point detection.

The SSA based anomaly detection is actually a quick change point detection method. An important assumption of this method is that, a fault or event always starts from a normal condition. When a fault or an event occurs, the measurement data will experience a sudden change from normal to abnormal. The abnormal detection based on SSA is to catch this particular change.

The basic SSA algorithm can be described as the following. Assume $\mathbb{X} = (x_1, x_2, \dots, x_N)$ is a period of PMU measurement series with the length of N . The original time series \mathbb{X} is truncated by a sliding window, with a fixed window length of M , into a series of lagged vectors \vec{X}_i , as in Eq. 5. These vectors are transformed to a trajectory matrix X , in size of $M \times K$, $K = N - M + 1$, as in Eq. 6. The trajectory includes the whole information of the original time series \mathbb{X} .

$$\vec{X}_i = (x_i, \dots, x_{M+i-1})', i = 1, \dots, K \quad (5)$$

$$X = [\vec{X}_1, \vec{X}_2, \dots, \vec{X}_K] = (x_{i,j})_{i,j=1}^{M,K} \quad (6)$$

Columns X_j of the trajectory matrix X can be considered as vectors in an M -dimensional space \mathbf{R}_M . A particular combination of a certain number l of the Singular Value Decomposition (SVD) eigenvectors determines an l -dimensional subspace \mathcal{L}_l in \mathbf{R}_M , $l < M$. The M -dimensional data X_1, \dots, X_K is then projected onto the subspace \mathcal{L}_l .

Apply SVD procedure on the trajectory matrix X (suppose the rank of X is d) and obtain d collection sets $(\sqrt{\lambda_i}, U_i, V_i)$, $i = 1, 2, \dots, d$, which is called i -th eigentriple of the SVD. The eigentriples include, d non-zero singular values $\sqrt{\lambda_1}, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_d}$ in decreasing order and the corresponding left singular vectors U_1, U_2, \dots, U_d , and right singular vectors V_1, V_2, \dots, V_d . Each eigentriple represents one direction of projection, and each eigentriple $(\sqrt{\lambda_i}, U_i, V_i)$ can rebuild a rank-one matrix $X_i = \sqrt{\lambda_i} U_i V_i'$. The X_i are bi-orthogonal to each other, and compose the basis set of the trajectory matrix X . The eigenvalue $\sqrt{\lambda_i}$ represents the level of contribution of the specific basis matrix X_i to the trajectory matrix, i.e., the level of information contained in matrix X_i .

After decomposing the trajectory matrix, the next step is selecting a subset indices I of basis eigentriples $(\sqrt{\lambda_i}, U_i, V_i)$ or basis matrices X_i to the l -dimensional subspace \mathcal{L}_l in \mathbf{R}_M , where $l < M$.

$$I = \{i_1, i_2, \dots, i_l\} \quad (7)$$

Diagonal averaging is used to transfer matrix $X_I = \sum_{i \in I} X_i$ into a time series \mathbb{X}_I (reconstruction), which is the component-sum of the original series \mathbb{X} .

$$\mathbb{X}_I(i) = \begin{cases} \frac{1}{i} \sum_{j=1}^i x_{j,i-j+1} & \text{for } 1 \leq i < M \\ \frac{1}{M} \sum_{j=1}^M x_{j,i-j+1} & \text{for } M \leq i \leq K \\ \frac{1}{N-i+1} \sum_{j=i-K+1}^{N-K+1} x_{j,i-j+1} & \text{for } K < i \leq N \end{cases} \quad (8)$$

With deliberate selection of I (usually the first l eigenvalues), the reconstruction time series will be reasonably close to

original series \mathbb{X} . The change-point detection is conducted in the following steps. Specify one section of target time series, use SSA to compute its l -dimensional reconstructed matrix X_I , and observe the distance between the l -dimensional matrix X_I and an embedded matrix of the test time series. Then the change point is supposed to be the time point when a significant increasing of distance is observed.

Step 1: Construction of target matrix

Firstly, embed the target time series $\overrightarrow{x_{target}^{(n)}}$ into the target matrix $X_{target}^{(n)}$,

$$\overrightarrow{x_{target}^{(n)}} = (x_{n+1}, x_{n+2}, \dots, x_{n+N}) \quad (9)$$

$$X_{target}^{(n)} = [\overrightarrow{X_1}, \overrightarrow{X_2}, \dots, \overrightarrow{X_n}] \quad (10)$$

$$\overrightarrow{X_i} = (x_{n+i}, \dots, x_{M+n+i-1})', \quad i = 1, \dots, K \quad (11)$$

where $K = N - M + 1$. Note that X is a Hankel matrix, which has the equal elements on its skew-diagonals $i + j = \text{const}$. Apply SVD to target matrix $X_{target}^{(n)}$, and get all of the eigentriples $(\sqrt{\lambda_i}, U_i, V_i)$, $i = 1, 2, \dots, d$. Then, select the first l eigentriples, and build the l -dimensional reconstructive target matrix $X_{target,l}^{(n)}$.

