Received 21 April 2023; revised 4 June 2023; accepted 25 June 2023. Date of publication 7 July 2023; date of current version 4 August 2023. Digital Object Identifier 10.1109/OJCOMS.2023.3292050

Artificial Intelligence-Assisted Edge Computing for Wide Area Monitoring

BIN HU^(D) (Senior Member, IEEE), AND HAMID GHARAVI^(D) (Life Fellow, IEEE)

Smart Connected Systems Division of Communication Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA CORRESPONDING AUTHOR: B. HU (e-mail: bhu@nist.gov)

ABSTRACT The massive digital information generated in conjunction with the ever-increasing phasor measurement data in the power grid has led to a tremendous constraint on the analysis and timely processing of real-time data. Under these conditions, leveraging Artificial Intelligence (AI) can play a crucial role in assisting more efficient data processing and analysis. In this paper an AI-assisted power grid event classification method is proposed, which aims at improving the overall power grid system performance. Furthermore, an edge cloud sharing scheme is introduced for a large-scale power grid system. To balance the load and reduce the maximum processing time, a multiple edge cloud node-based scheme is developed. The simulation results verify that the proposed AI-assisted event classification method, together with the edge cloud sharing scheme, can significantly improve the overall performance of the system.

INDEX TERMS AlexNet, classification and identification, CNN, edge cloud, GRU, LeNet, LSTM, machine learning, smart grid, SVM.

I. INTRODUCTION

7 ITH the increasing deployment of phasor measurement units (PMUs), smart meters, and Internet of Things (IoT) devices, together with the ever-increasing deployment of renewable and sustainable energy technologies, managing a huge amount of generated data is becoming a major challenge. Generally, the collected data is mostly unstructured and can be in various formats. This consequently complicates data processing and analysis. For instance, traditional power system analysis and management is primarily based on physical modeling and numerical calculations, which cannot meet the requirements of smart girds due to the increasing complexity, high uncertainty, and huge volume of digital information. On the other hand, the advancement of computer power and the recent development of complex AI algorithms as an enabling technology can be utilized to effectively assist the processing of such massive data

In the past decade, a great deal of work has been done on the topic of applying AI techniques to the smart grid. Deep Learning (DL), Reinforcement Learning (RL), and Deep Reinforcement Learning (DRL) are three widely used AI techniques. These algorithms can be used to ensure secure and stable operation in uncertain and complex environments.

Data driven DL and RL algorithms can play a crucial role in processing massive digital information for power grid event detection, identification, and prediction [1]. DL algorithms can be deployed to extract valuable features from collected data that can be used to obtain insights about the power system's state and dynamic behavior. For instance, ambient signals collected from the large number of PMUs installed throughout the power grid system can be exploited to detect disturbances as early as possible. In [2], Multilayer Perceptions (MPs), Deep Belief Networks (DBNs), and Convolutional Neural Networks (CNNs) are used to classify disturbances by processing PMU measurements data collected from the power grid system. A DBN based transient stability assessment method was proposed in [3] to separate stable cases from unstable cases. In [4], CNN was used to extract features during the transient process. CNN is shown to be an effective approach in classifying faults, such as line faults [5], [6]. For instance, a deep CNN architecture [7] is used to classify power line insulators based on aerial images. The authors in [8] demonstrate the effectiveness of a CNNbased approach in forecasting probabilistic wind power. For fault classification these methods also achieve good performance [4], [9]. For fault detection, [10] offers three CNN-based models, which are capable of handling big data generated from multiple PMUs and include measurements such as voltage, current, frequency, etc.

To ensure reliable and stable operation of a power grid, it is essential to detect and predict faults/defects in the early stages. To achieve this, DL algorithms can be utilized to analyze gradual changes caused by faults/defects. In [7], a high-level discriminative Convolutional Neural Network (CNNs) is proposed to extract features of the insulators experiencing gradual changes. For fault detection and prediction, Artificial Neural Networks (ANNs) based methods have been proposed in [11], [12], [13]. Nonetheless, ANN based methods may not be able to extract temporal information during disturbances, which is an essential requirement for fault prediction. Conversely, Recurrent Neural Networks (RNNs), which have a proven record of extracting hidden features in big data for image captioning, voice conversion, and language processing [14], [15], [16], [17], [18], [19] have been shown to achieve a good performance in detecting faults [20], [21]. However, conventional RNN normally suffers from vanishing gradient, which degrades the performance of capturing temporal features in a long-time span. Long short-term memory (LSTM) networks proposed in [22] can effectively overcome this problem. An LSTM based approach is presented in [23] to diagnose and predict faults in complicated scenarios. In [24], an LSTM network is utilized to timely detect and identify faults based on measurement data. The authors in [25] use LSTM networks to capture the temporal features caused by line trip faults, which is a gradual process. As an improved RNN, the LSTM network can achieve a better performance with longer time series. The recently introduced Gated Recurrent Unit (GRU) is another RNN algorithm, which can also perform well [26]. As opposed to LSTM, where 3 distinct gates are used, GRU reduces the gating signals to two, namely an update gate and a reset gate.

