2024 11th International Conference on Power and Energy Systems Engineering (CPESE) | 979-8-3503-7743-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/CPESE62584.2024.10841160

Joint Convolutional Neural Network and Bidirectional Gated Recurrent Unit Model for False Data Injection Attacks Detection in Smart Grid

1st Mengna Sun
School of Electronic and Electrical Engineering
Ningxia University
Yinchuan, China
18306763792@163.com

2nd Huan Pan*
School of Electronic and Electrical Engineering
Ningxia University
Yinchuan, China
pan198303@gmail.com
*Corresponding author

3rd Chunning Na
School of Electronic and Electrical Engineering
Ningxia University
Yinchuan, China
nana508@163.com

4th Hu Li
School of Electronic and Electrical Engineering
Ningxia University
Yinchuan, China
18254811767@163.com

***Abstract*—False data injection attack (FDIA), as a kind of covert cyber attack, can be secretly tampered with measurement data to steal power or destabilize the operation of the smart grid. With the upgrading of monitoring equipment, the amount of data in the smart grid is getting bigger, and the feature vectors are getting more and more complex. To further improve the detection performance of FDIA, this paper combines convolutional neural network (CNN) with bidirectional gated recurrent unit (Bi-GRU) to construct a new hybrid detection model. The model integrates the features of CNN and Bi-GRU can better extract data features, i.e., CNN adopts a layer-by-layer structure to extract data features, and Bi-GRU acts as a bi-directional network to capture both past and future information, and ultimately outputs the hidden state at each time step. We take a small smart grid as an example to simulate and verify the FDIA detection effect of different models, where the dataset comes from the phasor measurement units. The results show that for varying levels of attack intensity, CNN-Bi-GRU model consistently maintains high detection accuracy.**

***Index Terms*—False data injection attack (FDIA), convolutional neural network (CNN), bidirectional gated recurrent unit (Bi-GRU), smart grid**

I. Introduction

With the rapid development of digital communication technology, the power grid and information systems have been highly coupled, and power systems are becoming smarter and smarter. The emergence of smart grids has brought new advantages to the development of power systems. However, the convergence of communication networks, the Internet of Things, and power grids has also brought network threats [1], [2]. Various network attacks can cause potential harm to the security, reliability, and economy of the grid and can even lead to outage incidents and cascading failures. How to effectively defend, detect, and mitigate cyber-attacks has become a pressing problem in today's engineering and academic fields.

There are various types of cyber-attacks and false data injection attacks (FDIA), as stealthy attacks, can disrupt the regular operation of the power system by modifying the power grid measurement data and even cause inevitable power equipment failures [3]. The most crucial feature of FDIA is its covert nature; even if the attacker knows part of the information on the grid, he or she can construct false attack data and even cause the same destructive effect as a physical attack. Therefore, establishing an effective FDIA detection model is the first barrier to protecting the smart grid. The current methods for detecting and classifying false data on the grid include traditional techniques and machine learning (ML). Traditional approaches rely heavily on statistical and optimization algorithms that require extensive mathematical calculations and rely on experienced operators to determine the type or location of an intrusion based on the calculations [4]. The ML-based detection methods can directly obtain the decision result through offline training and online detection, which have the advantages of faster detection, more vital generalization ability, and less data requirement. [5], [6].

As the functionality of the grid's intelligent monitoring equipment continues to improve, the amount of measurement data obtained increases with each passing day. The large amount of data increases the computational complexity of

This work is partially supported by National Natural Science Foundation of China under Grant 61763040, the Natural Science Foundation of Ningxia Province under Grant 2022AAC05001, and the Western Light Talent Training Program of Chinese Academy of Sciences.

