Detecting Cyber Attacks in Smart Grids with Massive Unlabeled Sensing Data

Hanyu Zeng*‖, Zhen Wei Ng*, Pengfei Zhou*¶, Xin Lou*§, David K.Y. Yau*†, Marianne Winslett*‡
*Advanced Digital Sciences Center, Illinois at Singapore †Singapore University of Technology and Design
‡University of Illinois at Urbana-Champaign, USA §Singapore Institute of Technology
¶University of Pittsburgh, USA ‖National University of Singapore

***Abstract*—Modern power grids are undergoing significant changes driven by information and communication technologies (ICTs), and evolving into smart grids with higher efficiency and lower operation cost. Using ICTs, however, comes with an inevitable side effect that makes the power system more vulnerable to cyber attacks. In this paper, we propose a self-supervised learning-based framework to detect and identify various types of cyber attacks. Different from existing approaches, the proposed framework does not rely on large amounts of well-curated labeled data but makes use of the massive unlabeled data in the wild which are easily accessible. Specifically, the proposed framework adopts the BERT model from the natural language processing domain and learns generalizable and effective representations from the unlabeled sensing data, which capture the distinctive patterns of different attacks. Using the learned representations, together with a very small amount of labeled data, we can train a task-specific classifier to detect various types of cyber attacks. Experiment results in a 3-area power grid system with 37 buses demonstrate the superior performance of our framework over existing approaches, especially when a very limited amount of labeled data are available. We believe such a framework can be easily adopted to detect a variety of cyber attacks in other power grid scenarios.**

I. Introduction

Smart grid has been equipped with networks of sensors and generators allowing two-way communication within the system with information and communication technologies (ICTs), which can help the operators manage a larger scale area of power distribution [1]. This feature, however, also makes the power system more vulnerable to cyber attacks, e.g., false data injection (FDI) attack [2] and time delay (TD) attack [3]. The purpose of cyber attacks is mainly to cause drastic frequency passivity and finally crash the whole system [4].

Within the power grids, automatic generation control (AGC) is one of the most important systems but also vulnerable to cyber attacks [4, 5]. AGC adjusts the generators' output to make the frequency of the system within a safe range. Breach of this safe range due to frequency excursion caused by cyber attacks can cause damage in the system [2, 4]. Therefore, in this work, we consider a practical scenario where AGC is distributed in large scale with networked sensors collecting sensing data, e.g., the system frequency and power export. In a practical running system, we have no knowledge of when and where the attacks would happen and the massive sensing data collected in the wild are left unlabeled. As manually collecting and labeling sensing data for different cyber attacks is expensive and time consuming, it is challenging to leverage very limited number of labeled data to develop effective models to detect different cyber attacks in real time.

Researchers have proposed several mechanisms [6, 7, 8, 9, 10, 11] to detect and identify cyber attacks in power systems. Some previous works [6, 7, 8, 9] make use of supervised learning models and require a large number of labeled data to achieve accurate attack detection, which is not scalable for real systems. Unsupervised learning based methods, including the traditional machine learning based [1, 10, 12, 13] and deep learning based [14, 15], have been proposed to make anomaly detection in power grids using the unlabeled data. In practice, different cyber attacks have completely different means to disrupt the power system, which indicates different countermeasures need to be taken against them. As a result, it is not sufficient to do anomaly detection only. Recent works [14, 15] leverage unsupervised learning for the detection of FDI and TD attack, respectively. These methods, however, only target a single specific type of attack.

We aim to take a step further towards making multi-type cyber attack detection and classification with the knowledge learned from massive unlabeled sensing data and propose PowerBERT, a BERT-like [16] self-supervised learning model to deal with the sensing data in smart grids for cyber attack detection. The proposed PowerBERT learns effective and generalizable representations from massive unlabeled sensing data collected in the wild. Once the representations have been learned, together with a small amount of labeled data for the targeted types of attacks, we can easily train a task-specific classifier to detect various types of attacks.

The original BERT was designed for natural language processing (NLP) and lacks the methodology to deal with sensing data in power grids where the data distributions are different and require in-depth investigation. Inspired by the observation that cyber attacks usually cause both temporal and spatial signal variation across the power girds, in this paper, we propose to learn effective representations with the sensing data collected from neighboring areas in a region. We segment the time-series sensing data into data partitions and each partition corresponds to an *event*. A series of events in each detection window are fed into PowerBERT to learn effective representations that can capture the spatial-temporal

978-1-6654-3254-2/22/$31.00 ©2022 IEEE

signal fluctuation patterns caused by cyber attacks, and the patterns caused by different attacks are distinctive.