Step 2: Construction of test matrix

$$\overrightarrow{x_{test}^{(n)}} = (x_{n+1+p}, x_{n+2+p}, \dots, x_{n+M+q+1}) \quad (12)$$

$$X_{test}^{(n)} = [\overrightarrow{X_1}, \overrightarrow{X_2}, \dots, \overrightarrow{X_{q-p-1}}] \quad (13)$$

$$\overrightarrow{X_i} = (x_{n+p+i}, \dots, x_{M+n+p+i-1})', \quad i = 1, \dots, q - p + 1. \quad (14)$$

So the size of the test matrix is $M \times (q - p + 1)$.

Step 3: Computing distance

The difference between above two matrices is measured by the sum of the squared Euclidean distance.

$$\mathcal{D}_{n,I,M,p,q} = \sum_{i=p+1}^{M+q+1} ((\overrightarrow{X_{test,i}^{(n)}})' \overrightarrow{X_{test,i}^{(n)}} - (\overrightarrow{X_{test,i}^{(n)}})' U U' \overrightarrow{X_{test,i}^{(n)}}) \quad (15)$$

Note that the distance calculated in this equation describes the condition at time point $t = n + M + q + 1$.

B. KNN based Anomaly Detection

The idea of KNN based abnormal detection is pretty straightforward. Basically, the input is classified by a majority vote of the k nearest training examples in the feature space. First, all training data with classification labels are mapped into the l -dimension feature space, where l is the number of selected features. To classify a specific data point, the Euclidean distance in feature space between the test data point with each training data is calculated. Then the k nearest neighbors are identified as well as the category of the majority. In the end, the test data point is assigned to the category of the majority of its k nearest neighbors.

The feature extraction methodology processes the sampled voltage and/or current signals using windows of a size that is equivalent to one cycle acquired through PMU device. In total 16 features are obtained in both time domain and frequency domain. The time domain features can be divided into three groups. The first group summarizes the window in a global view of the data: harmonic mean (F_1), standard deviation (F_2), mean deviation (F_3), and Kurtosis (F_4).

$$F_1 = \frac{N}{\sum_{j=1}^N \frac{1}{d_j}} \quad (16)$$

$$F_2 = \sqrt{\frac{\sum_{j=1}^N (d_j - \text{mean})^2}{N - 1}} \quad (17)$$

$$F_3 = \frac{\sum_{j=1}^N |d_j - \text{mean}|}{N} \quad (18)$$

$$F_4 = \frac{\frac{1}{N} \sum_{j=1}^N (d_j - \text{mean})^4}{[\frac{1}{N} \sum_{j=1}^N (d_j - \text{mean})^2]^2} \quad (19)$$

The second group of features performs entropy calculations for the aforementioned window. The entropy is originally defined to work with random variables, but it describes the confusion degree of data in the aforementioned window. The features in this category are entropy (F_5), Shanon entropy (F_6) and Renyi entropy (F_7). α is a variable used in calculating Renyi entropy, which is adjusted by a variety tests and the best value is 0.4.

$$F_5 = \sum_{j=1}^N \log(d_j^2) \quad (20)$$

$$F_6 = - \sum_{j=1}^N d_j^2 \times \log(d_j^2) \quad (21)$$

$$F_7 = \frac{1}{1 - \alpha} \times \log \sum_{j=1}^N d_j^\alpha \quad (22)$$

The third group of features describes amplitude of the signal. This group includes root mean square value (RMS, F_8), peak value (F_9) and the difference between the maximum and the minimum amplitudes of the signal (F_{10}).

$$F_8 = \sqrt{\frac{1}{N} \sum_{j=1}^N |d_j|} \quad (23)$$

$$F_9 = \max(d_j) \quad (24)$$

$$F_{10} = \max(d_j) - \min(d_j) \quad (25)$$

The rest of the features are obtained in the frequency domain. The unique features in frequency domain are the harmonic components and obtained by fast Fourier transform (FFT). Note that FFT is already implemented in the hardware in PMU to generate phasors, so the computation cost is low. The FFT is calculated as:

$$D_k = \sum_{j=1}^N d_j e^{-j2\pi k \frac{j}{N}} \quad (26)$$

The feature F_{11} is the total harmonic distortion (THD) of the frequency components D_k :

$$F_{11} = \frac{\sqrt{\sum_{k=2}^N D_k^2}}{D_1} \quad (27)$$

The features F_{12} to F_{16} are the amplitudes of the fundamental and harmonic components of second to fifth orders.