the ML-based detection model, causing a decrease in FDIA detection accuracy. Deep learning (DL) is the latest in ML algorithms, which consists of layers of artificial neurons and has been applied to image recognition, visual art processing, natural language processing, and other fields. Trained on large amounts of data, it can process new data and ingest and analyze data from multiple sources in real-time without human intervention. For the advantages based on DL algorithms, numerous scholars have utilized them to detect measurement data from PMUs to automatically learn the features of state measurement data from each node in the grid to discover or classify abnormal state sequences [8]–[20]. Wang et al. designed a two-stage learner based on the Kalman filter (KF) and recurrent neural network (RNN) to perform FDIA detection [8]. Lei et al. presented a prediction residual classification to detect FDIA, where the nonlinear prediction was based on a gated RNN [9]. Yang et al. formulated a long and short-term memory (LSTM)-Autoencoder approach for online FDIA detection in smart grids [10]. Mukherjee presents a novel real-time FDIA identification scheme using nonlinear LSTM [11]. James et al. and Xu et al. adopted deep neural networks (DNN) in FDIA detection research [12], [13], and they considered AC and DC state estimation, respectively. Deep self-encoders were combined with randomized trees, generative adversarial networks (GAN), and deep representation learning in [14]–[16], respectively, to detect FDIAs. Yang et al. proposed a hybrid FDIA detection model consisting of the principal component analysis (PCA) and convolutional neural network (CNN) [17]. He et al. proposed a single hidden layer neural network algorithm to detect stealthy FDIAs [18]. Boyaci et al. proposed a graph neural network (GNN) based detector for FDIAs, which effectively combines model-driven and data-driven to model the graph topology and spatially relevant measurement data automatically [19]. Li et al. built a Transformer-based FDIA detection model for local training of each node [20].

Despite the various FDIA detection models proposed so far, most contain two steps or stages [8]–[17], the first for data processing and the second for classification. For example, RNNs are used to extract nonlinear features [8], [9], LSTMs are trained to learn the temporal correlation of multi-dimensional data [10], [11], autoencoders, and PCA are adopted to reduce the dimensionality of measurement datasets [14]–[17]; some typical classifications include Logistic Regression (LR) [10], error covariance matrix [11], DNN [12], [13], randomized trees [14], GAN [15], and CNN [17]. However, the above models are relatively complex in structure and need better real-time detection performance. The neural network developed in [18] is simple but focuses on a specific FDIA. Although GNN and Transformer are the latest DL algorithms, GNN is less adaptive and loses some features when feature extraction is performed online [19]. As a global network, Transformer tends to ignore the local characteristics of the input [20]. Due to some shortcomings of the existing available results, to improve the detection accuracy of FDIA, this paper will design a new FDIA detection model that can realize high-precision detection while meeting real-time requirements.

RNNs are often used to analyze temporal correlation in datasets [8]–[11]. However, RNNs have a short-term memory problem, which would result in the loss of some critical information and affect FDIA's detection performance. To overcome this problem, we adopt bidirectional gated recurrent unit (Bi-GRU) to store the data information in both directions [21]. CNNs have excellent properties such as local connectivity, weight sharing, pooling operation, and multilayer structure, enabling FDIA detection directly [22]. Nevertheless, CNN inevitably needs to catch up on some data features as a locally connected network. As a result, we will add Bi-GRU to the CNN structure to provide a training stage with moderate accuracy and no overfitting phenomenon for feature extraction of high-dimensional data. The main contribution of this paper is summarized as follows.

- Bi-GRU is combined with CNN to form a hybrid model, CNN-Bi-GRU, for FDIA detection.
- CNN-Bi-GRU can model the forward and backward of the measurement sequence after CNN processing, output the forward and backward hidden states of each time step, make full use of the adequate information of the time series data, and improve the final extraction effect of features.
- We have assessed the effectiveness of CNN-Bi-GRU on a standard micro power system. The simulation results have validated that the proposed scheme has higher detection performance than some representative FDIA detection models and fulfills the real-time requirements.

The rest of this paper is organized as below. Section II describes FDIA's basic attack principle. Section III designs the CNN-Bi-GRU hybrid model and introduces each part. The simulation analyses are done in Section IV. Section V draws the conclusion of the results.

II. FDIA Principle

A. Power System State Estimation

Conventional and synchronous phase measurements are the two main measurement methods for state estimation. Conventional measurements are collected through a supervisory control and data acquisition (SCADA) system or remote terminal units (RTU), which usually include bus-injected power P_i, Q_i and branch power flow P_{ij}, Q_{ij}. Synchronous phase measurements are taken from the PMUs through the wide-area measurement system (WAMS), including the phase angle and amplitude of the three-phase voltage and current, the system frequency, and its rate of change. Since the PMU can directly measure the bus voltage and branch current, these values follow Ohm's law on the complex domain and have linear relationships. Therefore, we use the DC state estimation directly to simplify the calculation. The specific expressions are as follows.

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e}, \tag{1}$$

where $\mathbf{z} = [z_1, z_2, \cdots, z_m]^T$, $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$, and $\mathbf{e} = [e_1, e_2, \cdots, e_m]^T$ denote the m-dimensional measure-

ment, the n-dimensional state, and the random measurement error vectors, respectively; $\mathbf{H}$ represent the grid topological Jacobi matrix.