We show the effectiveness of the learned representations for detecting the FDI and TD attacks with a random forest classifier. By leveraging a very small amount of labeled data (0.05%), We can achieve 98.0% and 78.8% detection accuracy for FDI and TD attack, respectively. Using 0.05%∼1% labeling rate, PowerBERT-based method outperforms existing models at least by 6.1% to 3.1% in terms of F1-score.

The main contributions of this paper are as follows:

- We propose PowerBERT, a BERT-like auto-encoder to learn the generalizable and effective representations with massive unlabeled sensing data from neighboring areas in smart grids. We propose to segment the time-series sensing data with different window sizes and learn the best configuration for detecting multiple types of attacks in the AGC control in power grids.
- We train a random forest classifier based on the learned representations with a small amount of labels for FDI and TD attack. The classifier can be easily adopted to detect other types of attacks with corresponding labeled data.
- We implement the proposed framework using reshape layers. We show the selection of hyper parameters, e.g., event window size, for the model and compare the performance of the proposed framework with existing approaches. The results demonstrate the effectiveness of PowerBERT in learning effective representations for identifying FDI and TD attacks. The code is available[1].

The rest of this paper is organized as follows. Section II introduces the system model and attack models. Section III presents the methodology and design of the framework. Section IV reports the experiment settings, ablation study results and comparative performance of PowerBERT-based method and state-of-the-art methods. Section V discusses the related work, and Section VI concludes this paper.

II. System model and attack models

In this paper, we consider the cyber attacks in automatic generation control (AGC) in power grids as our case study for the proposed detection framework. In the following, we first introduce the AGC model and then describe attack models for FDI attack and TD attack, respectively.

A. AGC Model

In a power system, AGC regulates the grid's frequency within a safe range by dynamically adjusting the system conditions in real-time [17]. A power grid can be divided into several separate areas, and the AGC can also control the power interchange rate among different control areas. In this paper, we discuss the discrete-time AGC system, where the time is divided into slots. We illustrate a three-area power grid with 37 buses in Figure 1(a) [5]. This system involves three control areas and the dotted lines between two control areas are called tie-lines. In this paper, we use this 37-bus system as our case study to explore the cyber attacks in AGC control, which is a representative power grid model denoting a small to middle-scale real-world grid.

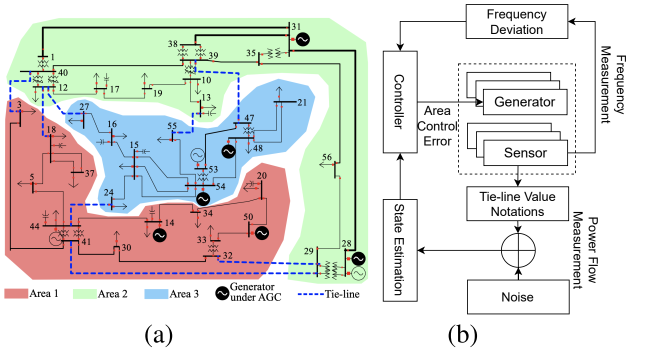

Figure 1. The system model. (a) Three-area power grid with 37 buses. (b) Overview of AGC.

In the AGC system, the area control error (ACE) is a control command used to regulate the generator output in the feedback control. For an area i in the grid shown in Figure 1(b), the command ACE$_i$ is a weighted sum of two signals inside the power grid i.e., the frequency deviation ($\Delta\omega_i$) and power export deviation (ΔP_{Ei}). Thus, it can be expressed as: $ACE_i = a_i \Delta P_{Ei} + b_i \Delta\omega_i$, where the a_i and b_i are two constant weights. The control center sends the ACEs command to adjust the generator output via the communication network in different control area i. This control process is called the AGC cycle which is usually about 2 to 4 seconds [17].

The power flow measurement from the power system is usually faulty and noisy, so the state estimation (SE) is designed to recover the information from noisy signal. The measurement vector y can be expressed as: $y = \mathbf{M}x + \mathbf{n}$, where $\mathbf{M}$ represents the measurement matrix, vector x denotes all the states in the grid, and the $\mathbf{n}$ denotes the noise. The target of SE is to estimate the state vector x by $\hat{x} = (\mathbf{M}^\top \mathbf{W}\mathbf{M})^{-1}\mathbf{W}y$, where $\mathbf{W}$ is a weighted matrix. Then the estimated power flow measurement is $\hat{y} = \mathbf{M}\hat{x}$. In Bad Data Detection (BDD) [18], the alarm will be triggered if the difference between y and $\hat{y}$, i.e., $||y - \hat{y}||$, is bigger than a defined threshold.

