Learning-Based Real-Time Event Identification Using Rich Real PMU Data

Yuxuan Yuan, *Graduate Student Member, IEEE*, Yifei Guo, *Member, IEEE*, Kaveh Dehghanpour, Zhaoyu Wang, *Senior Member, IEEE*, and Yanchao Wang

***Abstract*—A large-scale deployment of phasor measurement units (PMUs) that reveal the inherent physical laws of power systems from a data perspective enables an enhanced awareness of power system operation. However, the high-granularity and non-stationary nature of PMU data and imperfect data quality could bring great technical challenges for real-time system event identification. To address these challenges, this paper proposes a two-stage learning-based framework. In the first stage, a Markov transition field (MTF) algorithm is exploited to extract the latent data features by encoding temporal dependency and transition statistics of PMU data in graphs. Then, a spatial pyramid pooling (SPP)-aided convolutional neural network (CNN) is established to efficiently and accurately identify power events. The proposed method fully builds on and is also tested on a large real-world dataset from several tens of PMU sources (and the corresponding event logs), located across the U.S., with a time span of two consecutive years. The numerical results validate that our method has high identification accuracy while showing good robustness against poor data quality.**

***Index Terms*—Event identification, Markov transition field, phasor measurement unit, spatial pyramid pooling.**

I. Introduction

LARGE-SCALE blackouts, such as the Northeast blackout of 2003 in the U.S., which started with a local event but eventually affected 50 million customers, continuously remind us of the need for better and faster event detection and identification to enhance the wide-area situational awareness of power system operation [1]. Recent years have seen a rapid growth in the deployment of phasor measurement units (PMUs), providing a unique opportunity for preventing cascading failures and blackouts [2]. Unlike the supervisory control and data acquisition (SCADA) system that only offers power system monitoring at steady state, PMU collects high-granularity voltage and current phasor, frequency, and frequency variation (e.g., 30 or 60 samples per second in the U.S.), which enables capturing the fast dynamics of power systems. Therefore, exploiting PMU data for real-time event identification has attracted increasing attention.

Related Works: The existing works on PMU-based event *detection* and *identification* can be mainly classified into two categories: 1) signal processing-based methods [3]–[6]; and 2) machine learning-based methods [7]–[10]. In [3], a wavelet-based method was designed for detecting the event occurrence and classifying events. In [4], a dynamic programming-based swinging door trending method was developed to detect the start-time and placement of events. The authors in [5] proposed a quadratic fitting method to recover the dynamics of events and a knowledge-based criterion to classify events. In [6], the extended Kalman-filtering algorithm was applied to detect voltage events. Inspired by the recent success of machine learning techniques in data analytics, many researchers have adopted different machine learning methods to identify the types of events. In [7], a multiclass extreme learning machine classifier was utilized to perform near-real-time automatic event diagnosis. In [8], a data-driven algorithm consisting of an unequal-interval reduction method and principal component analysis was proposed to detect and locate events using PMU data. In [9], a hierarchical clustering-based method was proposed to determine the types of events, using several characteristics of multidimensional minimum volume enclosing. In [10], the k-nearest neighbor and support vector machine classifiers were exploited to perform event identification based on different pattern creation methods.

Challenges: While researchers have contributed numerous valuable works on this topic, several critical questions remain open, which may challenge the practical deployment of these methods. 1) Data quality issues, such as bad data, dropouts, and time error, arise frequently in reality, and can easily lead to misclassification of bad data as events, which were ignored in the previous works. Basically, data quality issues can disjoint the dimensional consistency of data samples during the training procedure, thus resulting in a failed event identification. To avoid this situation, a common solution is to drop data points with quality issues. However, this strategy is hard to apply during online testing, such as real-time power system operation, because data points cannot be dropped. Thus, poor robustness against data quality makes the data-driven event identification models insufficiently convincing in practice. 2) Most of the previous methods rely on the complicated data imputation and optimization in online event identification, which may affect the real-time performance of these methods [8]. 3) Some existing

Q1

Manuscript received June 17, 2020; revised December 5, 2020, February 25, 2021, and May 11, 2021; accepted May 16, 2021. This work was supported by the U.S. Department of Energy Office of Electricity under DEOE0000910. Paper no. TPWRS-01005-2020. *(Corresponding author: Zhaoyu Wang.)*

The authors are with the Department of Electrical~~,~~ and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: yuanyx@iastate.edu; yifeig@iastate.edu; kavehdeh1@gmail.com; wzy@iastate.edu; yanchaow@iastate.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPWRS.2021.3081608.

Digital Object Identifier 10.1109/TPWRS.2021.3081608

studies require the spatial information of PMUs (i.e., detailed system topology), which may be unavailable due to privacy protection.

Our Contributions: To solve these questions, in this paper, a learning-based method is developed to identify power event types using PMU measurements. The proposed method focuses on providing an efficient and accurate event identifier to enhance situational awareness, while introducing robustness against data quality issues in real-time operation. To achieve this, two stages are included in the proposed method: 1) the time-varying statistical characteristics of the PMU data (i.e., voltage magnitude and frequency variation) are extracted using a Markov-based time-series feature extraction. In this stage, the time-series PMU data is converted into image-like data. 2) A robust event identification model is developed to build a mapping relationship between the results of stage I and event types by adopting a spatial pyramid pooling (SPP) strategy in a convolutional neural network (CNN)-based model. One salient merit of the proposed method is that the dimension of the testing data can be different with that of the training data, thus providing a superior solution to the online data quality problem. Specifically, after the model is trained using the historical PMU data and the corresponding event labels, when a new data sample shows data quality issues, the relevant data points can be marked and then directly excluded. The remaining good-quality PMU data of arbitrary dimension is assigned as input to the trained model, and the output will be the estimated event type. Hence, our model does not generate any artificial data point that could reduce the accuracy of event identification. Moreover, our method provides an efficient way for encoding time-series PMU data into image-like data, which preserves both temporal ordering and statistical dynamics, under incomplete information of the transmission system (i.e., topology). To validate the performance of our method, a large amount of real-world PMU data over two consecutive years, gathered from several tens of PMUs throughout the U.S., and sufficient real event labels are utilized for model development and testing. It should be noted that the proposed method is fine-tuned on our dataset to optimize the values of the model hyperparameters. However, the methodology is general and can be applied to any other PMU datasets after some fine-tuning procedure. This is true for any data-driven solution. Our method is designed to address common challenges in all PMU datasets. The large number of real event labels contained in this dataset provides a good foundation for developing an efficient and practical event identification model. Besides, we have tested the sensitivity of our model accuracy to the size of missing data to demonstrate the robustness of the model.

The rest of this paper is constructed as follows: Section II introduces the available PMU dataset and data pre-processing. In Section III, an Markov-based time-series feature extraction algorithm is utilized to summarize the hidden features of PMU data in graphs. Section IV proposes the SPP-aided CNN-based event identification method based on MTF-graphs. The numerical results are analyzed in Section V. Section VI presents research conclusions.

TABLE I
STATISTICAL SUMMARY OF THREE INTERCONNECTIONS

	A	B	C
Record period	1 year	2 years	2 years
Data size	3 TB	5 TB	12 TB
Number of PMUs	215	43	188
Sample rates [frames/s]	30	30/60	30
Total number of events	29	4854	1884
Number of unidentified events	0	0	634
Resolution of event record	Daily	Minute	Minute
Number of event causes	13	3911	1883

II. PMU DATA DESCRIPTION AND PRE-PROCESSING

A. PMU Dataset Description

The available PMU dataset includes more than 440 PMU sources that are installed in the Eastern, Western, and Electric Reliability Council of Texas interconnections at different voltage levels with the nominal frequency of 60 Hz. For convenience, let A, B and C denote the three interconnections hereinafter. They are equipped with 215, 43 and 188 PMUs, respectively. Most data segment is archived at 30 frames/s and the remaining is archived at 60 frames/s. Each PMU measures voltage and current phasor, system frequency, frequency variation rate, and PMU status information. The dataset spans a time period of around two consecutive years (2016–2017). The total size of the dataset is more than 20 TB (in Parquet form).[1] These data files were read in Python and MATLAB environments. In total, around 670 billion sampling points have been used to conduct the analyses.