The weighted least squares (WLS) method is used to minimize the error between the measurement and the estimated state vectors to determine the existence of undesirable data in the measured values, and the objective function is described by

$$\min \mathcal{J} = (\mathbf{z} - \mathbf{H}\mathbf{x})^T \mathbf{R}^{-1} (\mathbf{z} - \mathbf{H}\mathbf{x}), \tag{2}$$

where $\mathbf{R}$ indicates the measurement covariance matrix.

The estimated value of the state vector is denoted as

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{R} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R} \mathbf{z}. \tag{3}$$

B. Bad Data Detection

In the actual operation of the power system, the bad data detection (BDD) module serves to check for the present of false data. The specific process of BDD is to substitute the optimal solution solved from state estimation into the measurement model to obtain the difference between the measured and estimated values. This difference is referred to as the state estimation residual $\mathbf{r}$, denoted as

$$\mathbf{r} = \mathbf{z} - \mathbf{H}\hat{\mathbf{x}}, \tag{4}$$

where $\mathbf{r}$ is the bias vector with $E(\mathbf{r}) = \mathbf{0}$.

When the measurement error $\mathbf{e}$ follows a normal distribution, the residual $\mathbf{r}$ satisfies the 3σ law, i.e., with 99.7% probability, $\mid r_j \mid \leq 3\sigma_j$, where σ_j is the standard deviation of the measurement device. If $\mid r_j \mid > 3\sigma_j$, it is reasonable to suspect that the measurement error does not obey a normal distribution and undesirable data may be injected, i.e., the system injects an attack vector $\mathbf{a}$, at which point the false measurement value $\mathbf{z_a}$ is

$$\mathbf{z_a} = \mathbf{z} + \mathbf{a}. \tag{5}$$

Denote $\mathbf{c} = [c_1, c_2, ..., c_n]^T$ as the error vector caused by the FDIA, and the state vector under the attack is expressed as

$$\mathbf{x_a} = \hat{\mathbf{x}} + \mathbf{c}. \tag{6}$$

Then, the residual expression is

$$\begin{aligned} \| \mathbf{r} \|_2 &= \| \mathbf{z_a} - \mathbf{H}\mathbf{x_a} \|_2 \\ &= \| \mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c}) \|_2 \\ &= \| \mathbf{z} - \mathbf{H}\hat{\mathbf{x}} + \mathbf{a} - \mathbf{H}\mathbf{c} \|_2 \end{aligned} \tag{7}$$

where $\| \cdot \|_2$ denotes L_2-norm of a vector.

When the attack vector satisfies $\mathbf{a} = \mathbf{H}\mathbf{c}$, the residual vector is

$$\| \mathbf{r} \|_2 = \| \mathbf{z_a} - \mathbf{H}\mathbf{x_a} \|_2 = \| \mathbf{z} - \mathbf{H}\hat{\mathbf{x}} \|_2 . \tag{8}$$

Eq. (8) illustrates that when the attack vector $\mathbf{a}$ satisfies the relation $\mathbf{a} = \mathbf{H}\mathbf{c}$, the FDIA initiated by an attacker can successfully pass through BDD thus posing a threat to the secure operation of the power grid [3].

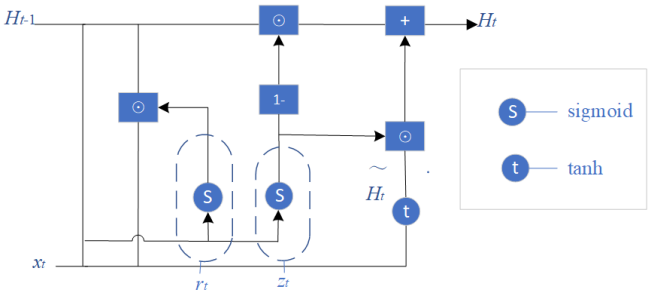

Fig. 1: GRU internal structure.

III. CNN-Bi-GRU Model

A. CNN Structure

As a DNN with a convolutional structure, CNN can reduce the amount of memory occupied by the deep network, effectively reducing the number of parameters of the network and alleviating the overfitting problem of the model [23]. At the same time, it helps the subsequent feature classification when performing feature extraction.

The CNN consists of an input layer, multiple convolutional layers, pooling layers, and an output layer. When the time series z enters the input layer, the convolution kernel in the convolution layer convolves the local information to extract features from the input data. The convolution operation in each layer is performed by the ReLU activation function, as shown in the following equation.

$$c_{1,j} = ReLU(z * h_{1,j} + b_{1,j}) \tag{9}$$

where c_{1j} represents the feature mapping generated from the first convolutional layer operation on the input data z; $h_{1,j}$ is the j-th convolution kernel, which acts as a filter, $*$ denotes the convolution operation and $b_{1,j}$ is the corresponding scalar deviation.

