Online Power System Event Detection via Bidirectional Generative Adversarial Networks

Yuanbin Cheng, *Student Member, IEEE,* Nanpeng Yu, *Senior Member, IEEE,* Brandon Foggo, *Member, IEEE,* Koji Yamashita, *Member, IEEE*

***Abstract*—Accurate and speedy detection of power system events is critical to enhancing the reliability and resiliency of power systems. Although supervised deep learning algorithms show great promise in identifying power system events, they require a large volume of high-quality event labels for training. This paper develops a bidirectional anomaly generative adversarial network (GAN)-based algorithm to detect power system events using streaming PMU data, which does not rely on a huge amount of event labels. By introducing conditional entropy constraint in the objective function of GAN and graph signal processing-based PMU sorting technique, our proposed algorithm significantly outperforms state-of-the-art event detection algorithms in terms of accuracy. To facilitate the adoption of the proposed algorithm, a prototype online platform is also developed using Apache Hadoop, Kafka, and Spark to enable real-time event detection. The accuracy and computational efficiency of the proposed algorithm are validated using a large-scale real-world PMU dataset from the Eastern Interconnection of the United States.**

***Index Terms*—Phasor measurement unit, event detection, generative adversarial networks.**

I. Introduction

PHASOR Measurement Units (PMU) are high-frequency sensors that measure voltage and current phasors with high accuracy. They are widely deployed in power transmission systems, providing useful real-time information for system monitoring [1]. The number of deployed PMUs has drastically increased in the past decade around the world [2]. The increase in PMU adoption has led to a significant increase in the amount of streaming data. With either 30 Hz or 60 Hz sampling rate, the PMUs in North America generate terabytes of data every month. This massive amount of PMU data not only brings new challenges [3], [4] but also enables the development of online data-driven algorithms to monitor and control power transmission networks [5].

Online event detection in the power transmission system is crucial to ensuring the security and reliability of the bulk power system. Prompt and accurate detection of abnormal power system events can facilitate system operators to take appropriate corrective actions. This paper aims to develop an online power system event detection algorithm that is both highly accurate and computationally efficient with a large-scale real-world streaming PMU dataset.

The existing literature for data-driven event detection in power systems can divide into three categories. The first category is based on signal processing techniques, such as wavelet transforms [6], [7] and Fourier transforms [8]. The papers in this category apply signal processing techniques on data from a single measurement channel of a single PMU. Most methods in this category do not fully exploit the spatial correlations between data from different PMUs or the physical relationships between different measurement channels. Thus, they do not yield state-of-the-art event detection accuracy on large-scale real-wold PMU dataset, which always has strong spatio-temporal correlations.

The second category of literature exploits the statistic properties of the PMU dataset to detect events. If the distance between pre-fault spatio-temporal correlation coefficient matrix and that of the current time stamp is larger than a threshold, then abnormal event is detected [9]. A hypothesis testing framework is establish to detect power system events, where the null hypothesis is that the sample covariance of the PMU data collected during normal system operations is the same as that of the current data [10]. The low-rank property of the PMU data matrices during normal system operation periods is identified and leveraged to detect power system events. The line outage detection is formulated as a sparse signal reconstruction problem and solved by iterative algorithms [11]. The algorithms in this categories have a few limitations. First, the time complexity of correlation matrix calculation [9], [10] and iterative singular value decomposition [12] are high, which limits their scalability. Second, the key statistics derived from the streaming PMU data are sensitive to bad data, which is prevalent in real-world operating scenarios. Third, the reliance on complete network topology and simplified power system model limits its applicability [11].

The third category of papers use machine learning and data mining techniques to detect power system events. A vector autoregressive model is fitted to predict PMU measurements [13]. If the prediction error exceeds a predefined threshold, then an event is detected. To achieve high accuracy, the prediction model needs to be retrained frequently, which incurs high computation cost for real-time applications. A data mining technique called matrix profile is adopted to discover new events that have similar electrical signatures as the previous ones [14]. The algorithm may not work as well when the new event has a very different motif than the existing ones. An ensemble learning algorithm, which combines isolation forest, K-means, and LoOP is developed to detect and classify anomalous PMU data [15]. A convolutional neural network (CNN)-based model is developed to detect frequency disturbance events by analyzing images created by frequency and relative angle shift data [16]. A two-level CNN-based

Y. Cheng, B. Foggo, N. Yu and K. Yamashita are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA. Corresponding Author: Nanpeng Yu. Email: nyu@ece.ucr.edu

regression model is used to perform online transient stability prediction by examining images generated from voltage, rotor angle, frequency deviation and complex power data [17]. The decision tree model is adopted to classify disturbance events recorded by PMUs, where the input features are derived from the multidimensional minimum volume enclosing ellipsoid of the measurements [18]. Autoencoder, deep belief networks, and CNN are trained on frequency measurements from PMUs to classify power system events [19]. An information loaded deep neural network is developed to classify power system events using a large-scale real-world PMU dataset from Eastern Interconnection of the U.S. [20].

Although deep neural network-based event detection and classification algorithms yield higher accuracy and computation efficiency, they often require thousands of confirmed events as training labels [16], [18]–[20]. The accuracy of event detection and classification drops quickly as the number of training label reduces. In practice, it is not only time-consuming but also labor-intensive to collect high-quality power system event labels.

To fill the knowledge gap in the field, we develop a Generative Adversarial Network (GAN)-based power system event detection algorithm using streaming PMU data, which does not depend on the availability of a large amount of high quality event labels. Unlike the other deep learning-based models, only a small number of labeled events is needed to tune two hyper-parameters in our proposed GAN-based event detection model. We name our proposed model bidirectional anomaly GAN (Bi-AnoGAN). We select GAN as a basis because it is a powerful generative model, which can accurately capture the complex patterns in high dimensional datasets [21]. The basic GAN models have been modified to successfully detect anomalies in images [22]–[24] and water treatment and distribution datasets [25]. When applying GAN-based algorithms to detect abnormal power system events, we first try to learn two mapping functions that project PMU data samples during normal operating conditions to the noise space and then back to the data space. If there is a large difference between an incoming PMU data sample and its reconstructed version, then it is very likely that the new sample corresponds to a system event. To improve the computation efficiency and develop an online event detection algorithm, we adopt the design of Bidirectional GAN (BiGAN) [26] by training an additional encoder network that can directly map a data sample to the noise space.

It is very challenging to directly apply the state-of-the-art GAN-based algorithm to detect power system events using streaming PMU data. This paper proposes three technical advancements to improve GAN model convergence, power system event detection accuracy, and computation efficiency.

First, the training of GANs often lead to divergence and unstable models. Although approaches such as Wasserstein GANs (WGAN) [27] and WGANs with gradient penalties (WGAN-GP) [28] do improve the numerical stability in the GAN training process, they cannot guarantee training stability improvements for all types of large-scale datasets [29]. We propose adding two conditional entropy constraint terms into the objective function of the GAN model to encourage cycle-consistency. The newly added term ensures that feeding the raw PMU data into an encoder and then a decoder will produce the same original PMU data. The ablation study results will show that the conditional entropy constraint improves not only training stability but also event detection accuracy.

Second, the convolution layers of GANs [30] used to process PMU data work best when the local patches of data are highly correlated. To improve the learning of convolution filter weights, this paper adopts a graph signal processing (GSP)-based sorting algorithm to systematically arrange the PMUs in the input layer so that PMUs with higher correlations are placed closer to each other. The ablation study results show that this technique improves the event detection accuracy.

Third, to enable the proposed GAN-based event detection algorithm to handle large-scale PMU data in real-time operations, we develop a prototype streaming platform consists of GPU clusters equipped with Apache Hadoop, Kafka, and Spark. This prototype system has extremely high computation efficiency and can finish event detection for a single snapshot of PMU data within 2 milliseconds.