B. Event Log Description

Since data-driven event identification can be converted to a classification problem, real event labels play a vital role in providing the ground truths. A unique advantage of our dataset is that we not only have 20 TB PMU measurements but also enough real event labels recorded by utilities. This is exactly the type of data that system operators have access to and can utilize for event identification model development in reality. Hence, the available dataset provides a good foundation for developing an efficient and practical event identification model. In summary, a total of 6767 event labels, consisting of 6133 known events and 634 unknown events (where the event type entry is empty or unspecified), are included in our dataset. Each available event label contains the interconnection number, start timestamp, end timestamp, event type, and high-level event cause, of which a detailed statistical summary is presented in Table I. The type and timestamp of events have been verified by matching them with the corresponding protection relay records, ensuring the high confidence of these event labels. Note that the proofreading of these events was done by the data providers. Thus, due to sensitive information protection purposes, this information

[1]The pacific northwest national laboratory (PNNL) team has formatted the raw dataset to 20 TB in Parquet form so as to save memory while facilitating the learning algorithm design and validation.

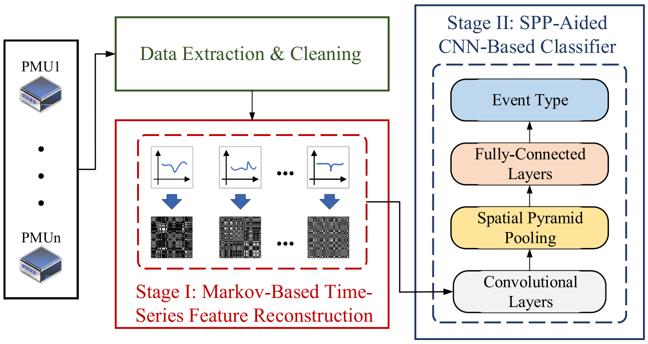
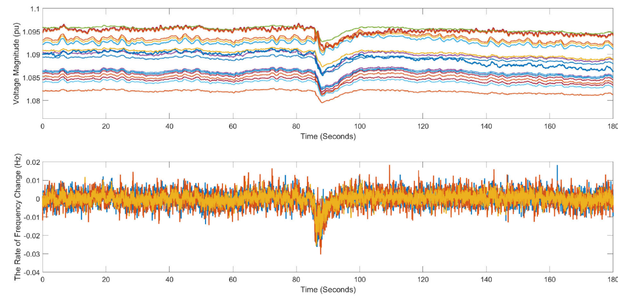
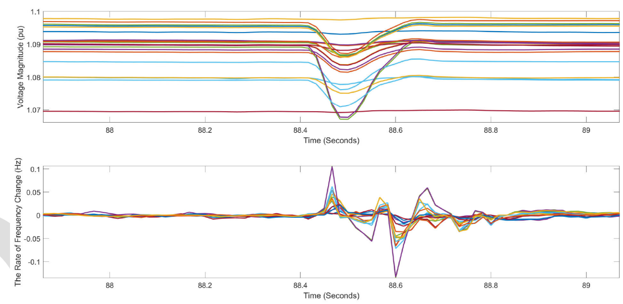


Fig. 1. Illustration of two-stage learning-based event identification framework. In the data extraction and cleaning, a 2-s time window is selected to extract the event data and then PMU status information and engineering intuition are utilized to eliminate the missing and/or bad data for training dataset. The stage I encodes the PMU data to a graph by characterizing the transition probability and temporal dependency. The stage II constructs an end-to-end mapping between the graphs and the event types by leveraging deep learning techniques.



(a) Frequency event example.



(b) Line outage example.

Fig. 2. Plots of multiple PMUs' data for two events.

is unavailable for us and cannot be utilized as input to the proposed event identification model. Moreover, the definition of each event type was left entirely up to the data providers. We did not make any manual changes to the event labels. In other words, we try to simulate the real situation faced by the system operators. The proposed model is based solely on the event labels from the data providers instead of integrating much prior event information, thus ensuring the practicability of our model. Since three interconnections have different event categorization systems, it is impossible to directly merge the three event logs into a single dataset. Therefore, in this work, we have used the event log from one interconnection that has the most known events (around 4800 known events) for model development and validation.

C. Data Pre-Processing

As a real-world dataset, our dataset is not perfect and has some vague and incomplete information. Hence, to eliminate the impact of these problems on model training, the available PMU dataset is initially passed through a data pre-processing that combines various methods and engineering intuitions. Note that this data pre-processing is developed on empirical knowledge rather than purely heuristic. The goal of the data pre-processing is twofold: 1) select an appropriate analysis-window to extract the data into frames corresponding to pre-event and event states for training a learning model; 2) eliminate missing and bad data caused by communication and meter malfunction.

Following the start timestamp in the event log, we have extracted 60 seconds of pre-event and 120 seconds of post-event data to visualize power events. Fig. 2 shows event plots of all PMUs in the interconnection. Note that this figure is plotted against a frequency event and line outage on the data provider's event log. As is demonstrated in Fig. 2, it is clear that the most critical changes happen around the inception of event, but the lengths of changes are different for different PMU-recordings. In addition, these figures show that the length of the change can be at second- or sub-second-levels for different types of events. Thus, to apply PMU-base event identifiers in real-world application, a second-level analysis-window is needed. Hence, in this work, a 2-second analysis-window is selected to extract the event data [10], [11]. Obviously, the 2-second analysis-window cannot cover all events, but it contains sufficient event features to determine the types. This has been demonstrated using numerical results. Basically, using the data-driven event identification model, most of the events could be identified with multiple post-event samples rather than data from the entire event. Moreover, the 2-second analysis-window can avoid the curse of dimensionality for model development and ensure the real-time performance of the event identification. Noted that the previous method also utilizes a similar analysis window for PMU-based event identification [2]. According to the sampling rate of PMUs, each analysis window should include 120 data points. However, as described in Table. I, the resolution of the available event logs collected by the data providers is minute-level, thus, not sufficient to directly extract the start timestamp of events at the second-level. To tackle this, a statistical algorithm is proposed to apply for the entire data set, which can detect the transition between the normal and event states. The rationale behind this is that, since PMUs are synchronized, the variations in PMU-recordings will occur at the same time. It should be noted that this statistical algorithm can be bypassed if the resolution of event logs is sufficient for a 2-second analysis-window. The proposed algorithm involves the following steps:

- **Step 1:** Define and initialize the 2-second event set $\mathbb{E} = \emptyset$ and the event counter $i \leftarrow 1$.
- **Step 2:** Select the i'th event from the event logs and then extract related 60 seconds of pre-event and 120 seconds of post-event data $\mathbb{D}_i$.

- **Step 3:** Utilize the modified z-score for $\mathbb{D}_i$ and identify the time stamps with the minimum score, of which the set is denoted as $\mathbb{T}_i$ [12].
- **Step 4:** Find the time stamp with the highest frequency of minimum values belonging to $\mathbb{T}_i$, denoted as t_i^*.
- **Step 5:** Sort $\mathbb{D}_i$ based on the 2-second analysis-window, and find the 2-second data that includes t_i^*, denoted as $\mathbb{D}_i^*$; add $\mathbb{D}_i^*$ to $\mathbb{E}$.
- **Step 6:** $i \leftarrow i+1$; go back to Step 2 until i equals the total number of events.

When the 2-second event dataset is obtained, PMU status flags information is utilized to perform data quality assessment [13]. The status flags are in binary form and all information is aligned as 16-bit long. Each bit corresponds to a different status based on IEEE C37.118.2-2011 standard, such as bits 03-00 reflecting the trigger reason and bits 05-04 showing the time error (i.e., asynchrony). When the value of the status flag is 0 in the decimal format, data can be used properly; otherwise, data should be removed due to the various PMU malfunction. Also, the engineering intuitions is used to design several simple threshold-based methods for further detecting the data quality problems which are not identified by the PMU. For example, a number of data windows contained a single sample with an unreasonable value compared to the nominal value, which is dismissed as bad data. Following our data quality assessment, when a consecutive missing/bad data occurs, the data is excluded from our study because it is hard to provide a high accuracy data imputation for the consecutive missing/bad data. The rest of the missing/bad data are filled and corrected through a linear interpolation. [11].