Pooling layers can effectively reduce the size of the parameter matrix, thus reducing the number of parameters in the final fully connected layer. The common pooling operations are average pooling and maximum pooling, and the maximum pooling method is chosen in this paper. The fully connected layer receives the features extracted through the convolution and pooling layers as input. Then, the weights and deviations of the features are calculated in the fully connected layer. In the last stage, the Softmax or Sigmoid function is applied to classify the extracted features. We chose the Sigmoid function here to perform binary classification for input data to obtain the final output.

B. GRU System Structure

The internal structure of GRU is similar to that of LSTM, which merges the unit states and hidden states by integrating the forgetting and input gates into a single update gate [24], as shown in Fig. 1.

Fig. 1 indicates that the GRU consists of an update gate and a reset gate instead of the three gates (input, output, and

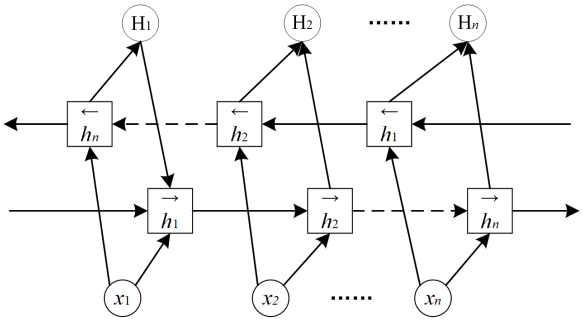

Fig. 2: The structure schematic of Bi-GRU.

forget gate) in the LSTM. The update gate z_t of the GRU is a combination of the input gate and the forgetting gate of the LSTM, i.e., a convex combination of the elements of the previous hidden state H_{t-1} and the candidate hidden state $\tilde{H}_t$; the reset gate r_t is used to determine how much of the past information should be forgotten. The specific expressions for z_t, r_t, $\tilde{H}_t$ and H_t are

$$z_t = \sigma(w_z x_t + \mu_z H_{t-1}), \tag{10}$$

$$r_t = \sigma(w_r x_t + \mu_r H_{t-1}), \tag{11}$$

$$\tilde{H}_t = \tanh(w_h x_t + \mu_h (r_t \odot H_{t-1})), \tag{12}$$

$$H_t = (1 - z_t) \odot H_{t-1} + z_t \odot \tilde{H}_t. \tag{13}$$

where x_t is the input layer vector; μ is the weight vector connecting the previously hidden layer and the current hidden layer; w is the input weight vector connecting the current hidden layer; H_t and H_{t-1} are the current and previous hidden layer outputs, respectively, $\tilde{H}_t$ represents the candidate activation vector.

C. CNN-Bi-GRU Model Framework

GRU has a more straightforward structure and lowers computational complexity. It has one less gate than LSTM and reduces matrix multiplication. The GRU is chosen to build a DL architecture, which can further reduce the model overfitting effect and improve the algorithm detection robustness while increasing the computing speed [25]. The structure of Bi-GRU is shown in Fig. 2, to extract depth features from the input sequence [26]. Further, a hybrid CNN-Bi-GRU model is developed to improve the detection performance of FDIA in a smart grid where the structure is given in Fig. 3, which includes four modules: input layer, hidden layer, output layer, and training module.

The pseudocode of the CNN-Bi-GRU model is given by Algorithm 1, where the model parameters and evaluation metrics will be given in the next section.

IV. Simulation Analysis and Comparison

The FDIA detection simulation is implemented in the Jupyter platform in Anaconda. The parameters of the simulation computer are as follows: two Intel Xeon Scalable Gold 6226R CPUs, a six-channel DDR4 memory controller, 128GB

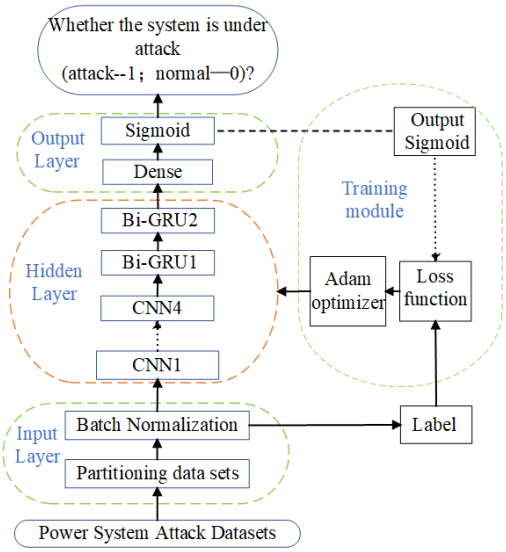

Fig. 3: The training framework of CNN-Bi-GRU.