Support vector machine (SVM) based schemes [27] are another effective classifier used in power grid systems and are considered one of the most robust classification models. In [28], [29] SVM-based schemes have been used for online transient stability assessment. In addition, [30] proposes an SVM-based algorithm to lower misclassification rates for voltage stability assessment. In [31], SVM has also been considered for fault detection and islanding. However, we should point out that the SVM's computational complexity is decided by the number of support vectors, instead of the dimension of the sample space. This consequently results in low computational complexity and good robustness. More importantly, SVM can achieve global optimal solution [27].

In this paper, a power grid events classification method is proposed to process various power events by integrating multiple machine learning schemes. Specifically, we first use SVM to quickly identify the event type. For example, those that are categorized as disturbances and expected to occur gradually, are sent to LSTM/GRU algorithm for identification, while faults are processed by using CNN algorithms. In addition, processing the massive data generated in a large-scale power grid network at the remote-control center can impact the latency, as well as impose a heavy load on the communication network. Multiple small-base-stations can be used as edge cloud nodes to overcome these drawbacks [32]. However, a single edge cloud node can only serve an area with limited range. Local traffic activities, which depend on the occurrence of power grid events in the domain of each edge cloud, may be different and unbalanced in terms of not only the computation, but also the amount of data needs to be reported to the remote cloud [33]. Furthermore, the complexity of the employed machine learning algorithms varies widely, leading to an imbalance and shortage of computing and networking resources in edge cloud nodes. To share computing resources and reduce processing delays, in this paper an edge cloud sharing mechanism, which includes an efficient bandwidth allocation strategy is proposed. Under the proposed scheme edge nodes that are experiencing the large number of events will be able to intelligently cooperate with the remote cloud for processing and real-time transmission of synchrophasor data generated by many PMUs that are placed at sensitive locations throughout the grid network. Their main function is to measure phase and voltage variations. GPS is mostly used to synchronize PMUs using a sampling clock, which is phase locked to one pulse per second (PPS). The basic idea of sampling electric waveforms using GPS is to examine phase and amplitude uncertainties, which can affect the fundamental frequency. For example, monitoring transmission lines using PMUs can give insights into power system phenomena like loss of stability, faults, and load encroachments. PMU data streaming is built on top of the user datagram protocol over Internet protocol (UDP/IP) for transmission to the Phasor Data Collector (PDC) [34].

In this paper our main objective is to investigate the impact of machine learning on identifying various types of faults based on massive amounts of real-time data generated by PMUs, which are dispersed throughout grid networks. The contributions of this paper are summarized as follows:

- Different machine learning algorithms are integrated to classify and identify power events based on their characteristics and features. Compared with traditional schemes where only one machine learning algorithm is used to process all power events, our method indicates a significant improvement in system performance.
- 2) A collaborative cloud and edge computing scheme is proposed to balance load and reduce overall processing delay, by using an efficient and adaptive bandwidth allocation strategy. To the best of our knowledge, the proposed scheme is the first to use multiple machine learning algorithms and take into consideration their complexities for resource allocation. The simulation results demonstrate that our scheme can effectively reduce the processing delay of the system, which is a crucial factor in preventing a possible blackout.

The paper is organized as follows. After a brief overview of multiple machine learning algorithms, we propose a power grid event classification and identification method in Section II. In Section III, an AI-assisted edge cloud sharing mechanism is then introduced for a large-scale power grid system with multiple edge cloud nodes. To assess the performance of the proposed scheme, we present the simulation results in terms of latency and reliability in Section IV. This is followed by the conclusion in Section V.

II. INTEGRATED MACHINE LEARNING FOR POWER EVENTS CLASSIFICATION

How to effectively extract valuable information from the big data generated in the smart grid has attracted considerable attention in past decades. To improve system performance, machine learning techniques are considered to support classification and identification of power grid events. We present an AI-assisted event detection and classification scheme for a power grid system where different machine learning algorithms are used to improve the performance.

In smart grid, power quality events include transient disturbances, such as Generation Trip (GT), Load Shedding (LS), Oscillation (OS), Line Trip (LT), and various power system faults including Generator Fault (GF), Line Fault (LF), Bus Fault (BF), and Transformer Fault (TF). Disturbances are mainly due to a gradual process in a long-time span, which may be caused by aging, damaged distribution equipment, bad insulation, weather changes, etc. For example, a line trip fault is caused by a gradual process of distribution line resistance, which can lead to massive blackouts. The corresponding change of measurements can be captured by LSTM/GRU. LSTM/GRU based algorithms are well known for processing time series-based problems. On the other hand, faults like line faults can occur suddenly and don't have a gradual process. Under such conditions, LSTM is not suitable. Instead, CNN (LeNet/AlexNet) can be used, after re-shaping the input data to produce a matrix structure that resembles the frame of an image. CNN (LeNet/AlexNet) is well known for handling problems with some spatial correlation like images data.