III. Markov-Based PMU Data Feature Extraction

Despite PMUs' high precision and ability to capture system dynamics, PMU-based event identification via simple features (i.e., voltage magnitude and frequency) is a difficult task. The source of this challenge is the non-stationary characteristics of real-world PMU data, which is caused by sudden variations in system behavior during events [3]. To address this issue, in this paper, a Markov matrix-based feature extraction method known as MTF is adopted to discover additional data features for event identification [14]. It should be noted that the feature extraction is a common theme as well in modeling any time-series data. Also, our MTF method is a general method that can be applied to any other PMU dataset for feature engineering.

Basically, the MTF method encodes the temporal dependency and transition statistics of PMU data in a compact metric. Compared to traditional feature extraction methods, such as Fourier transform, wavelet transform, and multidimensional minimum volume enclosing ellipsoid, our feature extraction method offers two unique advantages: 1) The MTF method can preserve both temporal ordering and statistical dynamics of the PMU data, thus improving accuracy. 2) Using the MTF method, PMU data is converted into the image-like structure without requiring any spatial information of PMUs (i.e., topology), which provides a basis for utilizing the recently-developed image-based deep learning techniques. In this work, based on the previous work [15], voltage magnitudes and frequency variations are selected as event indicators because they are deemed to closely correlated to power events. Hence, the input to the MTF method is the voltage magnitude and frequency variation of each PMU. Note that the MTF method can in principle be applied to the remaining PMU measurements (i.e., voltage phase angles and current phasor measurements). However, adding more inputs does not necessarily improve the performance of the event identification model due to the increased model complexity. Let V_i^j denotes the voltage magnitude data during event i as recorded by the j'th PMU. The objective of the proposed feature extraction method is to map this continuous signal $V_i^j = \{V_i^j(k) | k \in \mathbb{N}, V_i^j(k) \in \mathbb{R}\}$ to a network $\mathbb{G} = (\mathbb{O}, \mathbb{B})$, which consists of a set of vertices $\mathbb{O}$ and a set of edges $\mathbb{B}$ connecting different vertices. Since a direct mapping from continuous data to a network with finite nodes is not possible, we utilize a quantile-based approach to obtain a discretized dictionary for V_i^j [16]. Specifically, given a V_i^j, we create q quantile bins (states) $S_1, \ldots, S_q$ and assign each $V_i^j(k), k = 1, \ldots, n$, to the corresponding bins,[2] (see Fig. 3). While different strategies can be applied to assign V_i^j to the bins, our quantile strategy ensures that all bins in each data have the same number of points [16]. Compared to other strategies, quantile mapping is more data-specific and has shown the highest identification accuracy on our dataset. Following this strategy, a weighted adjacency matrix $W \in \mathbb{R}^{q \times q}$ is developed by counting the transitions among quantile bins similar to a first-order Markov chain. Each entry of W is a non-negative real number representing a transition probability that is determined as follows:

$$w_{S_a,S_b} = \Pr\left\{V_i^j(t) \in S_a | V_i^j(t-1) \in S_b\right\}, \quad \forall S_a \in \{S_1, \ldots, S_q\}, S_b \in \{S_1, \ldots, S_q\}. \tag{1}$$

After normalization by $\sum_{S_b} w_{S_a,S_b} = 1$, W becomes a standard Markov matrix that contains the transition probability on the voltage magnitude axis. However, W fails to capture the higher order temporal dependencies as it is based on a first-order Markov chain. Hence, to preserve information across the temporal dimension, we extend matrix W to a new matrix $M \in \mathbb{R}^{n \times n}$ by aligning each probability along the temporal order, as follows [14]:

$$M = \begin{bmatrix} m_{11} & \cdots & m_{1n} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{nn} \end{bmatrix} \tag{2}$$

with

$$m_{k_1,k_2} = w_{S_a,S_b}, \; V_i^j(k_1) \in S_a, V_i^j(k_2) \in S_b, \forall k_1, k_2.$$

So, the kth row of M represents the transition probabilities between the k'th point and all data points. In this way, M encodes the transition dynamics of the PMU data between different time lags. This process is applied to the remainder of event

[2]Note that, $S_1, \ldots S_q$ are different for different i, j. For simplicity, we omit the indexes i, j here.

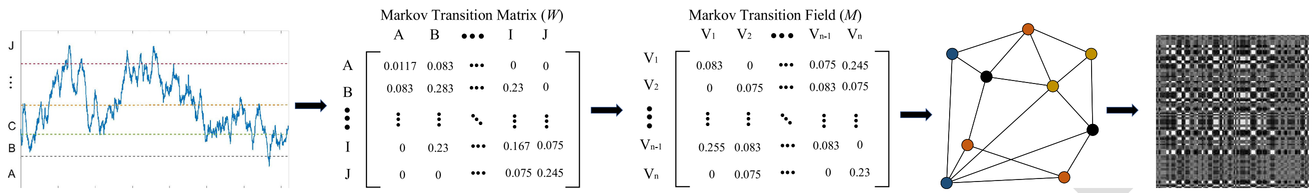


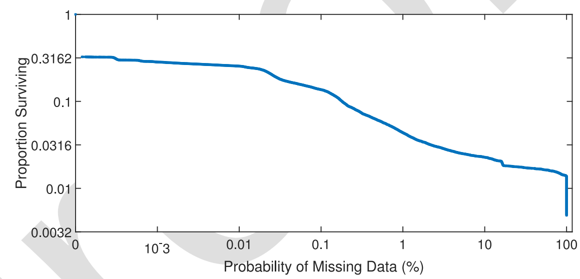
Fig. 3. Illustration of the proposed encoding map of MTF. As shown in Fig. 3, the square matrix M can be interpreted as a network $\mathcal{G}$, where m_{k_1,k_2} represents weight of the edge between any two nodes k_1 and k_2. The nodes in different colors precisely match different time points of V_i^j.

dataset including voltage magnitudes and frequency variations to obtain the MTF-based graph set, which are used for training our learning-based event identification model.

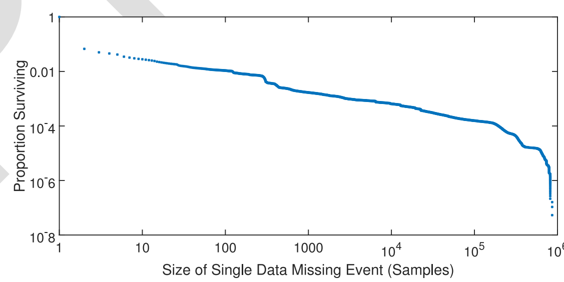
IV. SPP-Aided CNN-Based Event Identifier

In this section, we lay out our PMU-based event identification strategy. Considering that PMU-based models are developed to identify events and perform supervisory protection in real-time, high speed and accuracy are required [10]. Also, the robustness of the model should be considered because data quality problems are common in current PMUs. Several previous works have mentioned the impact of data quality problems in data-driven event identification task [2], [13]. Here, we also provide a basic statistical analysis, *survival function*, on our 20 TB PMU dataset to show the probability of occurrence of data quality problems. Specifically, the PMU status flag information and engineering intuition are leveraged to mark the data that has quality issues. The details are described in our data pre-processing procedure (Section II). Then, survival function is defined for the probability of missing data per PMU per day as follows:

$$S(k) = \Pr\left\{\frac{\text{number of missing data per PMU per day}}{\text{total number of data per PMU per day}} > k\right\}. \quad (3)$$



(a) Survival function of probability of missing/bad data



(b) Survival function of size of single data quality problem

Fig. 4. Statistical analysis results about data quality problems using 20 TB PMU data.