The main contributions of this paper are listed as follows:

- This paper develops a GAN-based online power system event detection algorithm using streaming PMU data, which does not rely on a massive amount of event labels.
- By introducing conditional entropy constraint and GSP-based PMU sorting approach, our proposed Bi-AnoGAN algorithm significantly outperforms state-of-the-art event detection algorithms in terms of accuracy on a large-scale real-world PMU dataset from the Eastern Interconnection of the United States.
- The combination of an additional encoder network and the proposed distributed streaming platform based on Apache Hadoop, Kafka, and Spark allows Bi-AnoGAN to process a large-scale streaming PMU data and detect power system events within 2 milliseconds.

The rest of this paper is organized as follows: Section II introduces the overall framework of the proposed algorithm. Section III presents the technical methods of the proposed Bi-AnoGAN algorithm. Section IV quantifies the performance of the proposed power system event detection framework with a large-scale real-world PMU dataset.

II. Problem Setup and Overall Framework

This section sets up the online power system event detection problem and presents the overall framework of the proposed Bi-AnoGAN algorithm. The input to the proposed algorithm is streaming PMU data from multiple measurement channels (e.g., real power, reactive power, voltage magnitude, and frequency). We regroup the streaming PMU data into sequential rolling window samples $[\mathbf{x}_1, \mathbf{x}_2, ...]$ with two parameters: window size w, and step size s as shown in Fig. 1. The ith sample $\mathbf{x}_i$ is a three-dimensional tensor, where the three dimensions represent the PMU ID, time-stamp, and measurement channel.

The overall framework of our proposed method is shown in Fig. 2. This consists of two key modules: offline training and online event detection. The offline training module uses a pre-processed historical PMU dataset of normal operation

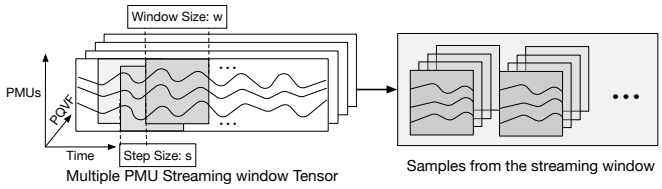

Fig. 1. Visualization of sliding window samples.

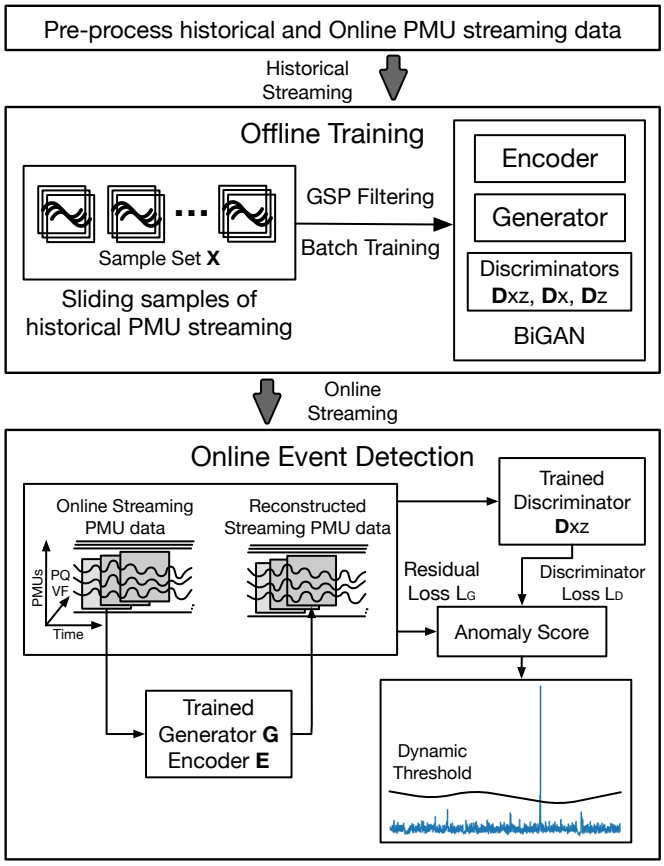

Fig. 2. Overall framework of online event detection with PMU data.

periods to train the Bi-AnoGAN model. The design of the offline training module is described in subsection III-D. Note that the raw PMU data goes through a pre-processing data pipeline, which removes outliers (including event data, if any) and fills in missing data.

The real-time event detection module uses the trained Bi-AnoGAN model and a snapshot of the streaming PMU data sample to calculate the corresponding anomaly score. The online event detection algorithm is presented in detail in subsection III-E.

The offline training and real-time event detection modules are developed in a distributed streaming platform. The technical details about the implementation of the big data platform are described in subsection III-F.

III. Technical Methods

In this section, we first review the basics of GAN and BiGAN models. Numerical instability problems often occur when training GAN models on large-scale datasets with high-dimensionality. Due to the complexity of the large-scale PMU dataset, we introduce Wasserstein Loss, which improves the convergence of GAN models in the training process in subsection III.B. To overcome another numerical issue of cycle-consistency violation in training GAN models, we introduce a conditional entropy regularization in subsection III.C. In subsection III.D, we present the design and training procedure of the proposed Bi-AnoGAN model. The online detection of power system events using the Bi-AnoGAN is described in subsection III.E. The implementation of the proposed online event detection algorithm in a distributed streaming platform is shown in subsection III.F.

A. Review of GAN and BiGAN Models

In standard GANs, there are two neural networks, a generator $\mathbf{G}$ and a discriminator $\mathbf{D}$. To train the GAN on a dataset $\mathbf{\Omega} = [\mathbf{x}_1, \mathbf{x}_2, ...]$, the generator maps a random variable $\mathbf{z} \sim \mathcal{N}(0, I)$ to the data space, and the discriminator attempts to distinguish between real samples $\mathbf{x}_i$ and samples produced by the generator. These two models are in a competitive relationship. $\mathbf{G}$ tries to generate samples that can deceive $\mathbf{D}$ while $\mathbf{D}$ tries to increase its discriminating ability. The training of GANs can be formulated as solving a two-player min-max problem with the following objective function:

$$\min_{\mathbf{G}} \max_{\mathbf{D}} V(\mathbf{D}, \mathbf{G}) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[\log \mathbf{D}(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[\log(1 - \mathbf{D}(\mathbf{G}(\mathbf{z})))], \tag{1}$$

where $p(\mathbf{x})$ is the distribution over the dataset $\mathbf{\Omega}$ in the data space and $p(\mathbf{z})$ is the distribution over the variable $\mathbf{z}$ in the noise space.

In the original AnoGAN model, once the generator is trained, a separate optimization problem is solved to find a latent space representation $\mathbf{z}$ that corresponds to a sample $\mathbf{G}(\mathbf{z})$ that is most similar to the given sample. This 'noise space optimization problem' incurs high computational cost [22].

To identify the noise space variable for a given sample in a computationally efficient manner, BiGAN [26] proposes to simultaneously learn an encoder network $\boldsymbol{E}$ during the training process of GANs.

The training of BiGAN can be formulated as solving the min-max problem with the following objective function:

$$\min_{\mathbf{G},\mathbf{E}} \max_{\mathbf{D}} V(\mathbf{D}, \mathbf{E}, \mathbf{G}) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[\log \mathbf{D}(\mathbf{x}, \mathbf{E}(\mathbf{x}))] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[\log(1 - \mathbf{D}(\mathbf{G}(\mathbf{z}), \mathbf{z}))] \tag{2}$$

Note the difference between this equation and the previous: in the BiGAN objective, the discriminator takes in a *pair* of points - one from the data space and one from the noise space.

B. Wasserstein Loss

We encountered the mode collapse problem when training the original BiGAN model using the spatial-temporal PMU data. Specifically, the generator always generates similar samples no matter what the noise feed. In other words, the

generator only captures a small portion of the entire data distribution. Thus, we adopt the Wasserstein loss [27] as the adversarial loss when training the discriminator network, which is known to yield improved convergence.

The training of BiGAN with Wasserstein loss can be formulated as the following optimization problem:

$$\min_{\mathbf{G},\mathbf{E}} \max_{\mathbf{D}\in\mathcal{D}} V_{xz}(\mathbf{D}_{xz},\mathbf{E},\mathbf{G}) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\mathbf{D}_{xz}(\mathbf{x},\mathbf{E}(\mathbf{x}))] - \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\mathbf{D}_{xz}(\mathbf{G}(\mathbf{z}),\mathbf{z})], \tag{3}$$

where $\mathcal{D}$ denotes the set of 1-Lipschitz continuous functions f, which satisfy $\|f(x_1)-f(x_2)\| \leq \|x_1-x_2\|, \forall x_1, x_2 \in dom(f)$. To enforce the Lipschitz constraint, we include a gradient penalty regularization term (with respect to the input) proposed by [28] during training.