B. Attack Models

In this paper, we use the representations learned using PowerBERT to detect two typical cyber attacks against AGC control in the power grid [4, 6], the latest FDI attack [2] and TD attack [3]. The learned representations can also be used to detect other types of cyber attacks in smart grid as long as they cause signal fluctuation in the system.

For traditional FDI attacks, after the adversaries know the power flow matrix $\mathbf{M}$, they can add attack vector $a = \mathbf{M}c$ into the power flow sensor measurement, where c is an arbitrary vector, and the measurement becomes $\hat{y} = \mathbf{M}(x + c) + \mathbf{n}$, so BDD is bypassed because the noise does not change. The targeted FDI attack [2] in our work not only lends matrix M to bypass BDD but also limits the magnitude of the false data added by the FDI attack in a reasonable range so that the attack minimizes disruptions when it initially enters the system and

[1] https://github.com/fridge23/PowerBERT

keeps the frequency excursion long enough to ensure system damage. Compared to traditional FDI attacks, the attack we exploit is stealthier and more destructive [2].

In the time delay (TD) attack, the adversary aims to delay the control command from the controller. Let $y(t)$ denote the control command generated and transmitted by the control center in the tth time slot. The adversary maliciously delays these packets by τ time slots. Thus, in the $(t+\tau)$th time slot, the command $y(t)$ arrives at the actuator. Since we consider the discrete-time AGC control system in this paper, the delay length τ is an integer. Moreover, different from FDI attacks, the adversary does not modify any content of the transmitted packet. The TD attack can be launched by compromising the data communication channels (e.g., compromised routers) between the controller and the actuator to delay the transmission of control commands [4]. Note that delayed signals may exist in the system even without the cyber attacks due to the natural communication latency. In the AGC, the attacker delays the control command in one of the areas i, i.e., $\text{ACE}_i(t)$, by τ slots, to create the system frequency excursion.

Overall, by either compromising the sensor readings (i.e., FDI attack) or delaying the control commands (i.e., TD attack), the purpose of the adversary is to make the system's frequency exceed the safety threshold and then force the disconnection between the generator and load or damage equipment. Same as the existing work [2, 6], we consider the safety range of the frequency deviation as [-0.5, 0.5] Hz, and the deviations out of this range are regarded as unsafe.

III. Methodology

In this section, we introduce the details of the proposed cyber attacks detection and classification model. The overview of our framework is illustrated in Figure 2, which consists of 3 phases, i.e., data preprocessing, PowerBERT, and downstream classifier training. The sensing data collected from neighboring control areas in AGC are firstly normalized and extracted with a specific data structure. All the extracted sets of data are then fed into the PowerBERT self-supervised learning model to learn representations, which are used to train the downstream task-specific classifier with supervised learning.

A. Data Preprocessing

Data normalization: We use the min-max scalar to normalize the collected ACE data. The normalized data sample can be express as:$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}$,where the $x_i' \in (x_1', x_2'..., x_n')$ is the scaled result, $x_i \in (x_1, x_2..., x_n)$ is the original value and n is related to the amount of data we have, x_{min} is the smallest value and the x_{max} is the biggest. By using scalar, all the data are in the range of [0, 1].

Data extraction: We use a sliding window to extract data clips from the normalized dataset. As illustrated in Figure 3, the sliding window with width w_1 is used to extract a set of data segments $B_i \in (B_1, B_2, ..., B_m)$ from a data trace collected in the 3 neighboring areas, and w_1 is also the window size of attack detection. For each B_i $(i = 1, 2, ..., m)$, we divide the data segment with a second window w_2 into a set of data partitions $E_i = E_{i1}, E_{i2}...E_{ik}$, where each partition E_{ij} $(j = 1, 2, ..., k)$ captures the sensing signal distribution at a short time interval, and we regard such a partition as an event. All the events in E_i capture the signal fluctuation in each detection window. As illustrated in Figure 3, before E_i is fed into PowerBERT, we use a reshape layer to reshape the data from $w_1 \times 3_{dim}$ to $\frac{w_1}{w_2} \times (3w_2)_{dim}$.