Therefore, in the proposed method shown in Fig. 1, SVM is first used to categorize the event type. The SVM not only has the advantage of low computational complexity and good robustness, it is also able to achieve a global optimal solution. For instance, after fast identification of the event type, those that are categorized as disturbances are sent to LSTM/GRU algorithm for identification, while events that are categorized as faults are processed using the LeNet/AlexNet algorithm. The simulation results in Section IV show that the proposed events classification method can improve the overall performance by using difference machine learning algorithms.

Most machine learning algorithms are good at solving one kind of problem or processing one type of dataset. However, for a large amount of data with different characteristics and features, these algorithms are not able to realize the full potential of AI by working alone. That's why we propose the integrated machine learning scheme for power event classification. The proposed integrated machine learning could



FIGURE 1. The flowchart of the proposed power grid events classification method.

take advantages of every algorithm and makes them to work together to complement each other. This can effectively avoid their weaknesses when solving problems that alone they were not designed to solve.

III. EDGE CLOUD COMPUTING AND SHARING

With large scale deployment of terminals (such as PMUs, smart meters and IoT devices) in smart grid, processing a large amount of data can be achieved by handling data processing tasks at the edge of the network [35]. In the case of detecting and predicting power gird events, edge cloud nodes can use machine learning to locally categorize and identify the events. Under these conditions, only a small portion of data, such as inter-area fault locating and inter-area oscillation detection, will be sent to the remote cloud for further processing. This method of parallel processing and analysis [36] of data from massive terminals through machine learning based edge computing not only provides rapid responses, but also significantly reduces network overhead.

Specifically, under the proposed method power grid events can be largely detected and predicted promptly by edge cloud nodes. They can implement data storage locally and respond in real-time without uploading data. As shown in Fig. 2, power grid events will firstly be categorized into four different types of events: local disturbances, global disturbances, local faults, and global faults. In our approach the local disturbances and local faults will be processed locally by



FIGURE 2. The flowchart of the proposed Edge Computing based power grid events classification method.

edge computing, where the local disturbances are sent to the LSTM/GRU algorithm for identification, while the local faults are processed by using the LeNet/AlexNet algorithm. Global disturbances and faults will be uploaded to the remote cloud for classification.

A. COLLABORATIVE CLOUD AND EDGE COMPUTING

To meet the requirements for smart grid synchrophanetworks, 5G Ultra-Reliable and Low-Latency sor Communications (URLLC) can be considered as viable options [37]. Therefore, in this paper URLLC-Based wireless network architecture that consists of multiple micro base stations (MIBSs) each acting as an edge cloud node that can communicate with its associated macro base station (MABS) is considered. With the ever-increasing deployment of PMUs for power gird monitoring, each edge node may have to handle multiple PMUs within its coverage area. However, some of these nodes may experience many events such as disturbances and faults and may not have sufficient processing power to handle events categorization and detection in a timely manner. Under these conditions, cooperation and interaction among edge nodes and the remote cloud can be crucial. With the aid of machine learning algorithm, edge nodes should be able to collaboratively offload a part of data processing to the centralized



FIGURE 3. Collaborative Cloud and Edge Computing.

cloud via MABSs. This can be achieved by intelligently exchanging the network status among edge nodes and the remote cloud. In addition, the end-to-end transmission delay caused by unstable wireless links between MIBSs and MABSs, as well as the transmission between MABSs and cloud platform, can impact the transmission efficiency. In this paper, a collaborative cloud and edge computing scheme is proposed to share the load and minimize the end-to-end latency.

In the case of detecting and predicting power gird events, edge cloud nodes will first use SVM to quickly categorize the power grid events to different types, such as disturbances, faults, and normal operations. The events categorized as disturbances are sent to an LSTM-based algorithm for identification and those categorized as faults are processed by a CNN algorithm. Note that different algorithms have different complexities and those with high complexity will need more computing resources, including extra time to process the events. Furthermore, edge cloud nodes experiencing events categorized as normal operation will become idle until the occurrence of the next event. Therefore, it is expected that some edge cloud nodes remain idle without having to transmit any information [33]. Consequently, this will affect the load-balancing and severely undermine the reliability and latency performance of the entire power grid. Under these conditions, the collaborative cloud and edge computing scheme can be used to assign more channel capacity to busy edge cloud nodes and offload the processed data from busy edge cloud nodes to the remote cloud. The collaborate resource allocation aided multi-edge-cloud architecture is shown in Fig. 3, where each small base station (MIBS), acting as the edge cloud node (ECN), accommodates an uncertain number of PMU terminals. Each ECN can communicate, exchange resources and status information with MABSs and the remote cloud.