The loss for the discriminator is nonstationary, which may cause the momentum-based methods (Adam) to perform worse [27]. To further improve the GAN training, we use the RMSProp optimization algorithm instead of the Adam optimizer, which is known to perform well on nonstationary problems.

C. Conditional Entropy Constraint

The joint distributions $(\mathbf{x}, \mathbf{E}(\mathbf{x}))$ and $(\mathbf{G}(\mathbf{z}), \mathbf{z})$ should theoretically be identical with adequate training. Nevertheless, in practice, this is often unsatisfied. The training process does not necessarily converge to the solution of the saddle-point problem. This leads to the violation of the desirable cycle-consistency property, which says that the inferred $\mathbf{z}$ of a corresponding $\mathbf{x}$ can reconstruct $\mathbf{x}$ itself with high probability [31]. To overcome this problem, the framework ALICE [31] proposes to approximate and constrain the conditional entropy in an adversarial manner by adding a term V_x in the objective function to encourage cycle consistency of $\mathbf{x} = \mathbf{G}(\mathbf{E}(\mathbf{x}))$.

In Bi-AnoGAN, we propose to enforce an additional conditional entropy constraint by adding V_z to the objective function to encourage cycle consistency of $\mathbf{z} = \mathbf{E}(\mathbf{G}(\mathbf{z}))$.

Specifically, V_x and V_z are defined as:

$$V_x(\mathbf{D}_x,\mathbf{E},\mathbf{G}) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\mathbf{D}_x(\mathbf{x})] - \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\mathbf{D}_x(\mathbf{G}(\mathbf{E}(\mathbf{x})))] \tag{4}$$

$$V_z(\mathbf{D}_z,\mathbf{E},\mathbf{G}) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\mathbf{D}_z(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\mathbf{D}_z(\mathbf{E}(\mathbf{G}(\mathbf{z})))] \tag{5}$$

Each of these terms uses a new discriminator - denoted $\mathbf{D}_x$ and $\mathbf{D}_z$ - which attempt to detect differences between samples and their cycle-reconstructions.

Gathering all of these terms together yields our final objective function for model training:

$$\min_{G,E} \max_{D_{xz},D_x,D_z} V(\boldsymbol{D}_{xz},\boldsymbol{D}_x,\boldsymbol{D}_z,\boldsymbol{E},\boldsymbol{G}) = V_{xz}(\boldsymbol{D}_{xz},\boldsymbol{E},\boldsymbol{G}) + V_x(\boldsymbol{D}_x,\boldsymbol{E},\boldsymbol{G}) + V_z(\boldsymbol{D}_z,\boldsymbol{E},\boldsymbol{G}) \tag{6}$$

D. Design and Offline Training Details of Bi-AnoGAN

The architecture of the Bi-AnoGAN neural network and the offline training framework are summarized in Fig. 3. The Bi-AnoGAN neural network model has three components: an encoder $\boldsymbol{E}$, a generator $\boldsymbol{G}$, and three discriminators $\mathbf{D}_x$, $\mathbf{D}_z$, and $\mathbf{D}_{xz}$. The offline training procedure is performed over normal (non-event) historical PMU data.

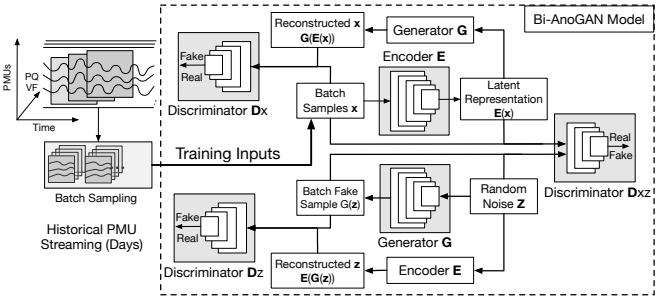

Fig. 3. Offline training procedure of Bi-AnoGAN.

1) Encoder: The encoder, $\mathbf{E}$, maps PMU data samples to the noise space. We denote the output of the encoder as $\hat{\mathbf{z}}$, which is derived as $\hat{\mathbf{z}} = \mathbf{E}(\mathbf{x})$. It consists of five convolutional layers, and each convolutional layer is followed by a batch normalization layer and a leaky *ReLU* activation function. The specific architecture design of the encoder can be found in Section IV-B2. Note that the encoder, $\mathbf{E}$, is introduced in the proposed framework to accelerate the identification of the noise space variable for a given PMU data sample.

2) Generator: The generator, $\mathbf{G}$ takes the noise space samples $\mathbf{z}$ and generates PMU data samples $\hat{\mathbf{x}} = \mathbf{G}(\mathbf{z})$ $\left(\mathbf{z} \in \mathbb{R}^{h\times b}\right)$. The generator has the reverse architecture as the encoder. It also consists of five *transpose* convolutional layers, and each transpose convolutional layer is followed by a batch normalization layer and a leaky *ReLU* activation function layer. The specific architecture design of the generator can be found in Section IV-B2.

3) Discriminators: The discriminators, $\mathbf{D}_x$, $\mathbf{D}_z$, and $\mathbf{D}_{xz}$, in the Bi-AnoGAN model distinguish between real PMU samples and generated samples created by the generator and encoder. $\mathbf{D}_x$ and $\mathbf{D}_{xz}$ have a similar structure as that of the encoder and generator, $\mathbf{D}_z$ consists of four fully connected layers. The specific architecture design of the discriminators can be found in Section IV-B2.

4) Graph Signal Processing (GSP)-based PMU Sorting: Convolutional layers use convolution to process local information. This works best when the local patches of data are highly correlated. However, if the PMUs are randomly placed in the raw input data, then it does not necessarily have strong spatial correlations. Since the power network topology information and the location of the PMUs are not available to us, we adopt a graph signal processing (GSP)-based sorting algorithm to systematically arrange the PMUs [20]. The GSP-based sorting algorithm places PMUs with higher correlations closer to each other. Algorithm 1 describes the GSP-based sorting algorithm.

Algorithm 1 GSP-based PMU Sorting Algorithm

1: Calculate the correlation coefficients between PMUs;
2: Construct weight matrix W and graph Laplacian L;
3: Perform eigendecomposition of L;
4: Sort PMUs according to eigenvector corresponding to L's second smallest eigenvalue;

5) One-sided label smoothing: GANs are infamously difficult to train in comparison to other deep learning models.

Some common issues for GANs are overfitting, high computation cost, vanishing gradients, exploding gradients, and poor convergence. To stabilize and accelerate the training process of GANs, we apply one-sided label smoothing [32]. Specifically, we assign hard labels of 1 and 0 to the first 70% of training batches and soft labels of 0.9 and 0.1 to the remaining 30% of training batches.

E. Online Power System Event Detection Algorithm

Once the Bi-AnoGAN model is trained, we can use it to perform online power system event detection. The online event detection algorithm consists of two modules: the streaming PMU data reconstruction module and the event detection module (Fig. 4). In the PMU data reconstruction module, streaming PMU data is sent into the encoder and reconstructed by the generator. The event detection module then calculates an anomaly score by combining the discriminator loss and the residual loss between the original and reconstructed PMU data sample. To cope with the time-varying operating conditions of the power system, we design a dynamic anomaly score threshold. If the anomaly score of the current frame of PMU data exceeds the dynamic threshold, then online event detection algorithm reports an abnormal event.

1) Reconstruction of streaming PMU data: The first step of the proposed online event detection algorithm is to reconstruct the streaming PMU data with the trained Bi-AnoGAN model. All sliding window samples of the streaming PMU data are fed into the encoder to produce the latent representations, which are then fed into the generator to output the reconstructed PMU data samples.