After feature extraction, a classifier determines whether or not the operation condition of the current period is normal. Considering the computing complexity at PMU, KNN is a good fit due to its simplicity.

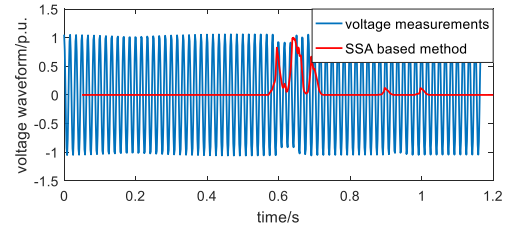
C. Experimental Study and Performance Evaluation

In order to evaluate and compare the detection performance and the computational cost of the SSA based and the KNN based anomaly detection methods, an IEEE 16-machine 68-bus system [9] is built using Power System Toolbox (PST) for a numerical simulation. Five representative fault types are employed, which are generator trip (GT), line-to-line fault (LL), line to ground fault (LG), line-to-line to ground fault (LLG), and three phase fault. The simulation of each fault record lasts 2s with time resolution of 0.0017 (1/600)s. Each fault randomly occurs at 0.3, 0.4, 0.5, 0.6, and 0.7s with duration for 0.01s (6 cycles). The location of each fault is also randomly distributed over the whole system area. The PMU's location is fixed at one of the 68 buses. The voltage measurement record are used for detection.

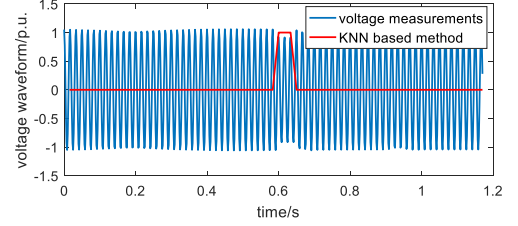
1) *Detection Accuracy*: First of all, the detection accuracy on different fault categories is studied. For each category, 37 records are simulated respectively. The ambient (load) noise level is set to 5%.

For SSA, the selection of parameters is $N = 36, M = 18, p = 18, q = 30$ and $l = 6$. To keep computing cost low, the sampling rate is reduced by five times, i.e., the time resolution of dataset for SSA is 0.0083 (1/120)s. The distance calculated for each time point is normalized and the threshold is selected as the 9×10^6 times of the average distance when the measurement signals describe a normal condition.

For KNN, the number of the interested neighbors is five. The window length is ten, and the time resolution is 0.0017s and the window length is equivalent to one cycle. Since KNN classification algorithm is an instance based learning algorithm, five fault records for each type of fault, i.e., 595 instances in feature space in total are randomly selected for training. And the rest of 32 fault records are utilized as testing data. Four features, Shanon entropy (F_6), RMS (F_8), peak value (F_9)

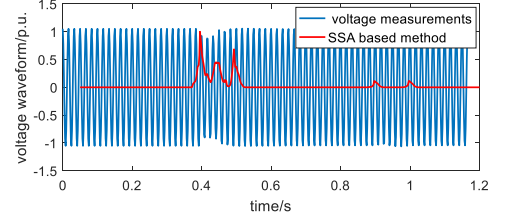


(a) a case of GT fault using SSA based abnormal detection

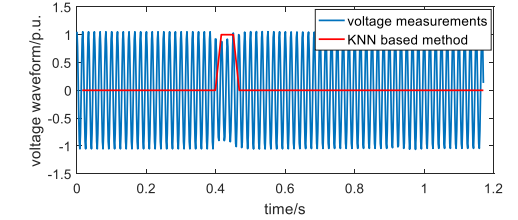


(b) a case of GT fault using KNN based abnormal detection

Fig. 4. A case of GT fault.



(a) a case of LL fault using SSA based abnormal detection



(b) a case of LL fault using KNN based abnormal detection

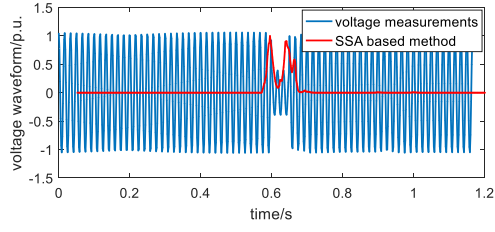
Fig. 5. A case of LL fault.

and the difference between the maximum and the minimum amplitudes of the signal (F_{10}), are selected as inputs to pursue a balance of computational cost and accuracy.