Based on the size of the processed data and complexity of the selected algorithm, we define the processing resource; *RS*, as:

$$RS = \frac{size \ of \ the \ processed \ data}{size \ of \ input \ vectors/matrix} \times the \ total \ parameter \ of \ the \ algorithm,$$
(1)

where the total parameter of the algorithm denotes the complexity of the used algorithm. Specifically, the complexity of SVM is O (N^3), where N is the data number of vectors [38]. The total number of parameters in an LSTM network can be expressed as,

$$P = 4n_c n_c + 4n_c n_i + n_c n_o + 3n_c,$$
(2)

where n_c is the number of memory cells (and the number of memory blocks in this case), n_i is the number of input units, and n_o is the number of output units. The total number of parameters in the GRU RNN can be expressed as,

$$P = 3\left(n^2 + nm + n\right),\tag{3}$$

where *n* and m are the sizes of hidden state and input vector, respectively. The total parameter of classic LeNet with an input image of $32 \times 32 \times 1$ is around 60k; The total parameter of AlexNet, with an input image of $227 \times 227 \times 3$, is around 60M. The integrated machine learning presented in Section II, presents our proposed collaborate approach, i.e., using multiple machine learning schemes is capable of processing different types of power events. Their complexities and sizes of input vector/matrix are taken into consideration when allocating computing resources and bandwidth for balance-sharing [32]. This makes our scheme unique and more efficient compared with traditional collaborative schemes, such as the one in [39] where only one machine learning algorithm is considered.

To balance the loads of multiple ECNs, an edge-cloud selection algorithm is described as follows: The processing resource used by every edge cloud in a given time period can be defined as,

$$\mathbf{RS} = \{RS_1, RS_2, \dots, RS_E\},\tag{4}$$

where E represents the total number of ECNs. This means that the processing resource by all ECNs is,

$$RS_C = \sum_{i=1}^{E} RS_i,$$
(5)

while the average processing resource is:

$$RS_{average} = RS_C/E.$$
 (6)

At the beginning of a time period, each ECN selects suitable machine learning algorithms for occurrences of power events within its domain, by which it calculates the needed RS_i . After exchanging information with MABSs, each ECN can obtain the RS_i of all ECNs and then decide whether to forward the measurements to the remote cloud. Assuming $\lambda_i \in [0, 1]$ is the data splitting ratio, denoting the data The entire processing time for idle and less busy ECN i, without offload to the remote cloud, is the computation delay and is denoted by,

$$t_i^{comp,e} = RS_i \cdot \tau_i,\tag{7}$$

where τ_i is the time spent on one unit of the processing resource in the ECN *i*. For a busy ECN, the whole processing time includes i) the computation delay of itself

$$t_i^{comp,e} = \lambda_i \cdot RS_i \cdot \tau_i, \tag{8}$$

ii) transmission delay from the ECN to the remote cloud

$$t_i^{tran,e} = (1 - \lambda_i) \cdot L_i / R_i + (1 - \lambda_i) \cdot L_i / W_i, \qquad (9)$$

where L_i is the size of the processed data, R_i is the wireless channel capacity assigned to the ECN*i* and W_i is the backhaul communication capacity used by the ECN *i* (R_i^{-1} and W_i^{-1} can be interpreted as the required time for the wireless and backhaul links to transmit one-bit data separately [39], [40]), and iii) computation delay of the remote cloud

$$t_i^{comp,c} = (1 - \lambda_i) \cdot \mathbf{RS}_i \cdot \tau_{c,i}, \tag{10}$$

where $\tau_{c,i}$ is the time spent on one unit of the processing resource of ECNi by the assigned cloud computation resources.

Problem Formulation: Based on the above delay types, the overall processing delay of the *i*-th busy ECN can be derived as,

$$T_i = max \{ t_i^{comp,e}, \ t_i^{tran,e} + t_i^{comp,c} \},$$
(11)

while the processing delay of the ECN without offloading can be expressed as,

$$T_i = t_i^{comp,e},\tag{12}$$

Which can be treated as a special ECN with $t_i^{tran,e} = 0$ and $t_i^{comp,c} = 0$. Assuming that all ECNs have the same priority, the processing delay of the entire system can be expressed as,

$$T_{whole} = \max\{T_i\}.$$
 (13)

It should be noted that the parameters λ_i , R_i , W_i , and $\tau_{c,i}$ need to be carefully selected and adjusted to minimize the overall system processing delay; T_{whole} . Therefore, the optimization problem can be formulated as:

$$\min_{\lambda_i, R_i, W_i, \tau_{c,i}} I_{\text{whole}},$$
s.t. $0 \le \lambda_i \le 1, \sum_{i}^{E} R_i \le R_{all}, \sum_{i}^{E} W_i \le W_{all},$

$$\sum_{i}^{E} \tau_{c,i} \le \tau_{c,all},$$
(14)

where R_{all} and W_{all} are the overall wireless and backhaul communication resource constraints of ECNs, $\tau_{c,all}$ corresponds to the maximum available computing resources provided by the cloud server. The optimization variables include the data splitting ratio λ_i , the communication resource allocation R_i and W_i , as well as the computation resource allocation $\tau_{c,i}$.