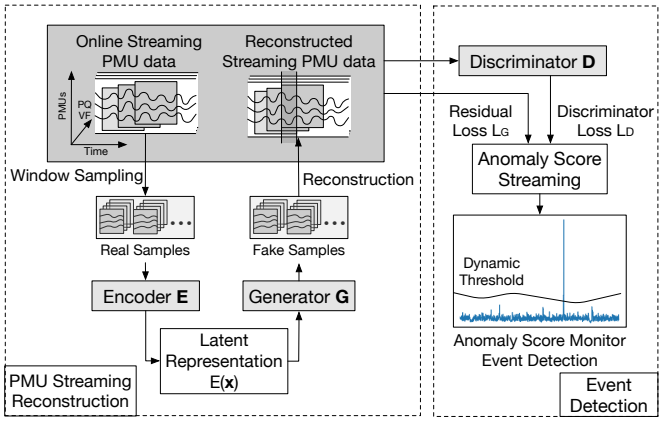

Fig. 4. Overall framework of the proposed online event detection algorithm.

2) Anomaly Score Calculation: We will calculate the anomaly score associated with each sliding window PMU sample. We denote this anomaly score as $\boldsymbol{L}$, which consists of the residual loss $\boldsymbol{L}_G$ and the discriminator loss $\boldsymbol{L}_D$.

The residual loss $\boldsymbol{L}_G$ is the reconstruction error of the generated PMU sample. It can be calculated as the Euclidean distance between original and reconstructed PMU data sample as shown in (7):

$$\mathbf{L}_G = \|\mathbf{x} - \mathbf{G}(\mathbf{E}(\mathbf{x}))\|_2 . \tag{7}$$

As shown in Section III-D, the GAN-based model is trained to generate PMU data during normal system operations. It can reconstruct PMU data during normal periods but not during event periods. Therefore, a large difference between the observed PMU data sample and the reconstructed one indicates the presence of an event.

The discriminator loss $\boldsymbol{L}_D$ is the discriminator's loss function against real PMU data samples:

$$\mathbf{L}_D = BCE(\mathbf{D}_{xz}(\mathbf{x}, \mathbf{E}(\mathbf{x}))), \tag{8}$$

where BCE denotes the binary cross-entropy loss function [33]. As shown in Section III-D3, the discriminator is trained to distinguish between the fake and real PMU data samples. It can serve a tool for anomaly detection as well. This is because power system event data will not look like real data to a model trained on data during normal system operations.

The final anomaly score of a streaming PMU data sample can be calculated as (9):

$$\mathbf{L} = \lambda \mathbf{L}_G + (1 - \lambda)\mathbf{L}_D, \tag{9}$$

where λ is a hyper-parameter between 0 and 1, which balances the weight between the residual loss and the discriminator loss.

We use the last time stamp of a sliding window, t to denote the time index of the corresponding anomaly score $\mathbf{L}_t$. By calculating the anomaly score of every sliding window, we will obtain a time series of the anomaly scores.

3) Dynamic Threshold for the Anomaly Score: The operating condition of a power system is constantly changing, which may result in small fluctuations in the anomaly scores. Instead of using a fixed threshold, we propose a dynamic threshold for the anomaly score to detect power system events. The dynamic threshold can be calculated as a function of the mean and standard deviation of the anomaly scores in the past minute as shown in (10):

$$\textit{Threshold} = mean(\mathbf{L}_{t-60:t-1}) + c \times std(\mathbf{L}_{t-60:,t-1}), \tag{10}$$

where c is a hyper-parameter that controls how frequent the proposed algorithm reports an anomaly. The selection of hyper-parameters including c will be discussed in Section IV.

F. Prototype Online Platform for Event Detection

To promote the adoption of the proposed data-driven power system event detection algorithm, we develop a prototype system that is capable of handling large-scale PMU data in real-time operations. The prototype system is developed using a distributed streaming platform consists of computer clusters equipped with Apache Hadoop, Kafka, and Spark. It collects streaming PMU data, performs data pre-processing, and identifies power system events in real-time. This prototype system raises the fault tolerance and reduces the computation time to detect the event by leveraging data replication, page cache, and multi-machine computation resources. The overall architecture of the prototype online event detection system is shown in Fig. 5. The prototype system consists of three components: data collection, data pre-processing, and online event detection, which will be described in detail below.

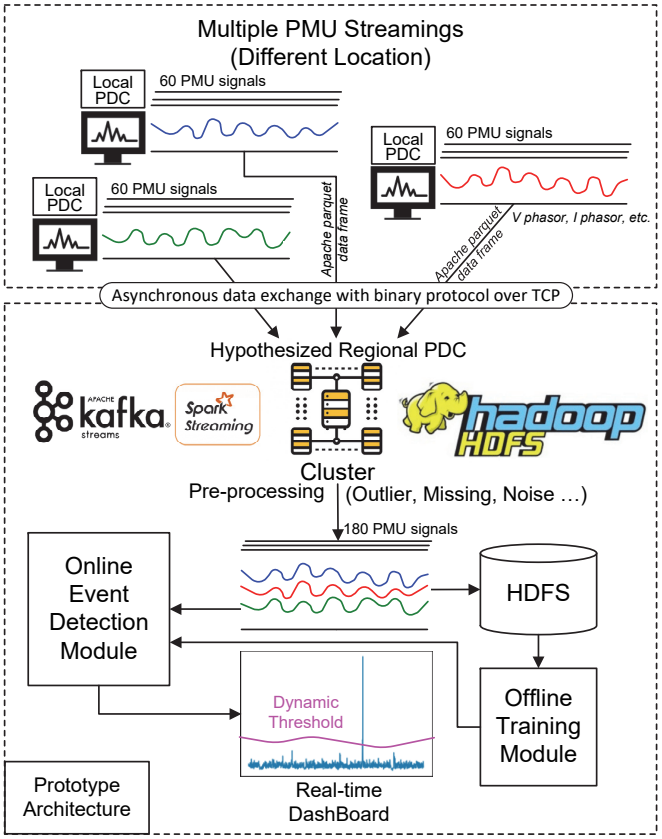

Fig. 5. Overview of prototype online event detection system.

1) Data Collection: In the data collection process, we use the Apache Kafka system which has the desired properties of fault-tolerant and highly-available. Specifically, we use the Apache Kafka to collect streaming PMU data from different locations. The Kafka producers at the remote locations keep gathering PMU measurements and push them into the computer cluster. The Kafka consumer at the centralized computer facility combines streaming data from multiple PMUs, which will then be used for online event detection. All data collected by Apache Kafka is replicated by multiple brokers over the computing cluster. By taking advantage of the high-speed sequential disk reads and writes with zero-copy messages processing mechanism, the Apache Kafka is compatible with online computing environment.

2) Data Pre-processing: We adopt Spark Streaming to perform data pre-processing because it provides high throughput and fault-tolerant processing for live data streams. This is achieved by efficiently utilizing the computation and storage resources on multiple machine in the computer cluster. The Spark Streaming takes collected PMU data from Kafka and uses high-level functions such as map-reduce and join to perform data pre-processing. The pre-processed PMU data is pushed out to the Hadoop Distributed File System (HDFS) and live dashboards.

The data pre-processing procedure consists of five steps.

a) Step One: We first remove the invalid data using the PMU status flag. According to the IEEE standard for synchrophasor data transfer in power systems (IEEE C37.118.2-2011) [34], the last two bits of the status flag indicate the PMUs' conditions such as no data available, PMU in test mode, and PMU error. Besides the status flag, the following thresholds are used to remove bad measurements.

- Voltage magnitude: $(-\infty, 0) \cup (1.5\text{p.u.}, +\infty)$,
- Voltage angle: $(-\infty, -180°) \cup (+180°, +\infty)$,
- Current magnitude: $(-\infty, 0) \cup (10\text{kA}, +\infty)$,
- Current angle: $(-\infty, -180°) \cup (+180°, +\infty)$,
- Frequency: $(-\infty, 59\text{Hz}) \cup (61\text{Hz}, +\infty)$.

b) Step Two: We fill in the missing PMU data using the baseline method described in [35].

c) Step Three: We remove the PMUs with excessive bad data and replace the removed PMUs with the other PMUs that have the highest Pearson correlation coefficient during periods with no missing data.

d) Step Four: We calculate real power P and reactive power Q using the voltage and current phasor data.

e) Step Five: We construct the three-dimensional input tensors with 4 measurements P, Q, $|V|$ and F.