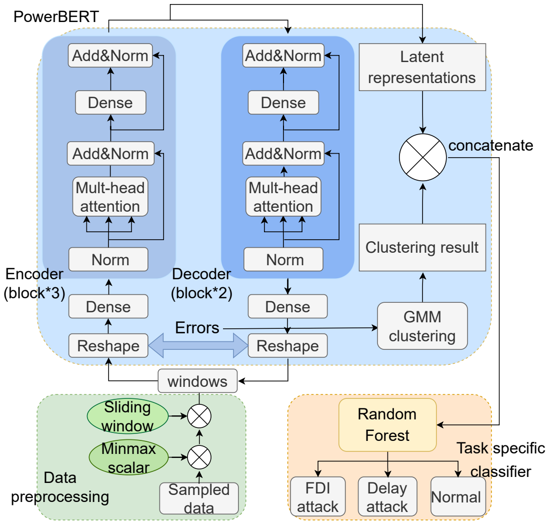

Figure 2. The proposed framework is comprised of data preprocessing, the PowerBERT model and a task-specific classifier.

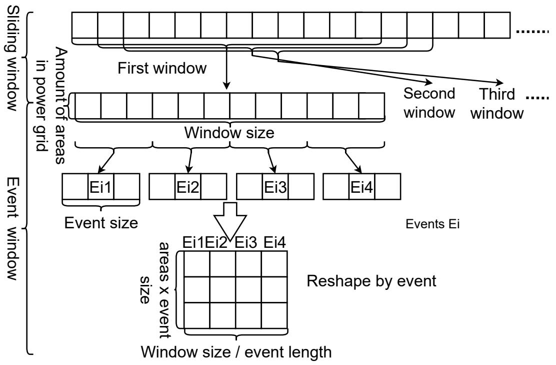

Figure 3. The process of data extraction.

B. PowerBERT

PowerBERT is adopted from BERT model [16, 19, 20] to extract the high-dimensional representations from the massive unlabeled data. In NLP domain, BERT is a bidirectional model used to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. In this paper, we do not make use of the span mask algorithm to train PowerBERT. It is because the inputs in original BERT are word tokens while the input data in PowerBERT are continuous data samples. The whole model is trained by back propagation to reduce the reconstruction error.

Before the processed data are fed into the encoder, a dense layer is used to embed data into high-dimensional tensor. For instance, the ws points' data is embedded from $ws \times 3_{dim}$ to $ws \times D_{dim}(D > 3)$.

Encoder: The encoder involves 3 transformer blocks. Each block includes a layer Normalization layer, a multi-headed attention layer, an adding layer that works as a residual connection, and finally a fully connected layer. The process of block i can be expressed as: $B^i = MultiAttn(LayerNorm(A^i_{in})), A^i_{out} = Dense(LayerNorm(B^i + A^i_{in})) + B^i + A^i_{in}$, where A^i_{in} denotes the data fed into block i, B^i denotes the data output by multi-headed attention layer, A^i_{out} denotes the output data of block i.

Decoder: The outputs of the encoder go to the decoder, where the extracted high-dimensional representations are reconstructed. The decoder has 2 blocks inside. The encoder has more blocks than the decoder, for the reason that we need a more complex encoder to extract better features for the downstream tasks. The formulas of the blocks m can be expressed as: $D^m = MultiAttn(LayerNorm(C^m_{in})), C^m_{out} = Dense(LayerNorm(D^m + C^m_{in})) + D^m + C^m_{in}$, where C^m_{in} is the inputs of block m, and C^m_{out} is the outputs of block m, and D^m denotes the outputs of multi-headed attention layer. After the decoder, a fully connected layer is designed to reshape the data back to the original structure.

Train: The loss of the autoencoder is computed based on the difference between the original data and the reconstructed data. By using back propagation, the weights in the model are updated, where Adam optimizer [21] is used for updating weights. Mean absolute error (MAE) function, i.e., $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$, is utilized to calculate the loss. Compared with the mean square error, MAE is more sensitive with errors less than 1. For the learning rate, we use the learning rate warm-up to speed up the training process.

Feature extraction: After training, we extract the appropriate features from PowerBERT for the training of the downstream classifier. In addition to the latent representations, we also make use of the reconstruction error distributions to better train the classifier. The representations are the outputs of the encoder of PowerBERT, and the error distributions, as illustrated in Figure 2, are extracted from the reconstruction errors of PowerBERT, i.e., $X_{in} - X_{out}$, where X_{in} is the input of PowerBERT, and X_{out} is the reconstructed results. We utilize a Gaussian Mixture Model (GMM) to analyze the distributions of the errors. Some unlabeled data are utilized to train a GMM which clusters the error distributions into k types. For the reason that one type of attack may cause several types of error distributions, k should be much bigger than the cyberattack types $l(k >> l)$. Lastly, we concatenate the representations and the error distributions as the features used for classifier training.