Optimal Data Splitting and Resource Allocation: For busy ECNs, the splitting ratio λ_i should be selected to minimize processing delay T_i and the overall processing delay T_{whole} . As mentioned in [39], optimization of splitting ratio λ_i is complicated and its solution depends on the available wireless channel capacity, backhaul link capacity, edge node computation resources, and remote cloud resources. However, to achieve event detection and classification, there are finite choices for λ_i . For example, in the case of a busy ECN where each period is composed of 5 successive time slots (one time slot for each event), the splitting ratio; λ_i can be selected only as 0%, 20%, 40%, 60%, 80% or 100% (note that the processed data for one event should not be split). Furthermore, if only 3 events are detected after SVM prediction (i.e., no event occurs in two of the time slots), the size of the processed data is reduced to 60% of the entire data. Under these conditions, choices for λ_i are 0%, 33.3%, 66.7% and 100%. As discussed in [39], when ECNs have far less edge computation resources than the remote cloud, they should offload more processed data to remote cloud in order to ease data processing load at ECNs. Similarly, when wireless channel and backhaul link capacities are not sufficient, more data should remain at ECNs. Furthermore, as far as the wireless channel and backhaul link capacity assignment are concerned, it is reasonable to assign more capacity to edge cloud nodes with more offload to the remote cloud to increase the values of R_i and W_i . Similarly, in the remote cloud, to reduce the value of $\tau_{c,i}$ more computation resources need to be assigned to the edge cloud node with more offloading. Since, in our approach we use multiple machine learning algorithms, their complexities and sizes of input vectors/matrices are significantly different. Therefore, more computation resources in the remote cloud should be allocated to the ECN using a high complexity algorithm.

After exchanging information with MIBSs and the remote cloud, MABSs will derive: the optimal offload data size of ECNs, optimal wireless channel, backhaul link capacity, and remote computation resources assignment. Specifically, the collaborative cloud and edge computing aim to minimize the processing delay T_i of ECNs, as well as the overall processing delay T_{whole} of the whole system according to the following steps:

- 1: Initial values of R_i , W_i and $\tau_{c,i}$ will be calculated by equally assigning wireless channel and backhaul link capacity and remote cloud computation resources to none-idle ECNs.
- 2: Based on these initial values, each none-idle ECN will derive the splitting ratio λ_i and the size of the offload to remote cloud by minimizing the processing delay T_i .

TABLE 1. Confusion matrix of SVM for events categorization.

Events	Disturbances	Faults		
Disturbances vs Faults	(GT, LS, OS, LT)	(GF, LF, BF, TF)		
Disturbances	346	54		
Faults	51	349		
Overall Accuracy = 86.43%				

- 3: The values of R_i and W_i will be updated by assigning wireless channel and backhaul link capacities proportionally to the size of the offload data. The value of $\tau_{c,i}$ will be updated by assigning remote cloud computation resources proportionally to the offload processing resource, where the complexity and size of the input vector/matrix are taken into consideration.
- 4: Based on the updated values of R_i , W_i and $\tau_{c,i}$, the splitting ratio λ_i and the size of the offload to remote cloud will be updated by re-minimizing the processing delay T_i .
- 5: Steps 3 and 4 will be repeated until achieving stable λ_i , minimum T_i and the T_{whole} .

Note that as there are not many choices for λ_i , it is not difficult to derive the optimal λ_i and minimize the overall processing delay T_{whole} .

IV. SIMULATION

In our simulation, the EMTP tool¹ is used to simulate power grid events and generate measurement data. First, we considered a scenario for a single ECN where 800 power quality events are used, including 100 GTs, 100 LSs, 100 OSs, 100 LTs, 100 GFs, 100 LFs, 100 BFs and 100 TFs. They are first categorized by the SVM algorithm. As shown in Table 1, 346 disturbances are correctly categorized, including 89 GTs, 87 LSs, 84 OSs and 86 LTs. 51 faults are wrongly categorized as disturbances, including 13 GFs, 15 LFs, 12 BFs and 11 TFs. These 397 events are sent to LSTM for processing. Meanwhile, 349 faults are correctly categorized, including 91 GFs, 87 LFs, 86 BFs and 85 TFs. 54 disturbances are wrongly categorized as faults, including 12 GTs, 14 LSs, 15 OSs and 13 LTs. These 403 events are sent to LeNet for identification. As shown in the confusion matrix of Table 2 and 3, LSTM achieves an accuracy of 90.18% for identifying the 397 events, while LeNet obtains an accuracy of 90.07% for identifying the 403 events. The overall accuracy of the proposed event classification method is 90.13%, which is the best performance as shown in the comparison Table 4, where 800 events are processed separately and comparatively using only LSTM, SVM, LeNet, and the proposed event classification method. Tables 5 and 6 display the relative performances of the LSTM, SVM, and LeNet on disturbances and faults, respectively. In Table 7, the proposed event classification method is compared with LSTM, SVM, and LeNet

^{1.} Certain commercial equipment, instruments, or materials are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

TABLE 2. Confusion matrix of LSTM for events categorization.