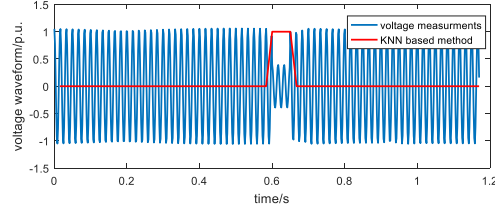
Figures 4, 5, 6, 7, 8 provide examples of anomaly detection for each type of faults respectively, where we can see the detail of detection result. Both the SSA and the KNN successfully identified the fault. While the SSA method is more sensitive to the marginal fluctuation of the voltage measurement, the detection of the KNN method always has a delay about one cycle, which is incurred by the window length.

Tables II and III present the performance on different fault categories by the SSA method and KNN method respectively. The SSA method, although has higher false positive rate (FPR) due to its sensitivity to small changes, generally outperforms the KNN method.

2) *Robustness on Different Noise Levels*: The performances of the SSA method and KNN method are also compared against noise levels. The testing ambient noise levels are set as 3%, 5% and 7%, which are close to the real situation in power systems. For each noise level, 140 fault records including all

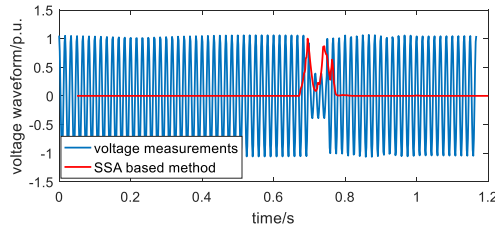


(a) a case of LG fault using SSA based abnormal detection

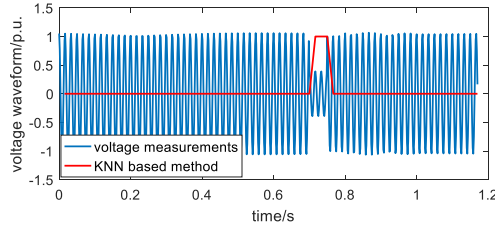


(b) a case of LG fault using KNN based abnormal detection

Fig. 6. A case of LG fault.

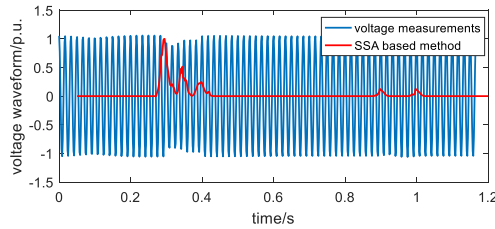


(a) a case of LLG fault using SSA based abnormal detection

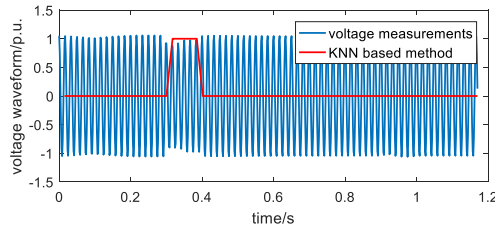


(b) a case of LLG fault using KNN based abnormal detection

Fig. 7. A case of LLG fault.



(a) a case of 3 phase fault using SSA based abnormal detection



(b) a case of 3 phase fault using KNN based abnormal detection

Fig. 8. A case of three-phase fault.

TABLE II. DETECTION ACCURACY OF THE SSA METHOD WITH 5% NOISE

	True positive rate	False positive rate
LL	100%	0
LG	100%	0
LLG	100%	3.13%
3Phase	100%	6.25%
GT	100%	18.75%

TABLE III. DETECTION ACCURACY OF THE KNN METHOD WITH 5% NOISE

	True positive rate	False positive rate
LL	90.63%	3.13%
LG	90.63%	0
LLG	93.75%	0
3Phase	87.50%	0
GT	78.50%	3.13%

five fault categories are randomly generated. The simulation of each fault record lasts 2 seconds and each fault randomly occurs at 0.3, 0.4, 0.5, 0.6, and 0.7 seconds with duration for 0.01s (6 cycles).

Figure 9 shows the Receiver Operating Characteristic (ROC) curves of SSA with various threshold settings. It is not a surprise that the lower the noise level is, the better the performance is. Consider each curve of different noise levels individually, the optimal detection result with noise levels of 3%, 5% and 7% are with TPR & FPR of 100% & 4.29%, 94.29% & 5.00% and 100% & 12.86%, respectively, as marked in Fig. 9.

KNN is an instance-based learning method, so it is not possible to plot a ROC curve with continuously threshold changing. Alternatively, TPR and FPR with different number of training data sets are used. Figure 10 shows the detection result with training data set number from one to seven, where the training data sets are delicately selected. Since the training data set size directly determines the method complexity, the trade-off of the detection performance and the computational cost is also investigated. With five selected data sets, the TPR & FPR for noise level 3%, 5% and 7% are 94.29% & 2.86%, 94.29% & 2.86% and 92.86% & 4.29%.