All five steps are implemented as a streaming data pipeline and performed in the Apache Spark Streaming cluster. The pre-processed streaming data is continuously fed into the event detection module for the online power system monitoring. At the same time, the pre-processed data is stored in the HDFS for the offline training procedure.

3) Online Event Detection: The online power system event detection module described in Section III-E is also integrated into the Spark Streaming framework. We use Elephas, an extension of Keras, to enable distributed inference with Bi-AnoGAN models at scale with Spark. The data-parallel inference of the deep learning models allows us to handle a very large-scale distributed PMU dataset efficiently. The trained power system event detection model is distributed to the individual machines in the cluster to enable real-time inference of streaming PMU data. By leveraging the power of Spark Streaming and taking advantage of the computing resources on multiple GPUs and machines, our proposed event detection module achieves high computation efficiency, which will be demonstrated in Section IV-D.

IV. Numerical Study

In this section, we validate the accuracy and computational efficiency of the proposed Bi-AnoGAN real-time event detection algorithm with a large-scale real-world PMU dataset from the Eastern Interconnection of the United States. We will first introduce the raw dataset and present the data pre-processing procedures that created the training and testing data for the Bi-AnoGAN. Next, the performance of the proposed event detection algorithm is compared with the state-of-art event detection methods in terms of accuracy and computation time. Finally, we analyze and interpret the representations learned by the proposed Bi-AnoGAN model.

A. Raw Dataset and Data Pre-processing

The dataset comprises 187 PMUs' data from the Eastern Interconnection of the U.S. transmission grid which is one of the largest grids across the globe with a wide variety of

grid components and dynamic event signatures. The data spans over one and a half years from May 2016 to December 2017. The raw data contains positive sequence voltage and current phasors and frequency measurements. The sampling frequency of the PMUs is 30 Hz. The size of the raw dataset is over 15 terabytes. In the data pre-processing step, we remove 8 PMUs from the analysis due to prevalent missing values and bad data. This brings the total number of PMUs down to 179.

The raw dataset also include 889 power system event labels with start time of 1-minute resolution. The event labels were generated based on the event log/history recorded by electric utilities and regional transmission operators. 807 of the events are voltage-related events, which are mostly caused by line and transformer faults. The remaining 82 events are frequency-related events, which are caused by generator tripping. Note that the event locations and PMU locations are not disclosed by the data provider. We reviewed all events and created fine-grained event timing with 0.1 second resolution. We use 200 voltage-related events to tune the hyper-parameters of our proposed event detection algorithm. The remaining events labels are used to evaluate the performance of the proposed and baseline algorithms.

We put together a 3-node Hadoop cluster to store and pre-process the raw dataset. The HDFS manages the data storage while Apache Spark and Kafka perform data pre-processing.

For each event in the dataset, we extract 4 minutes of PMU data around the event start time. The goal of the event detection algorithm is to accurately identify the start time of the events. Two examples of the 4-minute window for voltage-related and frequency-related events are shown in Figure 6.

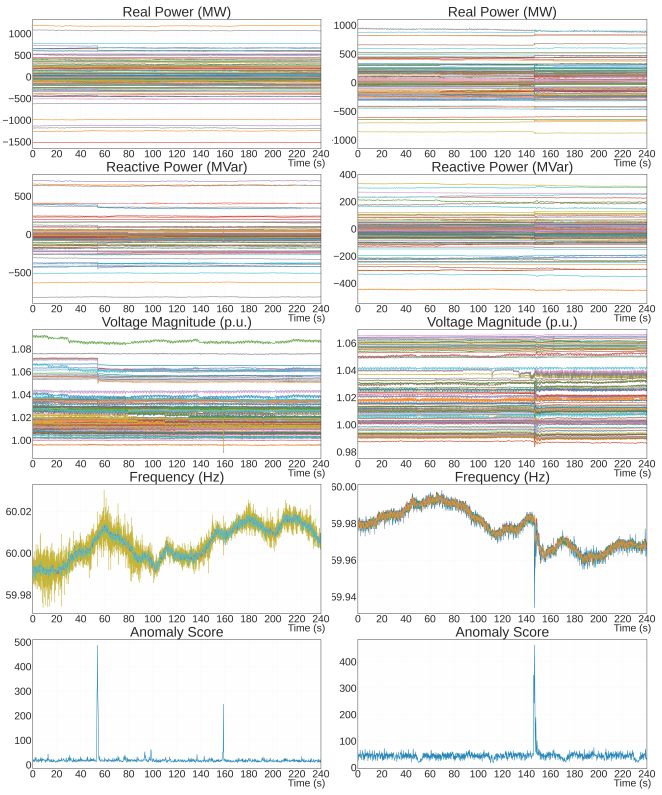

Fig. 6. Sample PMU data of the 4-minute event window and the anomaly scores obtained from Bi-AnoGAN. The left subfigure is a voltage-related event and the right subfigure is a frequency-related event.

B. Setup of the Bi-AnoGAN Model

1) Preparation of Training Data for Bi-AnoGAN Model: The training of GAN-based models often takes hours. Given the limited computing power, in this numerical study we only retrains the Bi-AnoGAN model every six months. In practical implementations, one could retrain the Bi-AnoGAN model on a daily basis, which could further improve the event detection accuracy. We split the one and half years of PMU measurements into the following three periods: 05/01/2016-11/01/2016, 11/01/2016-05/01/2017, and 05/01/2017-12/31/2017. The daily load profiles within each periods demonstrate stronger similarity than the daily load profiles between different periods.

For each period, the first day's PMU data is used to train the Bi-AnoGAN model. The remaining PMU data in the period is used to test the event detection performance of the proposed model. Each training sample has has a window size of $w = 30$ (1 second) and a step size of $s = 30$, which means there is no overlap among the training samples. The total number of samples in the training day is 86400. Each sample is a 3-dimensional tensor with 30 time stamps, 179 PMUs and 4 measurement channels.

As discussed in Section III-D, the real PMU data used to train Bi-AnoGAN model needs to come from normal system operating conditions. The 889 labeled events provided to us are only a small subset of all abnormal events in the raw dataset. There are many hidden unlabeled events in the raw dataset that were not captured by the system operators. To ensure that the trained Bi-AnoGAN model achieves a high level of performance, we need to exclude abnormal PMU data samples from the training dataset. To this end, we adopt a state-of-the-art graph signal processing (GSP)-based abnormal event detection algorithm [36] to remove potential power system events from the training dataset. After the abnormal events are removed, each daily training dataset contains around 86000 samples, which are sufficient to properly train our proposed GAN-based event detection model.

2) Bi-AnoGAN Model Architecture: The specific architecture design of the key components of the Bi-AnoGAN model are presented below.

a) Encoder: The encoder consists of five convolutional layers and a fully connected layer. The detailed architecture parameters of all layers are shown in Table I, where s refers to the stride, f refers to the number of convolutional filters per layer, and k refers to convolutional kernel width. The input shape, $30 \times 179 \times 4$ corresponds to 30 time stamps, 179 PMUs and 4 measurement channels. Note that zero padding is adopted in the convolutional layers to control the shrinkage of data dimension and avoid losing information at the boundaries.

b) Generator: The structure of the generator is opposite to the encoder. It starts with a fully connected layer, which is followed by five convolutional layers. The *first* convolutional layer of the generator is the transposed version of the *last* convolutional layer of the encoder. The transposed convolutional

TABLE I
ARCHITECTURE DETAILS OF THE ENCODER MODULE

Input Shape	Layer Detail	Output Shape
30×179×4	Conv_1 (k=5, s=2, f=64)	15×90×64
15×90×64	Leaky_ReLU+Batch Norm	15×90×64
15×90×64	Conv_2 (k=5, s=2, f=64)	8×45×64
8×45×64	Leaky_ReLU+Batch Norm	8×45×64
8×45×64	Conv_3 (k=5, s=2, f=128)	4×23×128
4×23×128	Leaky_ReLU+Batch Norm	4×23×128
4×23×128	Conv_4 (k=3, s=2, f=256)	2×12×256
2×12×256	Leaky_ReLU+Batch Norm	2×12×256
2×12×256	Conv_5 (k=3, s=1, f=256)	2×12×256
2×12×256	Leaky_ReLU+Batch Norm	2×12×256
2×12×256	Flatten+Fully Connected	100

layers of the generator and the original convolutional layers of the encoder share the same hyperparameters.

c) Discriminators: The architectural details for the discriminators $\mathbf{D}_x$ and $\mathbf{D}_{xz}$ are shown in Table II. Both of them consists of five convolutional layers, and a fully connected layer. The only difference is that the third dimension of the input of $\mathbf{D}_x$ is 4, which corresponds to the 4 measurement channels. $\mathbf{D}_{xz}$ has an additional input, which is calculated by feeding the input from the noise space $\mathbf{z}$ to a fully connected layer with a single output.