C. Downstream Classifier Training

Once the features have been well learned from PowerBERT, we use a small amount of labeled data to train the classifier to do the classification of FDI and TD attack. We deploy a random forest model with 1000 estimators as the classifier. Although being a small amount, the labeled data includes all the targeted types of cyber attacks. In our case study, the classifier is trained to identify the data without an attack (i.e., normal), FDI attack and TD attack.

IV. Evaluation

We now evaluate the performance of the proposed framework for detecting the FDI attack (FDIA) and TD attack (TDA) against AGC in the power grid. We first describe the dataset and evaluation metrics, and then briefly introduce other state-of-the-art attack detection models. After that, we show our model performance and the comparison with other models.

A. Methodology

Dataset: We use industry-strength power system simulator PowerWorld [22] to simulate cyber attacks against AGC in the three-area 37-bus model as shown in Figure 1(a). We add randomly generated load deviations to simulate real-world dynamics. The ACE data samples are collected every 4 seconds. We collect data from the 3 control areas shown in Figure 1(a) when the power system is under FDI attack [2], TD attack [3], and without attack, respectively, and all the attacks are launched in area 3 at random time. If the extracted data segment (as introduced in Section III-A) contains any data samples that are collected when the system is under attack, the segment is labeled as the corresponding type of attack. In total, we collect around 17,000 traces, where there are 6,944 traces without any attack, 4,990 traces involving TD attack and 5,000 traces involving FDI attack.

We randomly divide the data traces into training (59.5%), validation (10.5%), and testing (30%) set. The entire training set is used for self-supervised representation learning without using the labels. We randomly select $a\%$ $(a = 0.05, 0.1, 1)$ of the training set as the labeled set for supervised classifier training to mimic the practical scenario where only limited labels are available. The labeled set consists of 50% of normal data (no attack), 25% of FDI attack data and 25% of TD attack data.

Metrics: We use the precision, recall and F1-score to evaluate the model performance. Specifically, $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, $f_1 - score = \frac{Precision+Recall}{2}$, where TP denotes true positive, meaning the data segment is classified as the correct class; TN denotes true negative, meaning the data segment of other classes is not classified into the class; FP denotes false positive, meaning the data segment of other classes is classified as the class, and FN denotes false negative, meaning the data segment is classified as other classes.

B. Different learning models

In the evaluation, we compare the proposed model with other alternative models, which are based on state-of-the-art machine learning models in the literature.

PowerBERT+RF model: We use the PowerBERT to extract representations, a 27-component GMM to extract the reconstruction error distributions, and a 1000 estimators' random forest classifier to identify the types of attacks.

Table I
EXPERIMENT RESULT COMPARISON FOR EVENT WINDOW SIZE SELECTION.

Event size(s)	Normal f1	TDA f1	FDIA f1
4	95.8%	75%	98.3%
8	94.9%	69.3%	97.3%
12	94.6%	66.8%	97.1%
20	**96.1%**	**78.5%**	**98.6%**
24	96%	78%	98.5%
40	95.6%	74.4%	97.6%

DNN model [8]: It is a MLP model, which involves 3 hidden layers. Because it was only designed for FDI attack detection, so we change the last layer of the model from 2 units to 3 units and train it to do classification task.

RNN model [9]: RNN model is very sensitive with the temporal information. We used an RNN model with 3 LSTM layers which have 64 units and a 33-unit fully connection layer. For the output layer, we set 3 units to classify the data into different categories.

DB-RF [23]: A variant of random forest, which involves two random forest levels, and the first level performs anomaly detection, and the second level identifies the type of attacks. Two levels work with different kinds of features. We set the model with 330 estimators and train it to do triple classification.

RF: A random forest model that is trained with the raw data instead of the learned representations. The model has 1000 estimators and identify data into 3 categories.

PowerBERT and other models are implemented with python, scikit-learn and tensorflow [24, 25]. They are trained in a server equipped with a PC has 3.7GHZ 6-core processor. The learning rate and batch size in both self-supervised and supervised training phases are 1024.

C. Evaluation Results

1) Event window size selection.: As introduced in Section III, in each detection window, we divide the data samples into data partitions with window size w_2, and regard the partitions as *events*. We compare the performance of the models with different sizes $(w_2 = 4, 8, 12, 20, 24, 40s)$ for data partitioning. In this experiment, the detection window size w_1 is set to $120s$, which corresponds to 30 data samples collected from each grid area, and also involves $(30, 15, 10, 6, 5, 3)$ *events* according to different *event* window sizes. As presented in Table I, the model with *event* window size of $20s$, which corresponds to 5 samples from each area, outperforms models with other sizes for all the 3 classes. Thus we use $w_2 = 20s$ in our final model design.