Algorithm1: CNN-Bi-GRU

Input: Original data $D = \{x^{(n)}, y^{(n)}\}_{n=1,\ldots,N}$, learning rate, epoch, batch.

Split the dataset into training (80%) and test (20%) sets;
Training phase:
for number of epoch **do**
  for number of batch **do**
    $X_1 \leftarrow Conv1(X_{train})$
    $X_2 \leftarrow Conv2(X_1)$
    $X_3 \leftarrow Conv3(X_2)$
    $X_4 \leftarrow Conv4(X_3)$
    $X_5 \leftarrow Bi-GRU1(X_4)$
    $X_6 \leftarrow Bi-GRU2(X_5)$
    $\hat{y}_{train} \leftarrow FCL(X_6)$
    $\min\{binary\_crossentropy(y_{train}, \hat{y}_{train})\}$
    Update network weights
  end
end
Testing phase:
$\hat{y}_{test} = Network.predict(X_{test})$
Calculate the model evaluation indicators according to $(\hat{y}_{test}, y_{test})$

Output: Values of evaluation indicators

(8*16GB) of DDR4 RECC shared memory, and an AMD Radeon Pro WX 3200 graphics card.

The proposed CNN-Bi-GRU hybrid model is built and trained using Tensorflow, and the classification experiments are trained offline. The simulation process of FDIA detection mainly consists of preparing the dataset, giving performance evaluation metrics, using different models for FDIA detection, and comparing the performance of various detection methods. Each part is described in detail next.

A. Dataset

The dataset used in this section is the same as in [16], [27], [28], which is collected by Oak Ridge National Laboratory. The example power system topology is shown in Fig. 4, which includes several intelligent electronic devices, a monitoring

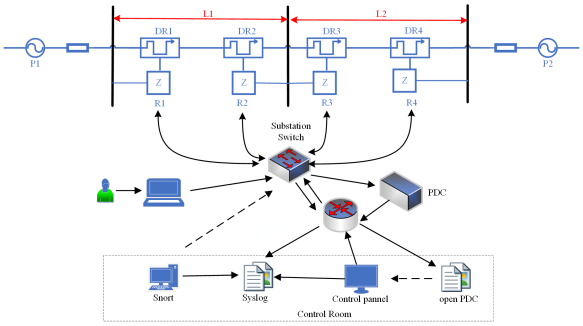

Fig. 4: Micro power system topology diagram.

system, and network monitoring equipment [28]. The specific implications of all notations in Fig. 4 are presented in Table I.

TABLE I: Description of notation in Figure 4.

Notation	Description
P1, P2	three-phase generators
R1, R2, R3, R4	intelligent electronic devices
DR1, DR2, DR3, DR4	transmission lines
L1, L2	Phase A-Phase C Current Magnitude

The micro power system described in Fig. 4 can operate under various scenarios to generate three data types: no event, natural event and attack event. FDIA detection aims to distinguish attack events from other types of events. Hence, we only need the dichotomous dataset and consider no and natural events normal.

The binary dataset consists of 15 sub-datasets, within each of which attack events are randomly injected, with 128 features per sub-dataset, including data collected by 4 PMUs, snort alerts, and system logs. Each PMU measures 29 feature variables, and the remaining 12 features are log information, feature name, and control panel description. The detailed information is shown in [16], [27], [28].

B. Detection Performance Evaluation Metrics

The four metrics, accuracy, precision, recall, and F1-score, are used to evaluate the FDIA detection effectiveness of the designed hybrid models, where the following equations give their expression.

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}\%, \tag{14}$$

$$Pre = \frac{TP}{TP + FP}\%, \tag{15}$$

$$Rec = \frac{TP}{TP + FN}\%, \tag{16}$$

$$F1 = 2 \times \frac{Pre \times Rec}{Pre + Rec}\% = \frac{2TP}{2TP + FP + FN}\%, \tag{17}$$

where TP represents the number of abnormal samples detected as abnormal samples, TN denotes the number of normal samples detected as normal samples, FP is the number of normal samples detected as abnormal samples, and FN is the number of abnormal samples detected as normal samples.