TABLE II
ARCHITECTURAL DETAILS FOR THE DISCRIMINATORS $\mathbf{D}_x$ AND $\mathbf{D}_{xz}$

Input Shape	Layer Detail	Output Shape
30×179× (4, 5)	Conv_1 (k=5, s=2, f=64)	15×90×64
15×90×64	Leaky_ReLU+Batch Norm	15×90×64
15×90×64	Conv_2 (k=5, s=2, f=64)	8×45×64
8×45×64	Leaky_ReLU+Batch Norm	8×45×64
8×45×64	Conv_3 (k=5, s=2, f=128)	4×23×128
4×23×128	Leaky_ReLU+Batch Norm	4×23×128
4×23×128	Conv_4 (k=3, s=2, f=256)	2×12×128
2×12×128	Leaky_ReLU+Batch Norm	2×12×128
4×23×128	Conv_5 (k=3, s=1, f=256)	2×12×256
2×12×256	Leaky_ReLU+Batch Norm	2×12×256
2×12×256	Flatten+Fully Connected	1

The discriminator $\mathbf{D}_z$ consists of five fully connected layer. The hyerparameters of the architecture are shown in Table III.

TABLE III
ARCHITECTURAL DETAILS FOR THE DISCRIMINATOR $\mathbf{D}_z$

Input Shape	Layer Detail	Output Shape
100	Fully Connected + Leaky_ReLU	100
100	Fully Connected + Leaky_ReLU	100
100	Fully Connected + Leaky_ReLU	100
100	Fully Connected + Leaky_ReLU	100
100	Flatten+Fully Connected	1

3) Training Setup and Hyperparameter Tuning for Bi-AnoGAN model: The optimizer used for Bi-AnoGAN model training is ADAM [37] with the learning rate of 0.0001. The slope of the LeakyReLU activation function is 0.2. The batch size is 256. The Bi-AnoGAN model is built with Tensorflow 2.3.0 and trained on NVIDIA GeForce RTX 2080 Ti GPU. The training time is about 8 hours for each daily PMU dataset.

There are two key hyperparameters in the Bi-AnoGAN model. The first is λ, which strikes a balance between residual loss and discriminator loss in determining the anomaly score. Note that during an event, the magnitude of the discriminator loss is about 10 times larger than the magnitude of the residual loss. The second is c, which controls how frequent the proposed algorithm reports an anomaly. A smaller c reports a larger number of abnormal events and yields a higher number of false positives.

These two hyperparameters are tuned using the validation dataset that contains 200 confirmed events to avoid overfitting. A grid search is carried out by increasing λ from 0 to 1 with an increment of 0.1. We also gradually increase c from 3 to 6 with an increment of 0.1. $\lambda = 0.9$ and $c = 4.8$ perform the best on the validation dataset and are used to evaluate our proposed algorithm over the testing dataset.

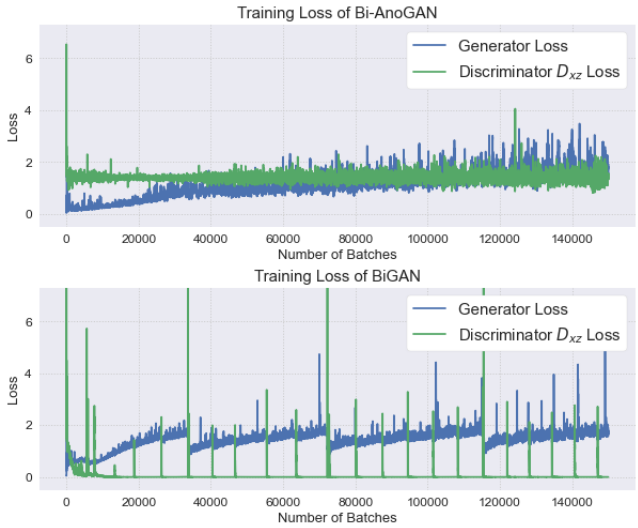

Fig. 7. Training batch loss for the Bi-AnoGAN and BiGAN models.

4) Numerical Stability of the Training Process for Bi-AnoGAN Model: The training loss of the proposed Bi-AnoGAN and the baseline BiGAN model are illustrated in Figure 7. As shown in the figure, without the Wasserstein loss, conditional entropy constrains, and GSP-based sorting algorithm, the baseline BiGAN model encounters the problem of vanishing gradient, where the discriminator loss frequently drops to zero. On the other hand, our proposed Bi-AnoGAN model exhibits great convergence and numerical stability during the training process.

C. Baseline Methods

The performance of our proposed Bi-AnoGAN model is compared with four state-of-the-art baseline methods.

1) OLAP: The online algorithm for PMU data processing (OLAP) is capable of performing both missing value replacement and anomaly detection for streaming PMU data [12]. OLAP decomposes the streaming PMU data matrix into a low-rank matrix and a noise matrix using the singular value decomposition (SVD) and sequentially computes changes in the ratio between the first two singular values to detect abnormal behaviors.

2) GSP-based Method: A graph signal processing (GSP) based method [36] leverages both temporal and spatial correlations in the streaming PMU data to detect events. It includes an offline training stage and an online event detection stage. The offline training stage constructs a graph Laplacian with a fitted vector autoregressive model. In the online stage, a computationally efficient GSP-based algorithm is used to detect power system events.

3) AnoGAN Model: AnoGAN [22] is a GAN-based method originally designed for anomaly detection in image dataset. This method first trains a deep convolutional adversarial network (DCGAN) [30] and uses it to recover each test data sample. The residual loss of the data reconstruction along with the discriminator feature are used to indicate anomaly. We tailor this method for power system event detection. Note that to reconstruct the data sample, AnoGAN needs to solve a computationally expensive optimization problem.

4) Deep Neural Network-based Classification Model: Although the training of our proposed Bi-AnoGAN does not not reply on any labeled event data, the tuning of two hyperparameters of our proposed model needs a small number of event labels. Thus, we introduce a deep neural network-based classification model as another baseline model, which classifies the PMU data into the normal operation class and the event class. We adopt ResNet50 [38] as the classification model. To make a fair comparison, the classification model is trained with the same dataset as the Bi-AnoGAN. In other words, the training data includes only 200 labeled events and the data during normal system operation conditions.

D. Event Detection Accuracy and Computation Efficiency

1) Event Detection Accuracy: We apply the proposed Bi-AnoGAN and the three baseline methods on the testing dataset to detect power system events. Various metrics quantifying the accuracy of the event detection algorithms on voltage and frequency-related events are reported separately in Tables IV and V. True positive denotes the successful detection of abnormal system events (less than 1 second difference to label). False positive denotes incorrectly indicating the presence of an event when it is not present. False negative denotes failure to indicate the presence of an event when it is present. Precision quantifies the number of event predictions that actually are events. Recall quantifies the number of event predictions made out of all event samples in the testing dataset. F_1 score is the harmonic mean of precision and recall.