As KNN is instance-based, the performance largely depends on the quality of training data. Figure 11 gives an example of a random training data set comparing with the selected one with noise level 5%. And it is obvious that the

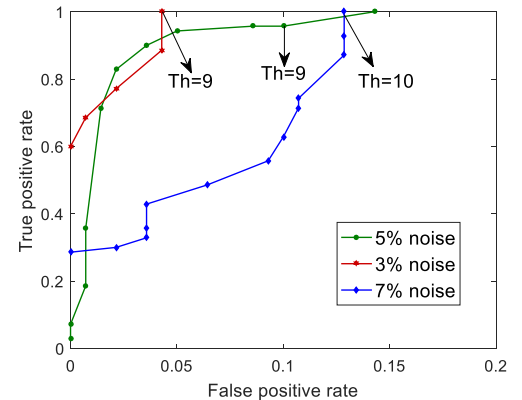


Fig. 9. Detection performance of SSA on different noise levels.

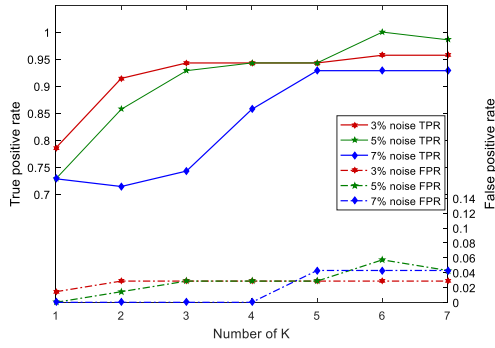


Fig. 10. Detection performance of KNN on different noise levels.

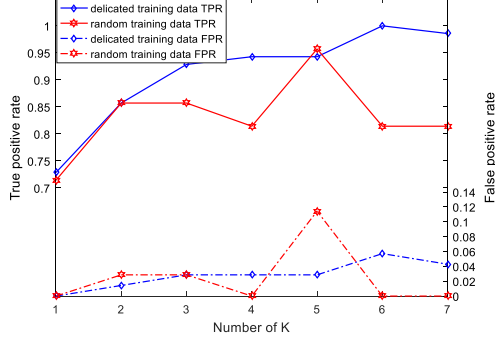


Fig. 11. Comparison of different training data sets.

performance of KNN is not stable with randomly selected training data set.

3) *Computational Complexity Analysis*: To verify the feasibility of our proposed PMU fog scheme in wide area smart grid anomaly detection, the algorithms are implemented and tested on two different computing devices. One is a laptop with an Intel i7-7700HQ processor, 2.80 GHz processing frequency and 8 GB RAM memory. The averaged running time of KNN approach is 0.0123s and the averaged running time of SSA algorithm is 0.0152s. Another device is Raspberry Pi 3 Model B. Its processor is 64/32-bit quad-core ARM Cortex-A53 and it has a processing frequency of 1.2 GHz. The memory is 1 GB. The KNN algorithm runs an averaged time 0.1512s on it while the averaged running time of SSA is 0.1896s.

4) *Discussions*: In general, both of the two anomaly detection methods can serve the purpose at the PMU device and are able to detect anomalies accurately. With their own advantages and constraints, they can be applied alternatively in practice according to actual working environments. More specifically, the advantages of the SSA method include higher detection accuracy, shorter detection delay, robust to different noise levels and different fault types and no need of historian data for training. In contrast, the KNN method is a better choice when low computing cost is a dominant factor, or additional information of feature extraction for other applications.

V. IMPROVEMENT IN NETWORK EFFICIENCY

A simulation model is built based on Riverbed Modeler [18] to evaluate the effects of the proposed PMU fog to the entire PMU communication network efficiency. Riverbed Modeler (used to name as OPNET) is a robust software specialized at network and system simulation, which provides a large amount of models of standard communication devices

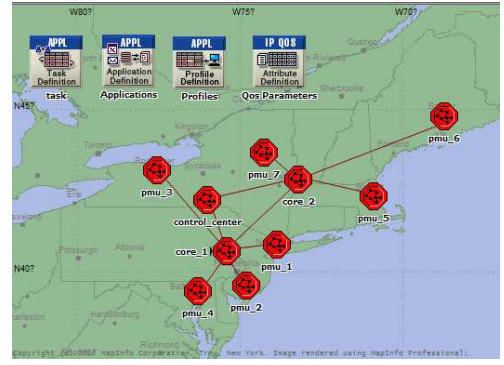


Fig. 12. PMU communication network on Riverbed.