As can be seen in Fig. 4 (a), PMUs show data quality issues more than 30% of time which is a non-negligible number. Moreover, the survival function of size of each individual data quality issue is obtained and plotted in Fig. 4 (b). It is clear that around 3% of data quality issues have more than 10 consecutive missing and bad data. Considering the extremely high sampling rate of the PMU, it is quite common to have consecutive missing and bad data due to long communication failure intervals or equipment malfunction.

These statistical analysis results confirm the need for a robust event identification model that can work well under various data quality issues. For most of the existing PMU-based event identification models, data quality issues cause a data dimension imbalance problem since these models only accept inputs with fixed dimensions. In other word, the testing input dimension of the models should be exactly equal to that of the training data (i.e., if n-dimensional data is used for training, then the data-driven model allows for n-dimensional test inputs). In the offline training procedure, the data dimension imbalance problem can be solved by dropping data points and performing data imputation techniques. It should be noted that our data pre-processing utilizes these solutions to address the data quality issues of the training dataset. However, in the online testing procedure, these solutions are not appropriate because data points cannot be dropped, and it is hard to generate accurate artificial data points for consecutive missing and bad data that is also common based on our statistical analysis. Meanwhile, many system operators avoid performing data imputation techniques for PMU data in the industry since they prefer not to modify the PMU data. Hence, to achieve reliable real-time event identification, we propose an SPP-aided CNN-based event classification method. As shown in Fig. 5, this method constructs an end-to-end mapping relationship between MTF-based graphs and the event types using deep learning techniques. The proposed method offers a unique advantage: the dimension of the testing data can be different with that of the training data, which provides a natural solution for the online PMU data quality problems. The rationale behind this is that the fixed-size constraint of the learning-based event identifier is removed by adopting a global pooling strategy, SPP.

Here, consider a training set $\{\mathbb{V}, \mathbb{F}, \mathbb{L}\}$, $\mathbb{V} := \{v^{(1)}, \ldots, v^{(h)}\}$ and $\mathbb{F} := \{f^{(1)}, \ldots, f^{(h)}\}$ are the MTF-based graphs based on the PMU-based voltage magnitude and the frequency variation data, and $\mathbb{L} := \{l^{(1)}, \ldots, l^{(h)}\}$ is the corresponding event label set from the event logs. Then, the probability that the label $l^{(i)}$ of $\{v^{(i)}, f^{(i)}\}$ is equal to j can be

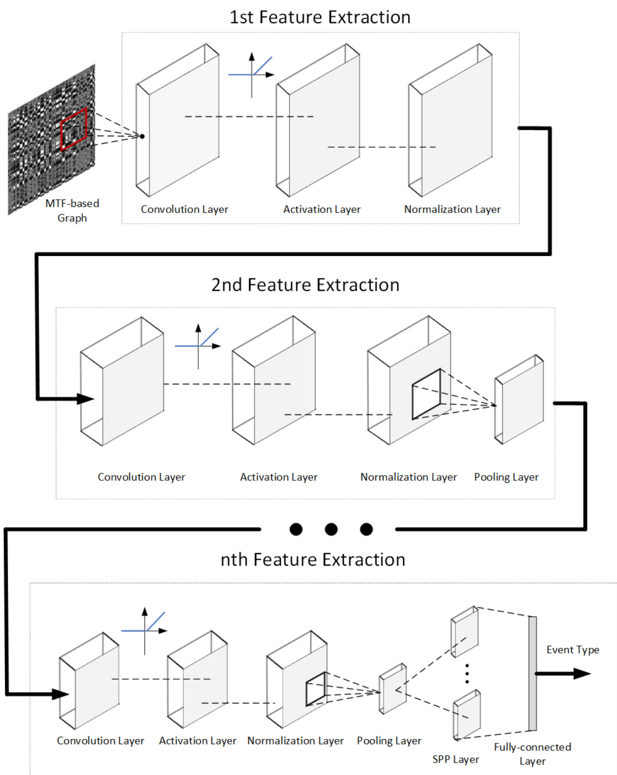


Fig. 5. Proposed SPP-aided CNN-based event classifier. As can be seen, our model is a multiple-layer architecture that consists of different layers. The input of this mode is the MTF-based graphs and the outcome is the event type.

calculated by:

$$\Pr\left\{l^{(i)} = j|z^{(j)}\right\} = \frac{\exp\left(\theta_j(v^{(i)}, f^{(i)})\right)}{\sum_{c=1}^{o} \exp(\theta_c(v^{(i)}, f^{(i)})} \tag{4}$$

where, o is the number of event types and $\theta_c(\cdot)$ denotes the mathematical model in the proposed SPP-aided CNN method. The learning parameters are obtained by minimizing the following cost function J:

$$J := -\frac{1}{h}\sum_{i=1}^{h}\sum_{j=1}^{o} \mathbb{1}\{j = l^{(i)}\}\ln\left(\frac{\exp(\theta_j(v^{(i)}, f^{(i)}))}{\sum_{c=1}^{o}\exp(\theta_c(v^{(i)}, f^{(i)}))}\right) \tag{5}$$

where $\mathbb{1}\{j = l^{(i)}\}$ equals 1, if j equals $l^{(0)}$; otherwise, it is 0. Here, $\theta(\cdot)$ consists of multiple convolutional, batch normalization, max-pooling, SPP, and the fully-connected layers. To help readers who are not familiar with machine learning, we provide the details of each typical layer as follows.

Convolutional Layer: The key component of the convolutional layer is the convolution operation: $*$. Basically, this layer computes convolutions of the input with a series of filters, which can be mathematically described as follows [17]:

$$\phi_g^{\zeta} = \sum_{u\in U} x_{g-1}^{u} * W_g^{\zeta} + b_g^{\zeta} \tag{6}$$

where, ϕ_g^{ζ} is the latent representation of the ζ'th feature map of the g'th layer (the first feature map is the MTF-based graph $\{v^{(i)}, f^{(i)}\}$); x_{g-1}^{u} is the u'th feature map of the previous layer

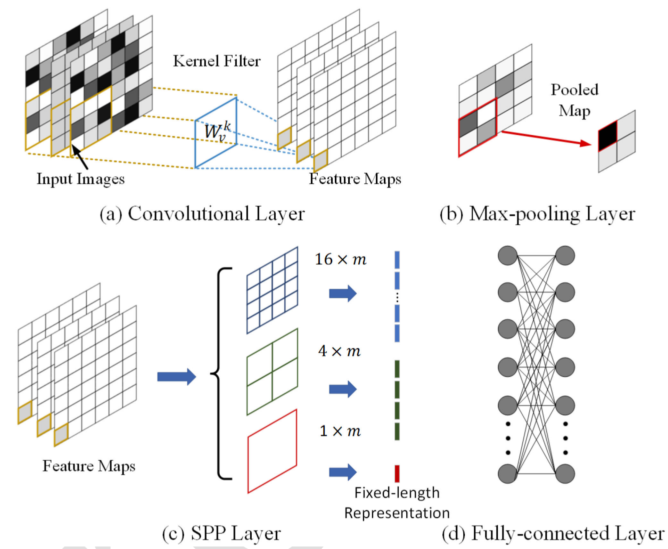


Fig. 6. Illustrate of the different layers in the proposed model; (a) Convolutional Layer; (b) Max-Pooling Layer; (c) SPP Layer (d) Fully-Connected Layer.

and U is the total number of feature maps; W_g^{ζ} and b_g^{ζ} are the kernel filter and the bias of the ζ'th feature map of the g'th layer, respectively. Since all event signals have been regarded as 2-dimensional MTF-based graphs after the feature reformulation, $x_{g-1}^{u} * W_g^{\zeta}$ can be written as [18],

$$(x_{g-1}^{u} * W_g^{\zeta})(i,j) = \sum_{\delta_i=0}^{U-1}\sum_{\delta_j=0}^{U-1} x_{g-1}^{s}(i-\delta_i, j-\delta_j)W_g^{\zeta}(i,j) \tag{7}$$

where, i and j are the row and column indices of the MTF-based graphs. Hence, the convolutional layer operates in a sliding-window way to output the feature maps (see Fig. 6(a)) [19]. The amount of horizontal and vertical movement in the sliding-window is set to 1 here. For each convolutional layer, the size of the output feature map is $\phi_g^{\zeta} \in \mathbb{R}^{(p-q+1)\times(p-q+1)}$ where x_{g-1}^{u} and W_g^{ζ} are $p\times p$ and $q\times q$ matrices, respectively. A typical drawback of the convolutional layer is that the impact of the data samples located on the border of data graph is much smaller than those at the center. To tackle this, a *padding strategy* is used by adding an additional layer to the border of the feature maps [20].