GT vs LS vs	GT	LS	OS	LT	GF	LF	BF	TF
OS vs LT vs								
GF vs LF vs								
BF vs TF								
GT	87	0	0	1	1	0	0	0
LS	2	80	0	4	0	1	0	0
OS	6	2	74	1	0	0	0	1
LT	4	3	1	76	0	1	1	0
GF	1	0	0	0	11	1	0	0
LF	0	1	0	1	0	12	1	0
BF	0	0	0	1	0	1	9	1
TF	0	0	0	0	1	1	0	9
	Overall Accuracy = 90.18%							

TABLE 3. Confusion matrix of LeNet for events categorization.

GT vs LS vs	GT	LS	OS	LT	GF	LF	BF	TF
OS vs LT vs								
GF vs LF vs								
BF vs TF								
GT	10	0	0	1	1	0	0	0
LS	1	11	0	2	0	0	0	0
OS	2	1	12	0	0	0	0	0
LT	2	0	0	10	0	1	0	0
GF	2	0	0	0	82	4	1	2
LF	0	3	0	1	0	81	3	1
BF	0	0	0	2	0	4	79	3
TF	0	0	0	0	2	4	1	78
Overall Accuracy = 90.07%								

TABLE 4. Comparison results of the proposed method.

Methods	LSTM	SVM	LeNet	Proposed
				Method
Accuracy	84.7%	85.1%	85.4%	90.13%

TABLE 5. Comparison results of the algorithms on 400 disturbances.

Methods	LSTM	SVM	LeNet
Accuracy	91.3%	81.1%	80.7%

TABLE 6. Comparison results of the algorithms on 400 faults.

Methods	LSTM	SVM	LeNet
Accuracy	82.1%	80.5%	89.5%

 TABLE 7.
 Comparative results of the proposed method using the real PMU data (200 oscillations and 200 line faults).

Methods	LSTM	SVM	LeNet	Proposed Method
Accuracy	83.5 %	84.0%	84.75%	89.75%

by using a set of real PMU data, which includes 200 oscillations and 200 transmission line faults. The PMU data of 200 oscillations is collected from the Frequency Monitoring Network System (FNET). FNET is a WAMS operated by the University of Tennessee Knoxville and Oak Ridge National Lab (ORNL). The real dataset of 200 transmission line faults is extracted from a two-year field-recorded PMU data in the U.S. Western Grid interconnection. Table 7 shows that the proposed event classification method achieves a similar performance on real PMU data as that on EMTP data.

In Figs. 4, 5 and 6, we compare performances of cloud and edge detections, where the scenario of a single edge cloud



FIGURE 4. The packet loss performance comparison of cloud detection and edge detection.



FIGURE 5. The average latency performance comparison of cloud detection and edge detection.



FIGURE 6. The traffic comparison of cloud detection and edge detection.

node and a remote cloud server is considered. In our simulations, 50 PMUs are generating synchrophasors real-time data with sampling rates ranging from 10 to 60 samples per second. The generated data will be first processed locally and categorized into four different types of events: namely local disturbances, global disturbances, local faults, and global faults. The local disturbances and faults will be processed locally using edge detection, where local disturbances are sent to the LSTM algorithm for identification, while the faults are processed by using the LeNet algorithm. The related data will be stored and processed locally. On the other



FIGURE 7. The performance of the proposed edge cloud sharing scheme.



FIGURE 8. The average convergence rate of the proposed edge cloud sharing scheme.

hand, the global disturbances and faults will be upload to the remote cloud for classification. In our simulations, the percentages of global events range from 10%, 20% and 50%. Figs. 4 and 5 demonstrate the significant advantage of edge detection over cloud detection in terms of packet loss and average latency, which verifies the efficiency of the proposed edge detection.

Fig. 6 evaluates the amount of traffic produced on the network. As can be observed the traffic produced by cloud detection is far more than by edge detection. Though the traffic of edge detection will increase with the increasing percentage of global events, it can still achieve a far better performance than cloud detection.

In Figs. 7 and 8, we evaluate the performance of the proposed collaborate cloud and edge computing in a three-edge-cloud-node scenario, where each edge cloud node is handling 50 PMUs. The wireless channel bandwidth between MIBSs and the MABS is set to 20 MHz, the backhaul link capacity is 100Mbps, edge cloud computation capacity is 1×10^{10} CPU cycle/s, and remote cloud computation capacity is 2×10^{11} CPU cycle/s. Within each time-period, multiple events, simulated using the EMTP, will be randomly distributed in the system. Each period can be split to

TABLE 8.	Comparison results	of the algorithms on	the same amount of data.
----------	--------------------	----------------------	--------------------------

Machine	LeNet	AlexNet	LSTM	GRU
learning				
Total	60K	60M	8626	6551
Parameters				
Input	32×32×	227×227×	60	60
Vector/Matri	1	3		
x size				
Processing	34s	156s	73s	57s
Time (s)				
Accuracy	89.5%	93.6% for	91.3% for	90.7% for
	for	faults	disturbance	disturbance
	faults		s	s