2) Sliding window size selection: We test the performance of our model with different sizes for the sliding window w_1. Figure 4 plots the F1-score of the models with w_1 size of 20, 40, 60 and $120s$. We see that as the window size increases, the model performs best in TD attack identification when the window size is $40s$, and performs best in FDI attack at window size $120s$. In order to reduce the computation overhead and ensure prompt detection, we use $w_1 = 40s$ in our model.

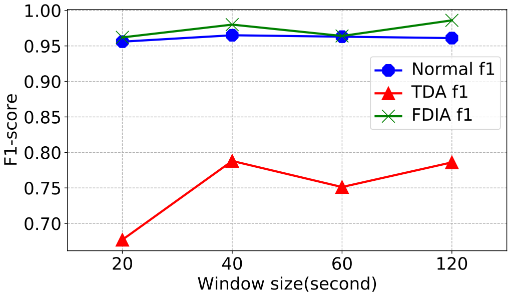

Figure 4. Experiment results in PowerBERT+RF for sliding window size selection.

Table II
PERFORMANCE COMPARISON OF POWERBERT+RF USING THE DATA FROM INDIVIDUAL CONTROL AREA AND ALL 3 CONTROL AREAS.

	Normal f1	TDA f1	FDIA f1
Single-area PowerBERT(average)	88.5%	53.1%	72.8%
Single-area PowerBERT(best)	94.4%	61.7%	95.5%
PowerBERT	**96.5%**	**78.8%**	**98.0%**

3) Effectiveness of spatial redundancy: We compare the performance of two different versions of our model trained using the ACE data from individual control areas and all 3 control areas, respectively. The results are reported in Table II, where we provide the average and best performance (for the model works on the area where attacks launched) of single-area PowerBERTs among the 3 control areas and PowerBERT. We see that the performance of PowerBERT is significantly better than single-area PowerBERT, which demonstrates the effectiveness of using spatial redundancy in smart grids.

4) Effectiveness of reconstruction error distributions: We compare the performance of three classifiers trained with different feature settings using the labeling rate of 0.05%. The classifiers include **representation+RF**: the classifier using the representations as features, **representation+mean+RF**: the classifier using the representations and the mean of reconstruction errors as features, and **PowerBERT+RF**: the classifier using the representations and reconstruction error distributions as features. As shown in Table III, adding the mean value of reconstruction errors does not result in any performance improvement. By combining the error distributions and the representations, we can increase the detection F1-score by 1% for the TD attack. Since the TD attack is one of the most difficult types of attacks to detect in practice, we employ the combination of representations and reconstruction error distributions as the features in our final framework design.

Table III
PERFORMANCE COMPARISON FOR CLASSIFIERS WITH DIFFERENT FEATURES.

Metrics		Representation +RF	Representation +mean+RF	PowerBERT +RF
F1-score	Normal	96.4%	96.4%	**96.5%**
	TDA	77.8%	77.8%	**78.8%**
	FDIA	97.9%	97.9%	**98.0%**

Table IV
MODEL COMPARISON USING DIFFERENT AMOUNT OF LABELED DATA FOR TRAINING.

labeling size	F1-score(%)	DNN	RNN	DB-RF	RF	PowerBERT +RF
0.05%	Normal	93.7	94.7	96.0	96.0	**96.5**
	TDA	58.7	57.7	70.3	72.7	**78.8**
	FDIA	85.7	94.0	96.0	96.3	**98.0**
0.1%	Normal	95.3	96.0	96.3	96.3	**96.7**
	TDA	67.7	72.3	74.0	75.0	**80.6**
	FDIA	91.7	96.7	97.0	98.0	**98.2**
1%	Normal	96.0	96.5	97.0	97.0	**97.2**
	TDA	77.0	80.5	81.0	81.7	**84.8**
	FDIA	97.7	98.5	98.3	**99.0**	98.6

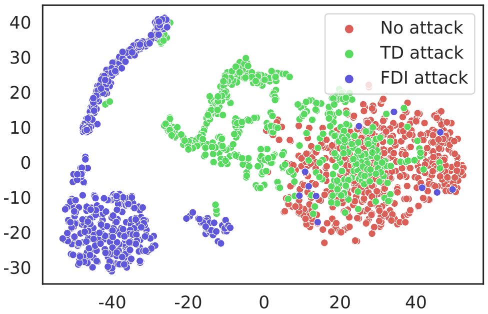

Figure 5. Representation visualization with t-SNE.