Activation Layer: To make up for the limitation of linear modeling in the convolutional layer, the outcomes of g'th convolutional layer are passed to an activation layer. A nonlinear function, such as sigmoid, hyperbolic tangent, or rectified linear unit (ReLU), is utilized to introduce nonlinearity to the model [18]. In this work, ReLU is used for all activation layers except for the output layer, as follows:

$$\phi_g^{\zeta} = \max(0, \phi_g^{\zeta}). \tag{8}$$

Unlike sigmoid and hyperbolic tangent activation functions, ReLU is robust to the vanishing gradient, thus, allowing deep models to learn faster and perform better [18].

Batch Normalization Layer: A batch normalization layer is added after the activation layer to avoid *internal covariate shift*, which leads to an exponential increase in computation burden by requiring much lower learning rates [21]. Thus, the output of each activation layer is normalized by subtracting the batch mean and dividing by the batch standard deviation for each training mini-batch.

Max-pooling Layer: After batch normalization, a max-pooling layer is utilized to summarize feature maps. Max-pooling can be considered as a sample-based discretization procedure that takes the feature map from the previous layer: $\phi_g^\zeta \in \mathbb{R}^{N_{\text{in}} \times N_{\text{in}}}$ and outputs a smaller matrix, denoted as $N_{\text{out}} \times N_{\text{out}}$. This is achieved by dividing the input matrix into N_{out}^2 pooling regions $P_{i,j}$ and selecting the maximum value [22]:

$$P_{i,j} \subset \{1, 2, \ldots, N_{\text{in}}\}^2, \forall (i, j) \in \{1, 2, \ldots, N_{\text{out}}\}^2. \quad (9)$$

In our work, a 2×2 max-pooling is used, as shown in Fig. 6 (b). Thus, $N_{\text{in}} = 2N_{\text{out}}$ and $P_{i,j} = \{2i - 1, 2i\} \times \{2j - 1, 2j\}$. The functions of the max-pooling layer generalize the results from the convolutional-normalization operation and reduce the model complexity while alleviating overfitting.

SPP Layer: In the proposed model, an SPP layer is adopted to replace the last max-pooling layer for mitigating the fixed-size constraint of the proposed model [19]. Unlike the standard pooling operation, such as max-pooling layer, which performs a single pooling operation, the SPP layer maintains spatial information by pooling in local spatial bins, as shown in Fig. 6(c). This figure provides an exemplary 3-level SPP layer. Suppose the last convolutional layer has r feature maps. In the first level, one bin is utilized to pool each feature map to become one value, thus, forming an r-dimensional vector. Then, four bins are leveraged to divide each feature map into 4 regions of equal size with a rectangular shape. The max-pooling strategy is applied to each region to form a $4 \times r$-dimensional vector. In the final level, each feature map is pooled to have 16 values, and form a $16 \times r$-dimensional vector. In general, the outputs of the SPP are $r \cdot B$-dimensional vectors, where B is the number of spatial bins, which is proportional to the MTF-graph size but is fixed. Basically, the SPP layer pools the features and generates fixed-dimensional outputs, which are then fed in to the last fully-connected layer. Hence, after the event identification is trained using the historical data and the corresponding event labels in offline, when PMU data quality problems (i.e., bad and missing data) occur in online, the related data points can be marked and then directly excluded. The remaining good-quality-data of arbitrary dimension can be assigned as the input of the proposed method. Moreover, while the conventional pooling operations use only a single window size, SPP utilizes multi-level spatial bins, which shows the better performance [23].

Fully-connected Layer: The last layer of the proposed method is a fully-connected layer, which integrates information across all locations in all the feature maps from the SPP layer. In this fully-connected layer, the softmax activation function is applied to calculate probabilities of the input being in a particular event type.

In the proposed SPP-aided CNN-based method, the adaptive moment estimation (Adam) algorithm is used to update the weight and bias variables [24]. Adam is a combination of gradient descent with momentum and root mean square propagation algorithms. Compared to backpropagation algorithms with constant learning rates (i.e., stochastic gradient descent), Adam computes individual adaptive learning rates for each parameter from estimates of first and second moments of the gradients [24], which significantly increases the training speed. To calibrate the hyperparameters of the proposed method, we utilize the random search method to find the appropriate sets [25]. It should be noted that the training procedure of the proposed model is an offline process. Hence, the high computational burden of the random search method does not impact the real-time performance of our event identification model. Moreover, the dropout strategy is utilized in our model to further reduce the risk of overfitting.

V. Numerical Results

To validate the effectiveness of the proposed event identification method, we test it on the PMU dataset and the related event log from interconnection B. This includes around 4800 known events that consist of line outage, XFMR outage, and frequency event. Moreover, the same number of the 2-second analysis-window in normal conditions have been added. Since each event type was left entirely up to data providers and we do not make any changes to the event log, the recorded line and XFRM trip categories in interconnection B cannot be determined as faults based on the current high-level description of the event logs. Hence, fault is not added as an event type in this work. We are trying to negotiate about the more detailed information of events with the data providers. The future work will be done to meet the gap once we acquire this information.

To ensure the generalization ability of the proposed method, it is necessary to observe whether the trained model suffers from an overfitting problem. To facilitate a better understanding, we provide a simple explanation about the overfitting problem. Overfitting refers to a method that can only model the training data well. In other words, if a model is highly customized for a specific training dataset, this model should suffer from a severe overfitting problem. Hence, in this work, the event dataset is randomly divided into two separate subsets for training (80% of the total data) and testing (20% of the total data). Moreover, to make the testing procedure more rigorous which can demonstrate the proposed model works well on unforeseen PMU data, we have applied k-fold cross validation strategy and k is selected as 5 in this work. The k-fold cross-validation strategy is performed in a rolling-horizon manner with a sliding window of PMU data. Specifically, the whole dataset is partitioned into k disjoint folds and $k - 1$ folds are utilized for model development and the remaining fold is used to validate the accuracy of the trained model. This procedure is repeated until each of the k folds has served for model validation. Then, the final accuracy of the proposed model is obtained based on k-time model validations. In other words, all data in the available dataset have been treated as the unseen data for calculating the final accuracy of our model. Based on the difference between the average training and testing accuracy, we can determine whether the overfitting problem arises. The case study is conducted on a standard PC with an Intel(R) Xeon(R) CPU running at 4.10GHZ and with 64.0 GB

TABLE II
THE STRUCTURE OF THE SPP-AIDED CNN-BASED MODEL

Layout	Type	Output Shape	Model Complexity
1/1	Conv2D	(120,120,32)	608
1/2	Activation	(120,120,32)	0
1/3	Batch Norm	(120,120,32)	128
2/1	Conv2D	(120,120,32)	9k
2/2	Activation	(120,120,32)	0
2/3	Batch Norm	(120,120,32)	128
2/4	Max-pooling	(60,60,32)	0
3/1	Conv2D	(60,60,64)	18k
3/2	Activation	(60,60,64)	0
3/3	Batch Norm	(60,60,64)	256
4/1	Conv2D	(60,60,64)	36k
4/2	Activation	(60,60,64)	0
4/3	Batch Norm	(60,60,64)	256
4/4	Max-pooling	(30,30,64)	0
4/5	Dropout	(30,30,64)	0
5/1	Conv2D	(30,30,128)	73k
5/2	Activation	(30,30,128)	0
5/3	Batch Norm	(30,30,128)	512
6/1	Conv2D	(30,30,128)	147k
6/2	Activation	(30,30,128)	0
6/3	Batch Norm	(30,30,128)	512
6/4	Max-pooling	(15,15,128)	0
6/5	Dropout	(30,30,64)	0
6/6	SPP	(1,2688)	0
7/1	Fully-connected	(1,1,5)	13k
7/2	Activation	(1,1,5)	0

of RAM and an Nvidia Geforce GTX 1080ti 11.0 GB GPU. The training computation time of the proposed model is around a few hours. However, since the training procedure is an offline process, the high computation burden of the training procedure does not impact the real-time performance of our event identification model. After the model is trained, we have tested the average testing time based on 5000 Monte Carlo simulations. In this work, the average testing time is around 1.4 ms. Even including the communication delays, our model is feasible in real-time, in accordance with the IEEE C37.118.2-2011 standard.