C. Analysis of Simulation Results

In the simulation process, 128 features collected by 4 PMUs are input as measurements, and 15 data subsets are sequentially trained, with the attack samples of each data subset consisting of random sampling. Each data subset is divided into training dataset and test dataset according to the ratio of 8:2. Then, the training dataset is divided into the training dataset and cross-validation dataset according to the proportion of 8:2.

One input layer, four CNN layers, two Bi-GRU layers, three fully connected layers, and one Sigmoid layer are used to form the CNN-Bi-GRU model, as shown in Table II.

TABLE II: CNN-Bi-GRU model parameters.

Types of parameters	CNN-Bi-GRU
Input feature dimension	(128, 128, 1)
Con. layer	4/Relu
Con. kernel size	5*5 / 3*3 / 3*3 / 3*3
Con. kernel quantity	128 / 128 / 256 / 256
Layer type	Bi-GRU layer
Layer number	2
Kernel quantity	128 / 256
Dense layer	3
Units on the first layer	128/Relu
Units on the second layer	128/Relu
Units on the third layer	1/sigmoid
Loss function	Binary crossentropy
Optimizer	Adam

The pooling and dropout layers in traditional convolutional networks help reduce model overfitting because the pooling layer can effectively downsampling high-dimensional features. However, due to the one-dimensional convolution operation in this paper, the introduction of pooling and dropout layers has little effect on reducing overfitting. It even minimizes detection accuracy to a certain extent. Therefore, the pooling and dropout layers are not incorporated into the proposed model structure to achieve the minimum computational burden without losing model performance. Instead, a fully connected layer between the CNN and Bi-GRU and a fully connected layer between the Bi-GRU and Sigmoid layers are applied to incorporate L2 regularizers. The function is gradually approximated to the desired optimal solution by adding penalty parameters to the model weights. The penalty parameter can be changed in different situations to make the model parameters more sparse and reasonable to optimize the model detection performance.

To demonstrate the superior performance of the designed hybrid model for FDIA detection, it is compared with CNN, Bi-LSTM, CNN-LSTM, and CNN-Bi-LSTM detection methods. We use the same labeled dataset to train and compare the performance of the above detection models, and the comparative results are displayed in Figures 5-8.

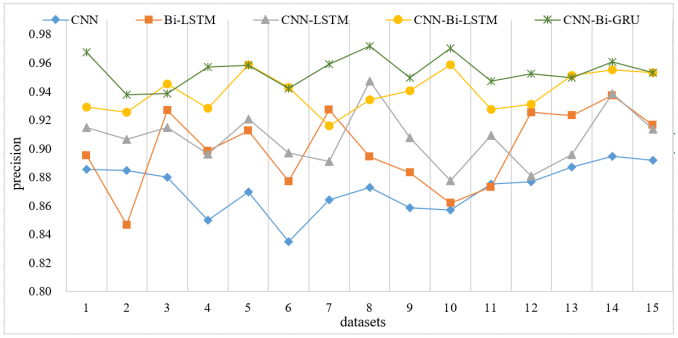

Fig. 5: Comparison chart of precision results.

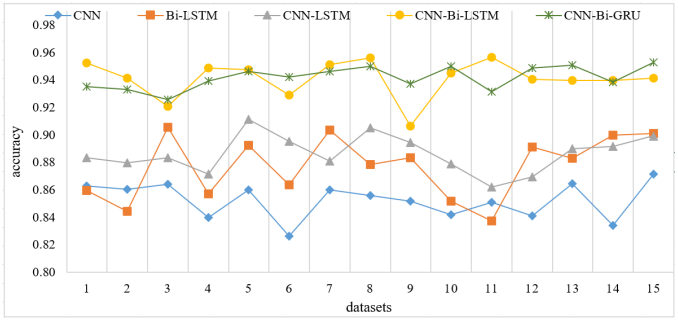

Fig. 6: Comparison chart of accuracy results.

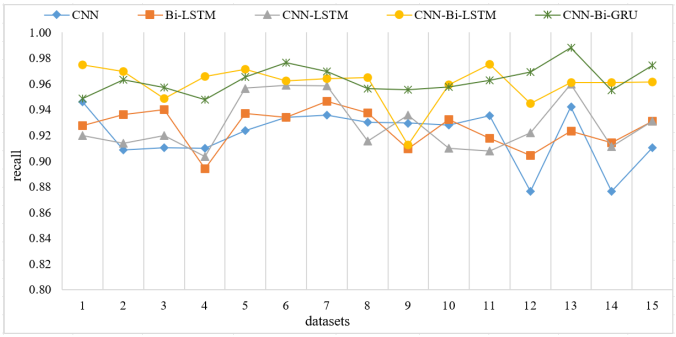

Fig. 7: Comparison chart of recall rate results.