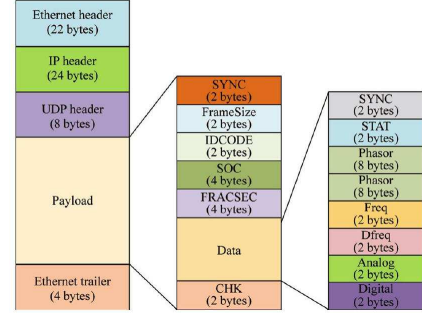


Fig. 13. Data packet format for IEEE C37.118.2 protocol.

and protocols. The version of software we used in this paper is Riverbed Modeler Academic Edition 17.5.

A. PMU communication Model in Riverbed

The communication model implemented on Riverbed is based a hypothetic locations of seven PMUs in the northeast of US. Note that, this synchronphasor system model including seven PMUs does not represent the actual synchronphasor system working in northeast, but is a model in equivalent geographic distance level and network architecture. Since in a wide area network, the distance between network devices is a crucial factor for the distance-based delay, the placements of the network components are important.

Figure 12 is a global view of the geographical topology of the communication network model. The octagons on the topological graph stand for subnets. The subnets PMU_i represent LANs at substations where PMU devices implemented. In each PMU_i subnet, PMU device is connected to the outer network via the substation router. The PMU device is simulated by *Ethernet Workstation* model and it will generate data traffic to its destination PDC. The subnets $core_1$ and the subnet $core_2$ represent a network of meshed routers. In the subnets $control_center$, PDC and WAMC are configured by *Ethernet Server* model. They connect with each other through a switch and connect to the wide area network with a router.

The data communication requirements of synchronphasor system are well addressed by IEEE C37.118 standard [16]. The IEEE C37.118.2 defines the data packet formats for four types of messages, data, configuration, header and command. In this paper we mainly focus on the data packets, which correspond to the PMU measurements. Figure 13 shows the data packet structure and the total size of a data packet is 112 Bytes.

The data format supports efficient and secure message exchange over the network. A paradigm of network protocols used in various layer is given in Table IV. PMU devices in substation are constantly sending data packet to the PDC, so Constant Bit Rate (CBR) is appropriate as an application layer protocol. User Datagram Protocol (UDP) is selected as the protocol in transportation level instead of Transportation Control protocol (TCP) due to consideration of latency. Also given the latency and bandwidth requirements, the optical fibers is a popular physical media in power grid communication networks. In the simulation, there is no existing

TABLE IV. A PARADIGM FOR COMMUNICATION MODEL(C37.118)

Layer	Protocol
Application	CBR
Transportation	UDP
Network	IP
Data link	Ethernet
Physical	Optical fiber

application in accordance with the IEEE C37.118 standard. We used *Task Config* model to customize an task generating data packets obeying the IEEE C37.118 standard. And then the *Application Config* model employs this task to configure an application with the transport protocol of UDP. To mimic a PMU device, the *Ethernet Workstation* will generates data message constantly with the speed of 30 packets/s and the size of 112 Bytes.

In order to transmit the phasor measurements of interest with high priority, a packet scheduling mechanism named Weighted Fair Queuing (WFQ) is used in Riverbed to support the QoS guaranteed application. Basically, WFQ assigns the data flow with high QoS requirement more bandwidth sharing than others. Therefore, application class, weight and buffer allocation are important for WFQ mechanism. In the simulation, we made WFQ classified by DiffServ. The data flow is sorted by the corresponding DiffServ Code Point (DSCP) at the data source, and assigned by specific weight and buffer size. In our case, the DSCP for all data flow is set to AF (Assured Forwarding) 23 for the scenario without QoS control, which means all of the phasor measurement data will be transmitted fairly. While in the scenario with QoS control, the phasor measurement data of interests are transmitted utilizing an application with the DSCP of EF (Expedited Forwarding), and the phasor measurement data of normal situation still use the service of AF 23. Note that EF class is the highest class with lower delay and loss compared with AF 23.

In the simulation, three scenarios varying in the configuration of communication links are designed. Two of them belong to dedicated network paradigm, where the network is assigned to the utilization of PMU communication only and each PMU device has a dedicated channel connecting to the PDC. These two scenarios are different in the bandwidth assignment. *Scenario 1* builds network with 10 Mbps communication links, while *Scenario 2* builds network with 100 Mbps communication link. And *Scenario 3* simulates the situation that PMU system shares network with traffic from other power grid devices like RTUs and IEDs. In this scenario, Ethernet 100BaseT serves as the communication link but a background traffic *T3_1hours_bps* is added to every links from substation router to WAMC control center in the network.