A. *Performance of the Proposed Method*

The detailed structure of the proposed classifier is presented in Table II. As can be seen, our model is a seven-layer architecture that includes multiple convolutional, activation, batch normalization, SPP, and fully-connected layers. Each row represents layers with specific layer type, the dimension of output feature map, and model complexity calculated with the number of hyperparameters. Based on this structure, the training and testing performances of the proposed method are shown in Fig. 7. As demonstrated in this figure, the training and testing accuracy converge to around 95.1% and 94.6%. The difference between these two values is small, which proves the generalization ability of the proposed model.

Moreover, the performance of the proposed method for each event type is explained using confusion matrix shown in Fig. 8. In this figure, the rows correspond to the estimated type and the columns correspond to the true type. The diagonal and off-diagonal cells correspond to events that are correctly and

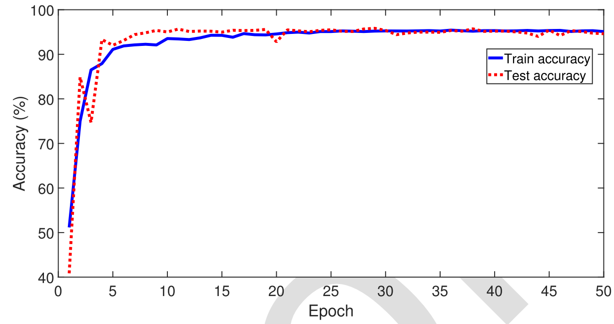


Fig. 7. Training and testing results of the proposed model.

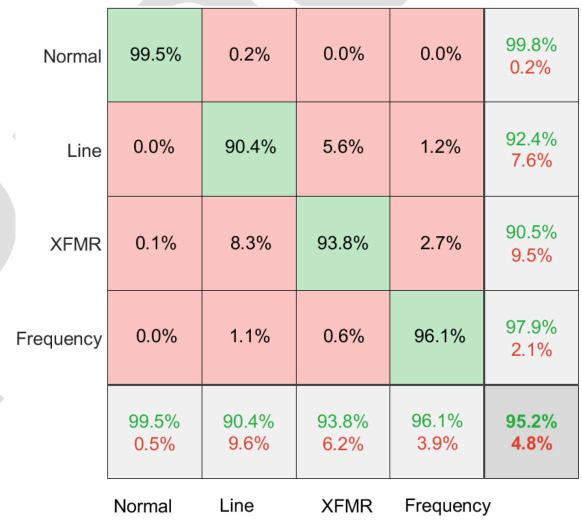


Fig. 8. Confusion matrix for interconnection B using the proposed model.

incorrectly classified, respectively. The value of each cell represents the accuracy of the specific event type. Here, two statistical indexes are utilized: the precision and the recall rates are presented in the cells on the far right and the bottom of the figure, respectively [12]. The cell in the bottom right of the figure is the overall accuracy. As seen in this figure, the worst-case precision and recall rates of the proposed method are around 90.5% and 90.4% for the XFRM outage and line outage classes, which still are acceptable values. It can be observed that the accuracy of the proposed method for the XFRM outage and line outage events is relatively lower than the rest. One possible reason is that the event patterns of these two types of events are some similarities, which is described in the confusion matrix (see Fig. 8). Around 8.3% of line outage events are inaccurately deemed to be XFRM outage events. As shown in the figure, the false positive rate (system is inferred to have an event while its actually state is normal operation) is pretty low, meaning that our model is extremely unlikely to provide inaccurate identification in the normal operation. When an event occurs, in more than 90% of cases, our model will provide an accurate event identification. Besides, in more than 99% of cases, our model will at least provide a meaningful event warning for system operators, which is important in emergency situations. In contrast, the false negative rate (system is inferred to be operating

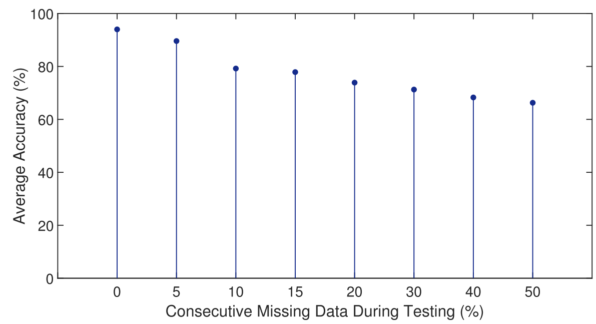


Fig. 9. Sensitivity of event identification accuracy to the size of consecutive bad/missing data.

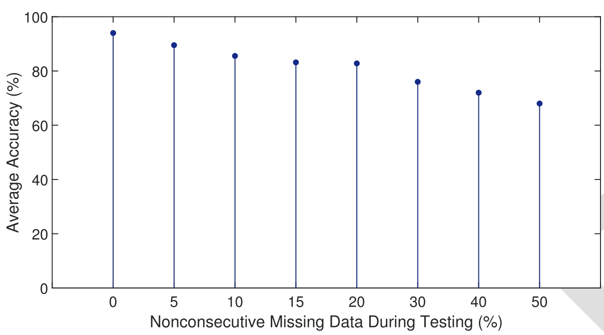


Fig. 10. Sensitivity of event identification accuracy to the size of nonconsecutive bad/missing data.

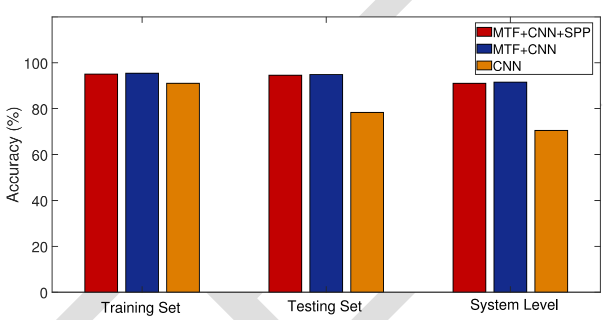


Fig. 11. The performance of MTF feature extraction and SPP layer.

normally, while its actual status is that an event has occurred.) is only around 0.5%.

In practice, operators are interested in knowing a single system-level classification outcome rather than multiple PMU-level outcomes. Hence, we have obtained and tested the system-level results by collecting the classification outcomes of all PMUs: for a specific event, if more than 90% of PMU-level outcomes are positive, the event is identified at the system-level, using the proposed method. In this case, the system-level accuracy of our technique is around 91.07%.

Considering that the proposed method is composed of three components: MTF, SPP, and CNN, we have tested the event identification accuracy for each component, as shown in Fig. 11. It is observed that the model that only includes MTF and CNN achieves similar accuracy with the proposed model. This indicates that the SPP strategy does not impact the identification performance; however, SPP is needed for resolving online data quality issues. Further, we compare the accuracy obtained by

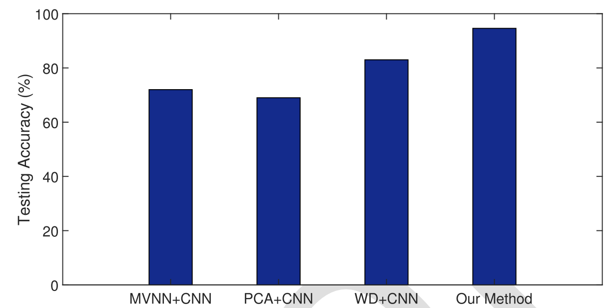


Fig. 12. Comparison results of four feature extraction methods.