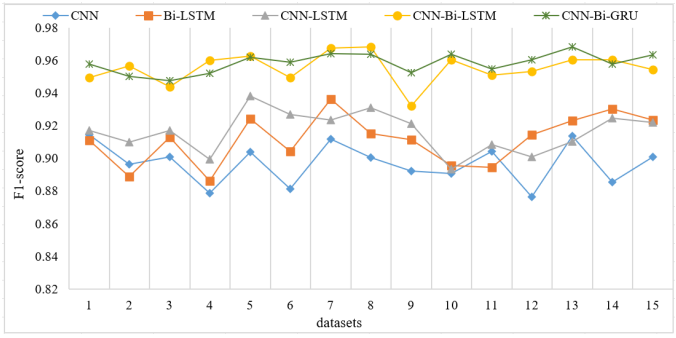

Fig. 8: Comparison chart of F1-score results.

Figs. 5-8 show the evaluation metric curves of false data detection of the five models in 15 sub-datasets, respectively. The comparison demonstrates that the designed CNN-Bi-GRU models is significantly better than the other algorithms in terms of accuracy, F1-score, precision, and recall. In 15 datasets with different random attack strengths, the CNN-Bi-GRU detection accuracy fluctuates less and shows the best robustness.

The mean values of the detection performance evaluation metrics of the above five models under 15 datasets are listed in Table III. It can be found that the difference between Bi-LSTM and CNN-LSTM detection accuracy is not much, but the CNN-Bi-LSTM detection accuracy is higher than the CNN-LSTM model. It is because the Bi-LSTM can combine the hidden states of the input time series forward at the moment $t-1$ and backward at the moment $t+1$ when computing the hidden states at the moment t, while the one-way LSTM can only utilize the hidden states at the moment $t-1$. Compared with LSTM, Bi-LSTM can benefit from the information of both front and back directions of the time series data and make full use of the data features collected by PMUs. The comparison results of the detection evaluation of the five models reveal that the CNN-Bi-GRU model has the best detection performance, with an average improvement of 1.55% in accuracy, 1.52% in recall, and about 1.11% in F1-score compared with the suboptimal model CNN-Bi-LSTM.

TABLE III: Mean values of detection performance indexes of five models.

Model	Pre	Acc	Rec	F1-score
CNN-Bi-LSTM	0.9397	0.9410	0.9598	0.9554
CNN-Bi-GRU	**0.9543**	**0.9417**	**0.9631**	**0.9586**
CNN-LSTM	0.9073	0.8863	0.9282	0.9164
Bi-LSTM	0.8998	0.8767	0.9256	0.9116
CNN	0.8721	0.8522	0.9197	0.8969

D. Comparative Analysis

This section compares CNN-Bi-GRU model with three typical ML algorithms (LR, SVM, K-nearest neighbors (KNN)) and two new DL algorithms (graph convolutional network (GCN), Transformer) for FDIA detection on 15 sub-datasets. The comparison of detection performance and runtime is given in Table IV.

TABLE IV: Detection performance indexes of five models.

Model	Pre	Acc	Rec	F1-score	Time/s
CNN-Bi-GRU	**0.9543**	**0.9417**	0.9631	**0.9586**	0.1820
KNN	0.8812	0.9039	0.9296	0.9165	0.0886
LR	0.7150	0.7321	0.9405	0.8225	**0.0819**
SVM	0.7080	0.7080	**1.0000**	0.8347	0.6470
GCN	0.7103	0.5224	0.7104	0.5912	2.2401
Transformer	0.7019	0.4947	0.7019	0.5798	4.1716

The CNN-Bi-GRU has the highest average precision, average accuracy, and average F1-Score among the five detection

models in Table IV. The average detection time of the CNN-Bi-GRU model for the 15 data subsets is 0.1820s, which meets the online detection demand (usually less than 5 minutes [29]).

The results of GCN and Transformer were surprising and did not achieve the expected detection performance. The main reasons are that the power grid shown in Fig. 4 is a small system; GCN focuses on extracting the structural features of the grid, ignoring certain data features, such as not considering time-series features; the Transformer is a globally connected network, and the long-range attention mechanism can easily overlook the local characteristics of the input. In summary, compared with other ML and DL algorithms, the CNN-Bi-GRU model has higher detection accuracy for FDIA and is more suitable for detecting FDIA in a real-time detection smart grid.