This background traffic constantly generates data in 45 Mbps, therefore, the background traffic level is 45%.

B. Simulations Result and Discussions

In the simulation, the ETE transmission delay statistics are collected. The ETE transmission delay includes (1) the delay between all PMU devices and the PDC, and (2) the delay between the PDC to the server where the WAMC applications are executed. The later is much smaller value than the previous one, which is in accordance with the model analysis. For each PMU device, the ETE delay is calculated by the average value of the delay from itself to the PDC adding the average value of the the delay from the PDC to the application server, and the later one is uniform for all of the PMUs. Tables V, VI, and VII show the simulation result of three scenarios. Table V represents the scenario of a dedicated network with 10 Mbps communication links (*Scenario 1*), Table VI stands for the scenario of a dedicated network with 100 Mbps communication links (*Scenario 2*). In Table VII, 45% background traffic is added to the 100 Mbps communication links (*Scenario 3*). Generally speaking, the delay is distance-dominated. The longer distance between the PMU to the PDC, the longer ETE delay. Meanwhile, the network with 10 Mbps links suffers a higher average delay than the network with 100 Mbps because of the bandwidth limitation. While the sharing communication in the third scenario does not lose too much on delay, which may be because the bandwidth of 100 Mbps is large enough for the PMU traffic and the performance won't sacrifice too much for sharing channel.

TABLE V. ETE TRANSMISSION DELAY WITH 10MBPS DEDICATED NETWORK (MS)

	No QoS control	With QoS control
PMU 1	4.7	3.1
PMU 2	4.8	3.0
PMU 3	5.1	3.9
PMU 4	5.5	3.7
PMU 5	8.9	4.7
PMU 6	11.6	5.6
PMU 7	10.5	4.6

TABLE VI. ETE TRANSMISSION DELAY WITH 100MBPS DEDICATED NETWORK (MS)

	No QoS control	With QoS control
PMU 1	5.4	2.7
PMU 2	5.2	2.6
PMU 3	6.6	5.0
PMU 4	3.8	2.8
PMU 5	4.7	3.4
PMU 6	5.8	4.7
PMU 7	5.1	3.6

TABLE VII. ETE TRANSMISSION DELAY WITH 100MBPS BACKGROUND COMMUNICATION (MS)

	No QoS control	With QoS control
PMU 1	4.7	2.6
PMU 2	4.1	2.8
PMU 3	6.5	4.9
PMU 4	5.2	2.8
PMU 5	4.1	3.1
PMU 6	5.9	4.7
PMU 7	4.7	3.6

In all scenarios, the PMU ETE transmission delays without a QoS control are all higher than that with a QoS control,

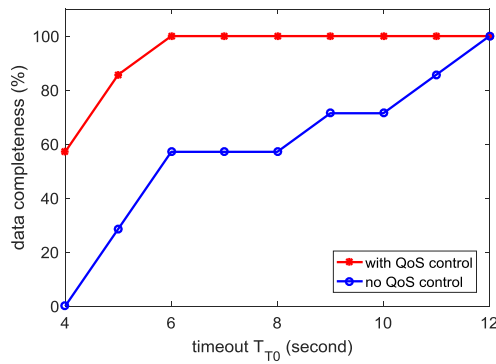


Fig. 14. Data completeness with 10 Mbps dedicated network.

especially in the bandwidth limited situation (*Scenario 1*). A significant reduction of ETE transmission delay is achieved if the routers know which data packets are more important than others and apply QoS control policies. Therefore, if the electrical abnormal situation is successfully detected by PMU fog, the data of interests can be transmitted with higher priority, which means a shorter transmission delay. The data completeness is also evaluated. Take Scenario 1 as an example, Fig. 14 shows the data completeness rate under different timeout T_{T0} values. Given a specific timeout T_{T0} value, the data completeness rate of a scenario with QoS control outperforms the scenario without QoS control. In another word, the data marking mechanism at the PMU fog can effectively take more advantage of the QoS control mechanism in the communication network. Such that a lower T_{T0} can be applied, which further reduces the ETE delay with a guaranteed data completeness rate.

VI. CONCLUSIONS

The efficiency of the PMU communication network is important for latency strict WAMC applications. In this paper, a prioritized data transmission mechanism for PMU measurement data flow is proposed. Inspired by the edge-fog-cloud computing paradigm, the measurement data of fault or disturbance is identified at the PMUs and marked with higher priority, such that the QoS mechanism existing in the communication networks can be leveraged to selectively accelerate the data delivery. For this purpose, two anomaly detection methods are studied at the PMU fog layer, SSA and KNN. Experimental results prove that both methods are fit at the fog layer with different advantages and constraints. The simulation of the communication network shows the prioritized transmission mechanism can effectively reduce the data flow ETE delay without sacrificing data completeness.