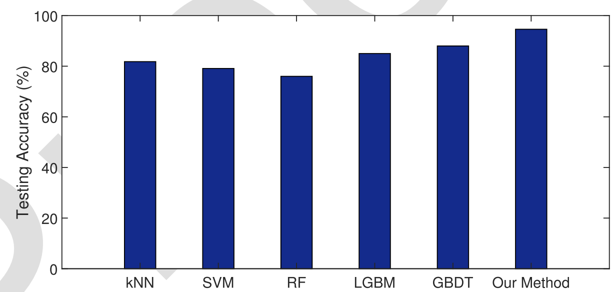


Fig. 13. Comparison results of six event identification models.

sending the PMU data before and after the MTF-based feature extraction to the model respectively. As described in the figure, utilizing the MTF-based feature extraction model, identification accuracy has been increased a lot. This result proves that the MTF-based feature extraction is valuable and can improve the identification accuracy. Moreover, to further evaluate the performance of the MTF, we have conducted numerical comparisons with several commonly-used feature extraction techniques for PMU data, PCA, wavelet transformation, and multidimensional minimum volume enclosing ellipsoid [11], [15], [26]. The result is shown in Fig. 12. To ensure a fair comparison between the four feature extraction methods, CNN is utilized to perform event identification for all feature extraction methods. It is observed that through the Markov-based feature extraction, the accuracy of event identification can be considerably improved.

B. Method Comparison

We have conducted numerical comparisons with three previous event identification models: k-nearest neighbors (kNN) [27], support vector machine (SVM) [10], and random forest (RF) [28]. Further, two state-of-the-art classification methods, light gradient boosting machine (LGBM), and gradient boosting decision tree (GBDT), have also been compared with our methods in terms of event identification accuracy [29]. As described in Fig. 13, the testing accuracy of the proposed method is around 94%. On the other hand, kNN, SVM, RF, LGBM, and GBDT, show the testing accuracy of $\{81.8, 79.1, 76.7, 85.3, 88.1\}$, respectively. Hence, based on this PMU dataset, the proposed method shows a better accuracy for event identification compared to the previous works.

C. Sensitivity Analysis

To demonstrate the sensitivity of the proposed event identification accuracy to the size of missing data, we have calculated the average performance of our method under various sizes of missing/bad data. For each percentage of missing/bad data, 1000 Monte Carlo simulations are conducted to obtain the average accuracy. Here, we consider two different data quality issues: consecutive and nonconsecutive data quality issue. In real-time event identification, consecutive data quality issue is more challenging compared to nonconsecutive data quality issue. The reason is that data with the nonconsecutive data quality issue can keep a portion of the critical information (i.e., event patterns). This information can still be used for accurate event identification. For the consecutive data quality issue, it is likely that all event information is lost within a time period. As the length of consecutive data quality issue increases, the probability for loss of event information significantly increases. Hence, we can expect performance degradation with the increase of consecutive missing/bad data. For each experiment, we have randomly selected a time-stamp as the start time for the data quality issue. Then, n consecutive data points after this time-stamp are removed, where n is determined by the percentage of bad/missing data. Here, we gradually increase n from 0 to 50% of the data samples. The result is shown in Fig. 9. As is presented in the figure, the model accuracy drops as the percentage of missing data increases from 0% to 20%. This result is expected. It is clear that no event identification model can provide a good estimate when event information is missing. Then, when n continues to increase to 50% of the data sample, the accuracy of the proposed model is stabilized around 65%. These results demonstrate that the proposed learning-based method can still provide meaningful results with 50% data loss. Note that the 50% consecutive bad/missing data is an extremely rare case.

Moreover, we have tested the robustness of our method for nonconsecutive data quality issues. In each experiment, we have randomly selected n independent data points as the missing/bad data points. The result is shown in Fig. 10. It can be observed that the nonconsecutive data quality issues are much easier to handle using the proposed method. As is described in the figure, the model accuracy slightly drops as the percentage of missing data increases from 0% to 20%. Even if 20% of the data points are treated as nonconsecutive missing data points, the proposed model can still reach an average accuracy of 83%. It should be noted that in practice most of the data quality issues are nonconsecutive. The consecutive data quality issue can be considered as the worst-case scenario. Last but not least, unlike the previous data-driven methods that rely on data imputation techniques to introduce robustness, our solution addresses online data quality issues by eliminating the fixed-size input constraint of the learning process. In comparison, our method can handle consecutive data quality issues without any additional computational burden in real-time. Meanwhile, based on discussions with our industry partners, many system operators avoid performing data imputation techniques for PMU data since they prefer not to modify the PMU data. Hence, our method provides a good solution for system operators to deal with online data quality issues, especially for consecutive data quality issues.

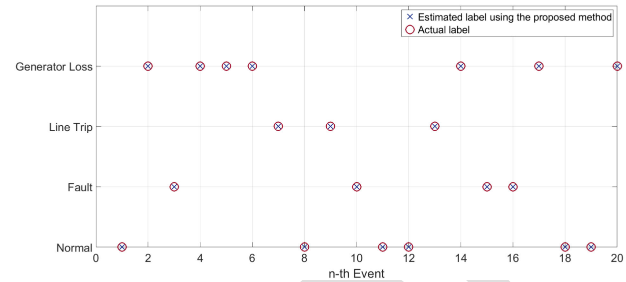


Fig. 14. Comparison of estimated event type and actual event type in ACTIVSg500 case study.

D. ACTIVSg500 Test Case

This subsection further explores the performance of the proposed method using a benchmark synthetic power system with artificial PMU data generated by simulated events. Specifically, this synthetic PMU dataset is generated by the Siemens Power System Simulator for Engineering (PSS/E). The Illinois 500-bus system, known as the ACTIVSg500 test case, is utilized to demonstrate the results. The detailed description and parameters of this power system can be obtained from [30]. To be consistent with the available real-world PMU dataset, the sampling rate of PMUs is set to be 60 recordings per second. PMUs are placed at buses 22, 66, 187, 308, and 500 to record voltage phasor and frequency. Three types of events described are simulated: line fault events, line trip events, and generator loss events. More precisely, we have simulated 350 events, including 150 line fault events, 150 line trip events, and 50 generator trip events at different locations with the same pre-event system condition. We have applied the aforementioned strategy to obtain the training and testing data. In this case study, the average testing accuracy converges to about 98.7%. Fig. 14 shows the estimated and actual labels for 20 events. As can be seen, the proposed method is able to accurately classify the event types. From a statistical perspective, based on this synthetic PMU dataset, the proposed method offers classification accuracy of 100% for line fault, 97% for line trip, and 98.2% for generator trip. Also, the Area under the Curve (AUC) index is employed to assess the classification performance of our method [31]. In this case, the proposed SPP-aided CNN-based method achieves an AUC value of 0.98. The comprehensive case study including the real-world dataset and the synthetic dataset helps to demonstrate the generalization of the proposed approach.

E. Cost of Misclassification

In this subsection, we analysis the cost of misclassification caused by the proposed model. It should be noted that we have developed a data-driven event analyzer rather than a protection module. The goal of our data-driven model is to enhance situational awareness by identifying system vulnerabilities (i.e., relay misoperations) in real-time. Hence, in normal operation, data-driven event identification models are treated as supervisory monitoring, which will not provide input to digital relays and or interfere with relay operation. In the worst-case scenario, if the trained model provides an incorrect estimation, the relay protection will still operate despite the loss of selectivity [10]. When SCADA is dysfunctional, as was the case during the 2003 North American large-scale blackout, data-driven models will still work, thus maintaining partial system awareness. Such strategies can reduce the risk of misclassification caused by the proposed model (i.e., inadvertent operations). Moreover, our method introduces robustness against data quality issues in real-time operation, which prevents the misclassification caused by missing and bad data. Since the historical relay operations are not available, we cannot exactly quantify the cost of misclassification at this stage. We leave it for future work once they are available. More comprehensive results will be provided.