5 successive time slots where each is allocated to one possible event. Each event consists of 15K bits measurement data. In this case, only local events are considered. The local disturbance and local faults are equally likely to occur. At the beginning of a time period, each ECN uses SVM to categorize the power events. The average processing time of SVM is around 2.3s. LSTM is used to process local disturbances and LeNet or AlexNet is considered to process local faults. If the power signal is categorized as normal no action will be taken. The processing resource used by every edge cloud will be then derived and its value will be exchanged with the MABS. The collaborate cloud and edge computing scheme aim to balance the load and to minimize the whole processing delay T_{whole} of the entire system. As can be seen from Fig. 7, the collaborate cloud and edge computing scheme can significantly reduce the processing delay T_{whole} of the system, especially for the scenarios with more events. Fig. 8 depicts the average convergence rate of the proposed edge cloud sharing scheme in a scenario where the average events number in each time period is 9. As mentioned earlier, there are not many choices for λ_i and the average convergence rate of the proposed scheme is around 10 iterative steps.

In Table 8, the processing time and complexity of the employed LeNet, AlexNet, LSTM, and GRU are investigated and compared. Events with same amount of data (15K measurement data) are used for all algorithms. Specifically, LeNet or AlexNet is used for processing faults, while LSTM or GRU is used for processing disturbances. Compared with LeNet, the AlexNet achieves a better accuracy at the expense of higher complexity. On the other hand, the GRU reduces the gating signals to two compared to LSTM. However, it has to increase the hidden state of size n to achieve a similar performance to LSTM, leading to a similar complexity for a similar accuracy performance.

V. CONCLUSION

In this article, we present a power grid event classification method where different AI-assisted algorithms are used to process different events that can effectively improve system performance. For a large-scale power grid system with multiple edge cloud nodes, a collaborative cloud and edge computing scheme is introduced to balance the load and reduce the overall processing delay, T_{whole} , of the whole system. The complexity and processing time of some widely used machine learning algorithms, such as LSTM, GRU, LeNet, and AlexNet, are investigated and compared. Using different scenarios our investigations show that the proposed event classification method, together with the collaborative cloud and edge computing scheme, can improve accuracy and effectively share the balance.

REFERENCES

- D. Zhang, X. Han, and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE J. Power Energy Syst.*, vol. 4, no. 3, pp. 362–370, Sep. 2018.
- [2] P. E. A. Cardoso, "Deep learning applied to PMU data in power systems," Ph.D. dissertation, Faculdade De Engenharia Da, Universidade Do Porto, Porto, Portugal, 2017.
- [3] W. Hu, L. Zheng, Y. Min, Y. Dong, R. Yu, and L. Wang, "Research on power system transient stability assessment based on deep learning of big data technique," *Power Syst. Technol.*, vol. 41, no. 10, pp. 3140–3146, Oct. 2017.
- [4] W. Liu, D. X. Zhang, X. Y. Wang, J. Hou, and L. Liu, "A decision making strategy for generating unit tripping under emergency circumstances based on deep reinforcement learning," *Proc. CSEE*, vol. 38, no. 1, pp. 109–119, Jan. 2018.
- [5] "Convolutional neural networks (LeNet) tutorial." [Online]. Available: http://yann.lecun.com/exdb/lenet/
- [6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [7] Z. Zhao, G. Xu, Y. Qi, N. Liu, and T. Zhang, "Multi-patch deep features for power line insulator status classification from aerial images," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 3187–3194.
- [8] H.-Z. Wang, G.-Q. Li, G.-B. Wang, J.-C. Peng, H. Jiang, and Y.-T. Liu, "Deep learning based ensemble approach for probabilistic wind power forecasting," *Appl. Energy*, vol. 188, pp. 56–70, Feb. 2017.
- [9] Y. Zhu, C. Liu, and K. Sun, "Image embedding of PMU data for deep learning towards transient disturbance classification," in *Proc. IEEE Int. Conf. Energy Internet (ICEI)*, 2018, pp. 169–174.
- [10] M. K. Alqudah, M. Pavlovski, T. Dokic, M. Kezunovic, Y. Hu, and Z. Obradovic, "Fault detection utilizing convolution neural network on timeseries synchrophasor data from phasor measurement units," *IEEE Trans. Power Syst.*, vol. 37, no. 5, pp. 3434–3442, Sep. 2022, doi: 10.1109/TPWRS.2021.3135336.
- [11] Y. Tao, J. Zheng, T. Wang, and Y. Hu, "A state and fault prediction method based on RBF neural networks," in *Proc. IEEE Workshop Adv. Robot. Soc. Impacts*, Shanghai, China, Jul. 2016, pp. 221–225.
- [12] E. Rakhshani, I. Sariri, and K. Rouzbehi, "Application of data mining on fault detection and prediction in boiler of power plant using artificial neural network," in *Proc. Int. Conf. Power Eng. Energy Elect. Drives*, Lisbon, Portugal, Mar. 2009, pp. 473–478.
- [13] M. Mahdi and V. M. I. Genc, "Artificial neural network based algorithm for early prediction of transient stability using wide area measurements," in *Proc. 5th Int. Istanbul Smart Grid Cities Congr. Fair (ICSG)*, 2017, pp. 17–21.
- [14] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, Jul. 2017.
- [15] H. Zuo, H. Fan, E. Blasch, and H. Ling, "Combining convolutional and recurrent neural networks for human skin detection," *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 289–293, Mar. 2017.
- [16] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using RNN pre-trained by recurrent temporal restricted Boltzmann machines," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 23, no. 3, pp. 580–587, Mar. 2015.
- [17] S. Leglaive, R. Hennequin, and R. Badeau, "Singing voice detection with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, South Brisbane, QLD, Australia, Apr. 2015, pp. 121–125.
- [18] V.-T. Tran, K.-H. Nguyen, and D.-H. Bui, "A vietnamese language model based on recurrent neural network," in *Proc. 8th Int. Conf. Knowl. Syst. Eng.*, Hanoi, Vietnam, Oct. 2016, pp. 274–278.