As shown in Tables IV and V, the proposed Bi-AnoGAN algorithm achieves a higher level of precision, recall and F_1 scores for both voltage and frequency-related events than all four baseline methods. Note that the deep neural network-based classifier does not achieve a satisfactory event detection performance due to the lack of sufficient amount of labeled events for training.

The comparative advantage of the proposed Bi-AnoGAN over the baseline methods is more pronounced for detecting frequency-related events.

2) Ablation Study: We also performed an ablation study, which examines the performance of the proposed Bi-AnoGAN by removing two key components to understand the contribution of them to the overall algorithm. These two components are the conditional entropy constraints introduced in Section III-C and GSP-based PMU sorting introduced in Section III-D. The F_1 scores of the proposed model and the proposed model without key components for voltage and frequency-related events are shown in Tables VI. "Without CE" means removing the conditional entropy constraints from the proposed model, "Without GSP" means removing the GSP-based sorting component from the proposed model, and "Without CE & GSP" means removing both the conditional entropy constraints and GSP-based sorting components. The ablation study results show that the conditional entropy constraints alone improves the F_1 scores by 2.32% and 2.3% for voltage and frequency-related events. The GSP-based PMU sorting component alone improves the F_1 scores by 2.78% and 1.7% for voltage and frequency-related events. The model with both these two method improves the F_1 scores by 3.94% and 2.88% for voltage and frequency-related events.

TABLE IV
ACCURACY OF DETECTION FOR VOLTAGE-RELATED EVENTS

	Bi-AnoGAN	OLAP	GSP-based	AnoGAN	ResNet50
True Positive	**584**	534	561	512	502
False Positive	**42**	67	138	77	231
False Negative	**23**	73	46	95	105
Precision	**93.29%**	88.89%	80.26%	86.92%	68.48%
Recall	**96.21%**	89.55%	92.42%	84.34%	82.70%
F_1 **Score**	**94.73%**	89.22%	85.91%	85.61%	74.92%

TABLE V
ACCURACY OF DETECTION FOR FREQUENCY-RELATED EVENTS

	Bi-AnoGAN	OLAP	GSP-based	AnoGAN	ResNet50
True Positive	**82**	72	71	75	73
False Positive	**5**	56	19	46	40
False Negative	**0**	10	11	7	9
Precision	**94.25%**	53.33%	78.89%	61.98%	64.60%
Recall	**100%**	88.89%	86.59%	91.46%	89.02%
F_1 **Score**	**97.04%**	66.67%	82.56%	73.89%	74.87%

TABLE VI
ABLATION STUDY RESULTS FOR BI-ANOGAN

	Bi-AnoGAN	Without CE	Without GSP	Without CE & GSP
Voltage: F_1	94.73%	92.41%	91.95%	90.79%
Frequency: F_1	97.04%	94.74%	95.34%	94.16%

We performed another ablation study to separately quantify the impact of residual loss and discriminator loss on the event detection performance of the proposed model.

The precision, recall, and F_1 score of the proposed model for detecting voltage and frequency events with only one of the two loss terms and both loss terms are reported in Table VII. As shown in the table, our GAN-based model with just one of the two loss terms still achieves fairly high F_1 scores. When we adopt both residual loss and discriminator loss, the proposed model yields the highest F_1 score.

A high discriminator loss indicates that the PMU sample

contains an abnormal event with high probability. However, if the actual event is not severe, the discriminator loss may be relatively low. This may result in failure to detect an event, which is considered to be false negative. Thus, if we only use the discriminator loss, the event detection model could achieve high precision and relatively low recall as shown in Table VII. This is because the calculation of recall does not include false negative.

The residual loss measures the discrepancy between the incoming PMU data and the PMU data during normal operating conditions that the model has learned. The occurrence of an event induces large residual loss. However, constant variations in net load could cause frequent changes in system operating conditions. This may also lead to a large residual loss and results in false positives. Thus, if we only use residual loss in the model, we could achieve high recall and relatively low precision as shown in Table VII. This is because the calculation of recall does not include false positive.

When we synergistically combine the two losses with a carefully tuned hyper-parameter λ, the proposed model achieves higher recall and precision.

TABLE VII
IMPACT OF RESIDUAL LOSS AND DISCRIMINATOR LOSS ON EVENT DETECTION PERFORMANCE

Voltage/Frequency	Discriminator Loss	Residual Loss	Combined Loss
Precision	**96.06%/98.66%**	90.26%/91.11%	93.29%/94.25%
Recall	92.42%/90.24%	**96.21%/100%**	**96.21%/100%**
F_1 **Score**	94.20%/94.26%	93.14%/95.34%	**94.73%/97.04%**

We also perform a sensitivity analysis to analyze how the number of events used for hyper-parameter tuning affects the model performance. Specifically, we increase the number events from 50 to 200 with an increment of 50. The tuned hyper-parameters and the corresponding F_1 scores on voltage and frequency-related events are reported in Table VIII. The results show that the event detection performance and the tuned hyper-parameters are not very sensitive to the number of events.

TABLE VIII
SENSITIVITY ANALYSIS WITH RESPECT TO THE NUMBER OF EVENTS USED FOR HYPER-PARAMETERS TUNING

Number of Events	50	100	150	200
λ	0.85	0.9	0.9	0.9
c	5.0	4.9 s	4.8 s	4.8
F1-score (Voltage)	93.28%	94.21%	94.73%	94.73%
F1-score (Frequency)	94.90%	96.42%	97.04%	97.04%

To evaluate the computational efficiency of the proposed Bi-AnoGAN and the baseline algorithms, we calculate the average computation time for all voltage and frequency-related events in the test datasets. The average runtime of different algorithms for performing event detection over a 4-minute window with 179 PMUs are reported in Table IX. The proposed Bi-AnoGAN achieves the second highest computation efficiency among the four algorithms. Both Bi-AnoGAN and GSP-based algorithms can be applied in real-time operations because they are able to finish event detection for a single snapshot of PMU dataset within 2 milliseconds. Compared to the AnoGAN, the proposed Bi-AnoGAN achieves a 64-fold increase in computation efficiency. This significant improvement in the GAN-based event detection method is attained through the introduction of the additional encoder network, which accelerates the mapping from the sample space to the noise space.

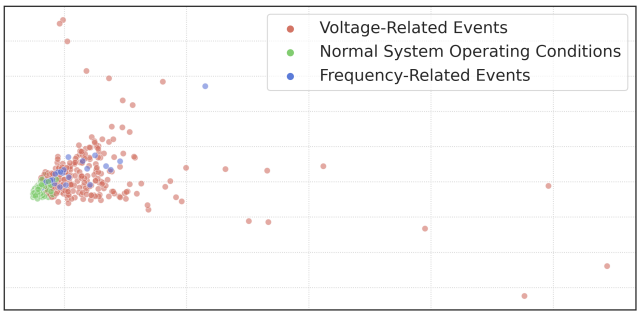

Fig. 8. Noise space representations of voltage-related events, frequency-related events, and normal system operation conditions.

TABLE IX
AVERAGE RUNTIME OF DIFFERENT ALGORITHMS FOR EVENT DETECTION

	Bi-AnoGAN	OLAP	GSP-based	AnoGAN
Voltage Events	**13.59 s**	21.75 s	7.16 s	876.58 s
Frequency Events	**13.47 s**	20.98 s	7.31 s	843.89 s

E. Visualization of Noise/Latent Space Representations

The performance of the proposed Bi-AnoGAN event detection algorithm primarily relies on the quality of presentations produced by its encoder. Increasing the quality of representations enhances the interpretability of the proposed GAN-based algorithm. In this subsection, we visualize the noise/latent space representations of the events and normal operating conditions of the Eastern Interconnection.

For each labeled event, we extract two data samples. The first sample is comprised of one second of data from the begging of the event. The second is comprised of one second of data before the event start time. We first map these samples to the noise space representations using the encoder $\mathbf{E}$, and then perform principal component analysis to reduce the dimensionality of the noise space representation to two. The dimension reduced representations of the event samples and normal system operating conditions are depicted in Fig. 8. As shown in the figure, the samples representing normal operating conditions are all located in the lower left corner. The samples representing voltage and frequency-related events are scattered and separated from the normal operating conditions. In particular, event samples located far from the lower left corner often represent stronger frequency and voltage disturbance.