VI. Conclusion

In this paper, we have presented a novel two-stage learning-based method for real-time event identification to enhance the situational awareness of power systems using PMU data. Comparisons with previous methods show that our method achieves more accurate event identification outcomes. Moreover, this approach shows robustness against data quality problems in online operation, which highly improves the practical applicability in real-world systems. The proposed method is successfully validated on a large-scale PMU dataset and the real event logs.

Acknowledgment and Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] D. White, A. Roschelle, P. Peterson, D. Schlissel, B. Biewald, and W. Steinhurst, "The 2003 blackout: Solutions that won't cost a fortune," *The Electricity J.*, vol. 16, no. 9, pp. 43–53, 2003.

[2] W. Ju *et al.*, "Real-time monitoring of area angles with synchrophasor measurements," 2020, *arXiv:2003.06476v1*.

[3] 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.

[4] M. Cui, J. Wang, J. Tan, A. R. Florita, and Y. Zhang, "A novel event detection method using pmu data with high precision," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 454–466, 2019.

[5] Y. Ge, A. J. Flueck, D. K. Kim, J. B. Ahn, J. D. Lee, and D. Y. Kwon, "Power system real-time event detection and associated data archival reduction based on synchrophasors," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 2088–2097, 2015.

[6] E. Perez and J. Barros, "A proposal for on-line detection and classification of voltage events in power systems," *IEEE Trans. Power Deli.*, vol. 23, no. 4, pp. 2132–2138, Oct. 2008.

[7] M. Biswal, S. M. Brahma, and H. Cao, "Supervisory protection and automated event diagnosis using pmu data," *IEEE Trans. Power Deli.*, vol. 31, no. 4, pp. 1855–1863, 2016.

[8] S. Liu *et al.*, "Data-driven event detection of power systems based on unequal-interval reduction of pmu data and local outlier factor," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1630–1643, Mar. 2020.

[9] J. Ma, Y. V. Makarov, R. Diao, P. V. Etingov, J. E. Dagle, and E. D. Tuglie, "The characteristic ellipsoid methodology and its application in power systems," *IEEE Trans. Power Syst.*, vol. 4, no. 27, pp. 2206–2214, May 2012.

[10] S. Brahma, R. Kavasseri, H. Cao, N. R. Chaudhuri, T. Alexopoulos, and Y. Cui, "Real-time identification of dynamic events in power systems using pmu data, ~~and~~ potential applications-models, promises, and challenges," *IEEE Trans. Power Deli.*, vol. 32, no. 1, pp. 294–301, Feb. 2017.

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

[12] N. Sokolova, M. Japkowicz and S. Szpakowicz, *Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[13] W. Ju *et al.*, "Real-time area angle monitoring using synchrophasors: A practical framework and utility deployment," *IEEE Trans. Smart Grid*, pp. 1–1, Jan. 2020.

[14] Z. Wang and T. Oates, "Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks," *Association~~ for the~~ Advanc~~ement~~~~ of~~ Artificial Intelli~~gence~~*, pp. 40–46, 2015.

[15] W. Gao and J. Ning, "Wavelet-based disturbance analysis for power system wide-area monitoring," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 121–130, 2011.

[16] A. S. Campanharo, M. I. Sirer, R. D. Malmgren, F. M. Ramos, and L. A. N. Amaral, "Duality between time series and networks," *PLoS one*, vol. 6, no. 8, pp. 40–46, 2011.

[17] T. V, C. E, and L. A, "A deep convolutional auto-encoder with pooling-unpooling layers in caffe," 2017, *arXiv:1701.04949*.

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.

[20] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 12, pp. 2263–2276, Dec. 2016.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*.

[22] B. Graham, "Fractional max-pooling," 2014, *arXiv:1412.6071*.

[23] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *IEEE Comput Soc Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 2169–2178, 2006.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[25] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[26] H. Li, Y. Weng, E. Farantatos, and M. Patel, "An unsupervised learning framework for event detection, type identification and localization using pmus without any historical labels," *IEEE Power Energy Soc. Gen. Meeting*. pp. 1–5, 2019.

[27] O. P. Dahal, H. Cao, S. Brahma, and R. Kavasseri, "Evaluating performance of classifiers for supervisory protection using disturbance data from phasor measurement units," *IEEE PES Innov. Smart Grid Technol., Europe*, pp. 1–6, 2014.

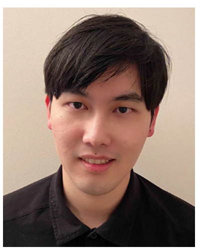
[28] D. W. X. Wang, Y. Zhang, and L. Jin, "Detection of power grid disturbances and cyber-attacks based on machine learning," *J. Inf. Secur. Appl.*, vol. 46, pp. 42–52, 2019.

[29] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inform~~ation~~ Process. Syst.*, pp. 3146–3154, 2017.

[30] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3258–3265, 2017.

[31] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve," *Radiol.*, vol. 143, no. 1, pp. 29–36, Apr. 1982.

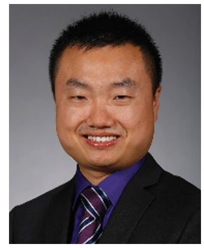
Yuxuan Yuan (Graduate Student Member, IEEE) received the B.S. degree in 2017 in electrical and computer engineering from Iowa State University, Ames, IA, USA, where he is currently working toward the Ph.D. degree. His research interests include distribution system state estimation, synthetic networks, data analytics, and machine learning.



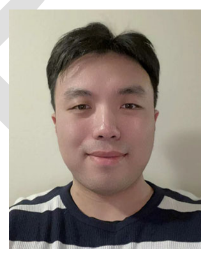
Yifei Guo (Member, IEEE) received the B.E. and Ph.D. degrees in electrical engineering from Shandong University, Jinan, China, in 2014 and 2019, respectively. He is currently a Postdoctoral Research Associate with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. During 2017–2018, he was a Visiting Student with the Department of Electrical Engineering, Technical University of Denmark, Lyngby, Denmark.

His research interests include voltage or var control, renewable energy integration, wind farm control, distribution system optimization and control, and power system protection.

Kaveh Dehghanpour received the B.Sc. and M.S. degrees in electrical and computer engineering from the University of Tehran, Tehran, Iran, in 2011 and 2013, respectively, and the Ph.D. degree in electrical engineering from Montana State University, Bozeman, MT, USA, in 2017. He is currently a Postdoctoral Research Associate with Iowa State University, Ames, IA, USA. His research interests include application of machine learning and data-driven techniques in power system monitoring and control.



Zhaoyu Wang (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, China, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA. He is the Harpole-Pentair Assistant Professor with Iowa State University, Ames, IA, USA. His research interests include optimization and data analytics in power distribution systems and microgrids. He is the Principal Investigator for a multitude of projects focused on these topics and funded by the National Science Foundation, the Department of Energy, National Laboratories, PSERC, and Iowa Economic Development Authority. He is the Chair of IEEE Power and Energy Society (PES) PSOPE Award Subcommittee, the Co-Vice Chair of PES Distribution System Operation and Planning Subcommittee, and the Vice Chair of PES Task Force on Advances in Natural Disaster Mitigation Methods. He is the Editor of the IEEE TRANSACTIONS ON POWER SYSTEMS, IEEE TRANSACTIONS ON SMART GRID, IEEE OPEN ACCESS JOURNAL OF POWER AND ENERGY, IEEE POWER ENGINEERING LETTERS, and *IET Smart Grid*. He was the recipient of the National Science Foundation (NSF) CAREER Award, the IEEE PES Outstanding Young Engineer Award, and the Harpole-Pentair Young Faculty Award Endowment.



Yanchao Wang received the Bachelor of Engineering degree in optical information and technology from the Beijing Institute of Technology, Beijing, China, in 2014. He is currently working toward the Ph.D. degree with Iowa State University, Ames, IA, USA. His research interests include deep learning in power systems, machine learning, and signal processing