REFERENCES

- [1] N. Chen, Y. Chen, E. Blasch, H. Ling, Y. You, and X. Ye, "Enabling smart urban surveillance at the edge," in *2017 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE, 2017, pp. 109–119.
- [2] N. Chen, Y. Chen, X. Ye, H. Ling, S. Song, and C.-T. Huang, "Smart city surveillance in fog computing," in *Advances in Mobile Cloud Computing and Big Data in the 5G Era*. Springer, 2017, pp. 203–226.
- [3] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang, and R. Zimmermann, "Dynamic urban surveillance video stream processing using fog computing," in *Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on*. IEEE, 2016, pp. 105–112.

- [4] M. Chenine and L. Nordstrom, "Modeling and simulation of wide-area communication for centralized pmu-based applications," *IEEE Transactions on Power Delivery*, vol. 26, no. 3, pp. 1372–1380, 2011.
- [5] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.
- [6] M. Cosovic, A. Tsitsimelis, D. Vukobratovic, J. Matamoros, and C. Anton-Haro, "5g mobile cellular networks: Enabling distributed state estimation for smart grid," *arXiv preprint arXiv:1703.00178*, 2017.
- [7] U. D. E. Delivery, "Synchrophasor technologies and their deployment in the recovery act smart grid programs," 2013.
- [8] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid: the new and improved power grid: A survey," *IEEE communications surveys & tutorials*, vol. 14, no. 4, pp. 944–980, 2012.
- [9] R. Graham, "Power system oscillations," 2000.
- [10] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.
- [11] M. Jamei, A. Scaglione, C. Roberts, E. Stewart, S. Peisert, C. McParland, and A. McEachern, "Anomaly detection using optimally-placed μ pmu sensors in distribution grids," *IEEE Transactions on Power Systems*, 2017.
- [12] P. Kansal and A. Bose, "Bandwidth and latency requirements for smart transmission grid applications," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1344–1352, 2012.
- [13] N. Kumar, S. Zeadally, and J. J. Rodrigues, "Vehicular delay-tolerant networks for smart grid data management using mobile edge computing," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 60–66, 2016.
- [14] H. Lee, S. Niddodi, A. Srivastava, and D. Bakken, "Decentralized voltage stability monitoring and control in the smart grid using distributed computing architecture," in *Industry Applications Society Annual Meeting, 2016 IEEE*. IEEE, 2016, pp. 1–9.
- [15] Y. Liu, J. E. Fieldsend, and G. Min, "A framework of fog computing: Architecture, challenges and optimization," *IEEE Access*, 2017.
- [16] K. Martin, G. Brunello, M. Adamiak, G. Antonova, M. Begovic, G. Benmouyal, P. Bui, H. Falk, V. Gharpure, A. Goldstein *et al.*, "An overview of the iec standard c37. 118.2 synchrophasor data transfer for power systems," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1980–1984, 2014.
- [17] A. Meloni, P. Pegoraro, L. Atzori, A. Benigni, and S. Sulis, "Cloud-based iot solution for state estimation in smart grids: exploiting virtualization and edge-intelligence technologies," *Computer Networks*, 2017.
- [18] R. Modeler, "Riverbed," 2014.
- [19] M. S. H. Nazmudeen, A. T. Wan, and S. M. Buhari, "Improved throughput for power line communication (plc) for smart meters using fog computing based data aggregation approach," in *Smart Cities Conference (ISC2), 2016 IEEE International*. IEEE, 2016, pp. 1–4.
- [20] M. Patel, S. Aivaliotis, E. Ellen *et al.*, "Real-time application of synchrophasors for improving reliability," *NERC Report, Oct*, vol. 1, 2010.
- [21] A. S. Polunchenko and A. G. Tartakovsky, "State-of-the-art in sequential change-point detection," *Methodology and computing in applied probability*, vol. 14, no. 3, pp. 649–684, 2012.
- [22] W. Ramirez, X. Masip-Bruin, E. Marin-Tordera, V. Souza, A. Jukan, G. Ren, and O. G. de Dios, "Evaluating the benefits of combined and continuous fog-to-cloud architectures," *Computer Communications*, vol. 113, pp. 43–52, 2017.
- [23] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [24] V. Stantchev, A. Barnawi, S. Ghulam, J. Schubert, and G. Tamm, "Smart items, fog and cloud computing as enablers of servitization in healthcare," *Sensors & Transducers (1726-5479)*, vol. 185, no. 2, 2015.
- [25] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, 2014, pp. 1–8.
- [26] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*. ACM, 2015, pp. 37–42.