For two individual voltage and frequency-related events, we also visualize how the one second samples move around in the representation space as time processes from 30 seconds before

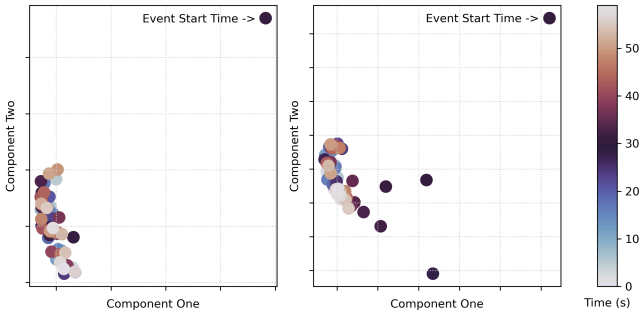

Fig. 9. Noise space representations of 1-minute samples surrounding the event. The left subfigure is a voltage-related event and the right subfigure is a frequency-related event.

the event start time to 30 seconds after the event start time in Fig. 9. Note that the event start time is the mid-point (30-second) of the 1-minute time window. As shown in the figure, the noise space representations before the event start time are clustered in the lower left corner. Once the event start, the corresponding representation shifts to the upper right corner of the representation space. It can also be observed that one second after the event start time, the noise space representations of the voltage-related event samples shift back to the cluster representing normal operating conditions. In contracts, it took a few seconds for the noise space representations of the frequency-related event samples to shift back to the lower left corner. This is because a typical voltage-related event ends within a second, whereas a frequency-related event often lasts a few seconds.

V. Conclusion

This paper develops an accurate and computationally efficient GAN-based power system event detection algorithm using PMU data. Our proposed Bi-AnoGAN model achieves great computational efficiency by training an additional encoder network, which can quickly map a PMU data sample to the representation space. By introducing conditional entropy constraints in the objective function and graphical signal processing-based PMU sorting in the input layer, our proposed Bi-AnoGAN yields higher event detection accuracy. The numerical study results on real-world PMU data from Eastern Interconnection of the U.S. transmission grid show that Bi-AnoGAN algorithm outperforms the state-of-the-art algorithms in terms of event detection accuracy. The Bi-AnoGAN algorithm is implemented on an Apache Spark platform to perform online event detection on real-world streaming PMU data. Testing results show that our proposed platform finishes event detection for each snapshot of PMU data within 2 milliseconds, which makes it suitable for real-time applications. This work can be further extended to not only detect events but also classify power system events.

References

[1] A. Monti, C. Muscas, and F. Ponci, *Phasor measurement units and wide area monitoring systems*. Academic Press, 2016.

[2] A. G. Phadke and T. Bi, "Phasor measurement units, WAMS, and their applications in protection and control of power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 4, pp. 619–629, 2018.

[3] M. Ghorbanian, S. H. Dolatabadi, and P. Siano, "Big data issues in smart grids: A survey," *IEEE Systems Journal*, vol. 13, no. 4, pp. 4158–4168, 2019.

[4] C. Tu, X. He, Z. Shuai, and F. Jiang, "Big data issues in smart grid–a review," *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 1099–1107, 2017.

[5] A. Von Meier, E. Stewart, A. McEachern, M. Andersen, and L. Mehrmanesh, "Precision micro-synchrophasors for distribution systems: A summary of applications," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2926–2936, 2017.

[6] D. Kim, T. Y. Chun, S. Yoon, G. Lee, and Y. Shin, "Wavelet-based event detection method using PMU data," *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1154–1162, 2017.

[7] S. S. Negi, N. Kishor, K. Uhlen, and R. Negi, "Event detection and its signal characterization in PMU data stream," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3108–3118, 2017.

[8] S. Sohn, A. J. Allen, S. Kulkarni, W. M. Grady, and S. Santoso, "Event detection method for the PMUs synchrophasor data," in *2012 IEEE Power Electronics and Machines in Wind Applications*, 2012, pp. 1–7.

[9] J. Wu, J. Xiong, P. Shil, and Y. Shi, "Real time anomaly detection in wide area monitoring of smart grids," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2014, pp. 197–204.

[10] Z. Ling, R. C. Qiu, X. He, and L. Chu, "A new approach of exploiting self-adjoint matrix polynomials of large random matrices for anomaly detection and fault location," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 548–558, 2021.

[11] H. Zhu and G. B. Giannakis, "Sparse overcomplete representations for efficient identification of power line outages," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2215–2224, 2012.

[12] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1006–1013, 2016.

[13] C. Hannon, D. Deka, D. Jin, M. Vuffray, and A. Y. Lokhov, "Real-time anomaly detection and classification in streaming PMU data," in *2021 IEEE Madrid PowerTech*, 2021, pp. 1–6.

[14] J. Shi, N. Yu, E. Keogh, H. K. Chen, and K. Yamashita, "Discovering and labeling power system events in synchrophasor data with matrix profile," in *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*. IEEE, Nov. 2019, pp. 1827–1832.

[15] E. Khaledian, S. Pandey, P. Kundu, and A. K. Srivastava, "Real-time synchrophasor data anomaly detection and classification using isolation forest, KMeans, and loOP," *IEEE Trans. Smart Grid*, vol. 12, no. 3, May 2021.

[16] W. Wang, H. Yin, C. Chen, A. Till, W. Yao, X. Deng, and Y. Liu, "Frequency disturbance event detection based on synchrophasors and deep learning," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3593–3605, 2020.

[17] L. Zhu, D. J. Hill, and C. Lu, "Hierarchical deep learning machine for power system online transient stability prediction," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2399–2411, 2020.

[18] O. P. Dahal and S. M. Brahma, "Preliminary work to classify the disturbance events recorded by phasor measurement units," in *2012 IEEE Power and Energy Society General Meeting*, Jul. 2012, pp. 1–8.

[19] V. Miranda, P. A. Cardoso, R. J. Bessa, and I. Decker, "Through the looking glass: Seeing events in power systems dynamics," *International Journal of Electrical Power & Energy Systems*, vol. 106, pp. 411 – 419, 2019.

[20] J. Shi, B. Foggo, and N. Yu, "Power system event identification based on deep neural network with information loading," *IEEE Trans. Power Syst.*, vol. 36, no. 6, pp. 5622–5632, 2021.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, Dec. 2014, p. 2672–2680.

[22] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157.

[23] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *2018 IEEE International Conference on Data Mining (ICDM)*, Nov. 2018, pp. 727–736.

[24] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," in *Asian conference on computer vision*. Springer, 2018, pp. 622–637.
[25] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 703–716.
[26] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *CoRR*, vol. abs/1605.09782, 2016. [Online]. Available: http://arxiv.org/abs/1605.09782
[27] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. PMLR, 06–11 Aug 2017, pp. 214–223.
[28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5769–5779.
[29] M. Lucic, K. Kurach, M. Michalski, O. Bousquet, and S. Gelly, "Are GANs created equal? a large-scale study," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 698–707.
[30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4*, 2016.
[31] C. Li, H. Liu, C. Chen, Y. Pu, L. Chen, R. Henao, and L. Carin, "ALICE: Towards understanding adversarial learning for joint distribution matching," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5501–5509.
[32] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 2234–2242.
[33] A. Kapoor, *Hands-On Artificial Intelligence for IoT: Expert machine learning and deep learning techniques for developing smarter IoT systems*. Packt Publishing, 2019.
[34] "Ieee standard for synchrophasor data transfer for power systems," *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–53, 2011.
[35] B. Foggo and N. Yu, "Online PMU missing value replacement via event-participation decomposition," *IEEE Trans. Power Syst.*, vol. 37, no. 1, pp. 488–496, 2022.
[36] J. Shi, B. Foggo, X. Kong, Y. Cheng, N. Yu, and K. Yamashita, "Online event detection in synchrophasor data with graph signal processing," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Tempe, AZ, USA, Nov. 2020, pp. 1–7.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9*, 2015.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.