- [19] W.-W. Liu, W.-Q. Zhang, Y. Shi, A. Ji, J. Xu, and J. Liu, "Improved phonotactic language recognition based on RNN feature reconstruction," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Florence, Italy, May 2014, pp. 5322–5326.
- [20] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection," *IEEE Trans. Instrum. Meas.*, vol. 57, pp. 981–988, 2008.
- [21] C. Xu, G. Wang, X. Liu, D. Guo, and T.-Y. Liu, "Health status assessment and failure prediction for hard drives with recurrent neural networks," *IEEE Trans. Comput.*, vol. 65, no. 11, pp. 3502–3508, Nov. 2016.
- [22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] T. de Bruin, K. Verbert, and R. Babuška, "Railway track circuit fault diagnosis using recurrent neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 523–533, Mar. 2017.
- [24] M. Yuan, Y. Wu, and L. Lin, "Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network," in *Proc. IEEE Int. Conf. Aircraft Utility Syst.*, Beijing, China, Oct. 2016, pp. 135–140.
- [25] S. Zhang, Y. Wang, M. Liu, and Z. Bao, "Data-based line trip fault prediction in power systems using LSTM networks and SVM," *IEEE Access*, vol. 6, pp. 7675–7686, 2018.
- [26] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, arXiv:1409.1259.
- [27] V. Vapnik and L. Bottou, "Local algorithms for pattern recognition and dependencies estimation," *Neural Comput.*, vol. 5, no. 6, pp. 893–909, Nov. 1993.
- [28] N. G. Baltas, P. Mazidi, J. Ma, F. de Asis Fernandez, and P. A. Rodriguez, "A comparative analysis of decision trees, support vector machines and artificial neural networks for on-line transient stability assessment," in *Proc. Int. Conf. Smart Energy Syst. Technol.* (SEST), Seville, Spain, Sep. 2018, pp. 1–6.
- [29] W. Hu et al., "Real-time transient stability assessment in power system based on improved SVM," *J. Mod. Power Syst. Clean Energy*, vol. 7, no. 1, pp. 26–37, Jan. 2019.
- [30] H. Mohammadi, G. Khademić, M. Dehghani, and D. Simon, "Voltage stability assessment using multi-objective biogeography-based subset selection," *Int. J. Elect. Power Energy Syst.*, vol. 103, pp. 525–536, Dec. 2018.
- [31] H. R. Baghaee, D. Mlakić, S. Nikolovski, and T. Dragicević, "Support vector machine-based islanding and grid fault detection in active distribution networks," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 8, no. 3, pp. 2385–2403, Sep. 2020.
- [32] M. Chen, Y. Miao, H. Gharavi, L. Hu, and I. Humar, "Intelligent traffic adaptive resource allocation for edge computing-based 5G networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 499–508, Jun. 2020.
- [33] H. Gharavi and B. Hu, "Scalable synchrophasors communication network design and implementation for real-time distributed generation grid," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2539–2550, Sep. 2015.
- [34] IEEE Standard for Synchrophasor Data Transfer for Power Systems, IEEE Standard C37.118.2011-2, Dec. 28, 2011.
- [35] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.
- [36] S. Chen et al., "Physical-layer channel authentication for 5G via machine learning algorithm," Wireless Commun. Mobile Comput., vol. 19, Oct. 2018, Art. no. 6039878.
- [37] B. Hu and H. Gharavi, "A hybrid wired/wireless deterministic network for smart grid," *IEEE Wireless Commun.*, vol. 28, no. 3, pp. 138–143, Jun. 2021.
- [38] A. Abdiansah and R. Wardoyo, "Time complexity analysis of support vector machines (SVM) in LibSVM," *Int. J. Comput. Appl.*, vol. 128, no. 3, pp. 28–34, Oct. 2015.
- [39] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5031–5044, May 2019.
- [40] G. Zhang, T. Q. S. Quek, M. Kountouris, A. Huang, and H. Shan, "Fundamentals of heterogeneous backhaul design—Analysis and optimization," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 876–889, Feb. 2016.