5) Model comparison: In this subsection, we compare the detection performance of our model with other state-of-the-art models as introduced in Section IV-B. We show the performance of all the models with three different settings, where the amount of labeled dataset for model training is different. We use the labeling rate of 0.05%, 0.1%, and 1% to train all the models respectively and compare their performance.

Table IV summarizes the F1-score of attack detection using different models. For all of the models, the F1-score increases as the labeling rate increases. PowerBERT+RF achieves the best performance for most of the time, especially when the labeling rate is low. When the labeling rate increases to 1%, all of the models can achieve good performance. In practice, however, 1% labeling rate is usually too expensive to get given the large amounts of sensing data collected in the wild.

6) Representation visualization: To gain a more intuitive understanding of the effectiveness of the representation learned by our model in the classification task, we visualize the learned high-dimensional representations of data in 2D space by t-distributed stochastic neighbor embedding (t-SNE) [26]. We randomly select a total of 1500 equal amounts of no attack data, TD attack data and FDI attack data. Then they are feature extracted by PowerBERT, reduced to two-dimensional data with t-SNE and drawn on a scatter plot. The result is shown in Figure 5. It is obvious that samples belonging to the same types of attacks exhibit high clustering effect. The representations of TD attack and no attack are close to each other, which explains the relatively low classification F1-socre for TD attack as reported in the experiment results.

7) Computation overhead: We show the inference speed of our model and other models in this part. For each model, we let it performs attack detections and calculate the average time needed for one detection. We use a PC has 3.7GHZ 6-core processor to do the experiments. The results are reported in Table V, and we see that all models can make real-time detection and the overhead is affordable for workstations.

Table V
THE INFERENCE TIME FOR EACH SAMPLE IN DIFFERENT MODELS (S).

DNN	RNN	DB-RF	RF	PowerBERT+RF
2.17E-7	2.50E-7	4.40E-6	8.10E-5	5.46E-4

V. RELATED WORK

Researchers have proposed signal processing based and machine learning based approaches to detect cyber attacks in smart grids.

Signal processing based. Some works [27, 28] proposed to detect cyber attacks using the classic signal processing models, where the approaches such as Kalman filter and wavelet singular entropy is used for detecting the existence of cyber attacks. [27] presents a two-stage kalman filter to detect cyber attacks and estimate the bias of the attacks. [28] uses the wavelet singular entropy in FDI attack detection. These methods only perform anomaly detection without the ability to identify different types of attacks.

Machine learning based. Compared with the signal processing based approaches, machine learning based approaches are more robust to the changes and noises in the environment. They include supervised learning approaches and unsupervised learning ones. Supervised learning based models [6, 7, 8, 9] have been proposed to detect various types of cyber attacks in smart grids. Lou et al. [7] exploit BiLSTM based model for the detection of TD attack. Mohammad Ashrafuzzaman et al. propose DNN based model [8] for FDI attack detection. Qingyu Deng et al. use LSTM based model [9] to detect FDI attack in a power grid. These approaches require a large amount of labeled data to train the model for accurate detection. To reduce the reliance on labeled data, unsupervised learning approaches [10, 11, 12, 13, 14, 15] are studied. [11] compares the performance of the combination of machine learning models and statistical feature extraction methods. [10] detects anomaly by leveraging KNN model. One-class SVM also be used in anomaly detection [12], as well as isolation forest [13]. Yang at el. propose WPD-ResNet model to do transfer learning and detect anomalies in power station communication [15]. And stacked denoising autoencoder is used to detect and classify several types of FDI attack [14]. These methods either do anomaly detection [10, 12, 13] instead of attack classification or only target a specific type of attacks [14]. In this paper, we propose a BERT-like model to learn the generalizable and effective representations that capture distinctive patterns of different attacks and, as a result, can be used to identify different types of attacks.

VI. CONCLUSION

In this paper, we proposed PowerBERT, a self-supervised learning model to learn the generalizable features from massive unlabeled sensing data for cyber attack detection in smart grids. We demonstrated the effectiveness of the PowerBERT-based framework in detecting and identifying two common types cyber attacks in AGC in power grids, and it has a better performance in the downstream cyber attacks classification than other signal processing based and machine learning based models. We believe the proposed framework can be easily adopted to be implemented in other scenarios since it only requires easily accessible unlabeled data and a very small amount of labeled data to achieve superior performance.

Acknowledgement

This research is supported in part by the National Research Foundation, Singapore, and the Energy Market Authority, under its Energy Programme (EP Award No. NRF2018-NCR003-0013), and in part by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

[1] Jacob Sakhnini, Hadis Karimipour, and Ali Dehghantanha. "Smart grid cyber attacks detection using supervised learning and heuristic feature selection". In: *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*. IEEE. 2019, pp. 108–112.

[2] Weili Yan et al. "A Stealthier False Data Injection Attack against the Power Grid". In: *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE. 2021, pp. 108–114.

[3] Arman Sargolzaei, Kang K Yen, and MN Abdelghani. "Time-delay switch attack on load frequency control in smart grid". In: *Advances in Communication Technology* 5 (2013), pp. 55–64.

[4] Xin Lou et al. "Assessing and mitigating impact of time delay attack: Case studies for power grid controls". In: *IEEE Journal on Selected Areas in Communications* 38.1 (2019), pp. 141–155.

[5] Rui Tan et al. "Modeling and mitigating impact of false data injection attacks on automatic generation control". In: *IEEE Transactions on Information Forensics and Security* 12.7 (2017), pp. 1609–1624.

[6] Prakhar Ganesh et al. "Learning-based simultaneous detection and characterization of time delay attack in cyber-physical systems". In: *IEEE Transactions on Smart Grid* 12.4 (2021), pp. 3581–3593.

[7] Xin Lou et al. "Learning-based time delay attack characterization for cyber-physical systems". In: *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE. 2019, pp. 1–6.

[8] Mohammad Ashrafuzzaman et al. "Detecting stealthy false data injection attacks in power grids using deep learning". In: *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE. 2018, pp. 219–225.

[9] Qingyu Deng and Jian Sun. "False data injection attack detection in a power grid using RNN". In: *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2018, pp. 5983–5988.

[10] Yu An and Dong Liu. "Multivariate Gaussian-based false data detection against cyber-attacks". In: *IEEE Access* 7 (2019), pp. 119804–119812.

[11] Jacob Sakhnini, Hadis Karimipour, and Ali Dehghantanha. "Smart grid cyber attacks detection using supervised learning and heuristic feature selection". In: *2019 IEEE 7th international conference on smart energy grid engineering (SEGE)*. IEEE. 2019, pp. 108–112.

[12] Muhammad Sharif Uddin and Anthony Kuh. "Online least-squares one-class support vector machine for outlier detection in power grid data". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 2628–2632.

[13] Saeed Ahmed et al. "Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest". In: *IEEE Transactions on Information Forensics and Security* 14.10 (2019), pp. 2765–2777.

[14] Chengming Hu, Jun Yan, and Chun Wang. "Robust Feature Extraction and Ensemble Classification Against Cyber-Physical Attacks in the Smart Grid". In: (2019), pp. 1–6. DOI: 10.1109/EPEC47565.2019.9074827.

[15] Ting Yang et al. "WPD-ResNeSt: Substation station level network anomaly traffic detection based on deep transfer learning". In: *CSEE Journal of Power and Energy Systems* (2021).

[16] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[17] Prabha Kundur. "Power system stability". In: *Power system stability and control* 10 (2007).

[18] Yao Liu, Peng Ning, and Michael K Reiter. "False data injection attacks against state estimation in electric power grids". In: *ACM Transactions on Information and System Security (TISSEC)* 14.1 (2011), pp. 1–33.

[19] Kaiming He et al. "Masked autoencoders are scalable vision learners". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16000–16009.

[20] Huatao Xu et al. "LIMU-BERT: Unleashing the Potential of Unlabeled Data for IMU Sensing Applications". In: *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 2021, pp. 220–233.

[21] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[22] *PowerWorld*. http://www.powerworld.com/ Accessed April 4, 2010. 2021.

[23] Yasir Ali Farrukh et al. "A sequential supervised machine learning approach for cyber attack detection in a smart grid system". In: *2021 North American Power Symposium (NAPS)*. IEEE. 2021, pp. 1–6.

[24] Martín Abadi et al. "TensorFlow: a system for Large-Scale machine learning". In: *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 2016, pp. 265–283.

[25] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[26] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[27] Ayyarao SLV Tummala and Ravi Kiran Inapakurthi. "A two-stage Kalman filter for cyber-attack detection in automatic generation control system". In: *Journal of Modern Power Systems and Clean Energy* 10.1 (2021), pp. 50–59.

[28] Moslem Dehghani et al. "Cyber attack detection based on wavelet singular entropy in AC smart islands: False data injection attack". In: *IEEE Access* 9 (2021), pp. 16488–16507.