V. Conclusion and Discussion

As smart grids are threatened by FDIA, this paper has designed DL-based detection algorithms that combine CNN with Bi-GRU to extract features from multiple high-dimensional input sequences and identify whether there are false data in the grid. To validate the FDIA detection performance of CNN-Bi-GRU model, five different detection models have been trained and validated on 15 data subsets using publicly available power system attack datasets, and the detection results of each model are evaluated with four statistical performance assessment metrics (i.e., accuracy, precision, recall, and F1-score). Furthermore, CNN-Bi-GRU has been compared with three machine learning models for detection performance. The simulation results showed that CNN-Bi-GRU outperformed CNN, LSTM, CNN-LSTM, KNN, LR, and SVM regarding detection performance because Bi-GRU can extract more data from the information of the forward and backward directions of the time series data.

References

[1] H. Karimipour, and V. Dinavahi, "On false data injection attack against dynamic state estimation on smart power grids," 2017 IEEE International Conference on Smart Energy Grid Engineering, pp. 388-393, 2017.

[2] H. Karimipour, A. Dehghantanha, R. M. Parizi, et al, "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids," IEEE Access, vol. 7, pp. 80778-80788, 2019.

[3] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," ACM Trans. Inf. Syst. Secur., vol. 14, pp. 1-33 , 2011.

[4] G. Luo, Y. Tan, M. Li, et al, "Stacked auto-encoder-based fault location in distribution network," IEEE Access, vol. 8, pp. 28043-28053, 2020.

[5] A. Pinceti, L. Sankar, O. Kosut, "Load redistribution attack detection using machine learning: A data-driven approach," 2018 IEEE Power & Energy Society General Meeting, pp. 1-5, 2018.

[6] A. Pinceti, L. Sankar, O. Kosut, "Detection and localization of load redistribution attacks on large-scale systems," J. Mod. Power Syst. Clean Energy, vol. 10, pp. 361-370, 2021.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436C444, 2015.

[8] Y. Wang, Z. Zhang, J. Ma, et al, "KFRNN: an effective false data injection attack detection in smart grid based on kalman filter and recurrent neural network," IEEE Internet Things J., vol. 9, pp. 6893-6904, 2021.

[9] W. Lei, Z. Pang, H. Wen, et al, "FDI attack detection at the edge of smart grids based on classification of predicted residuals," IEEE Transactions on Industrial Informatics, vol. 18, pp. 9302-9311, 2022.

[10] L. Yang, Y. Zhai, Z. Li, "Deep learning for online AC false data injection attack detection in smart grids: An approach using LSTM-autoencoder," J. Netw. Comput. Appl., vol. 193, no. 103178, 2021.

[11] D. Mukherjee, S. Chakraborty, A. Y. Abdelaziz, et al, "Deep learning-based identification of false data injection attacks on modern smart grids," Energy Rep., vol. 8, pp. 919-930, 2022.

[12] J. Q. James, Y. Hou, V. O. K. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," IEEE Trans. Ind. Inform., vol. 14, pp. 3271-3280, 2018.

[13] B. Xu, F. Guo, C. Wen, et al, "Detecting false data injection attacks in smart grids with modeling errors: A deep transfer learning based approach," arXiv preprint arXiv:2104.06307, 2021.

[14] S. H. Majidi, S. Hadayeghparast, and H. Karimipour, "FDI attack detection using extra trees algorithm and deep learning algorithm-autoencoder in smart grid," Int. J. Crit. Infrastruct. Prot., vol. 37, no. 100508, 2022.

[15] Y. Zhang, J. Wang, and B. Chen, "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach," IEEE Trans. Smart Grid, vol. 12, pp. 623-634, 2020.

[16] R. Qi, C. Rasband, J. Zheng, et al, "Detecting cyber attacks in smart grids using semi-supervised anomaly detection and deep representation learning," Information vol. 12, no. 328, 2021.

[17] H. Yang, R. Cao, H. Pan, et al, "Deep learning-based hybrid detection model for false data injection attacks in smart grid," The 6th International Conference on Industrial Cyber-Physical Systems, pp. 1-7, 2023.

[18] Z. He, J. Khazaei, F. Moazeni, et al, "Detection of false data injection attacks leading to line congestions using Neural networks," Sust. Cities Soc., vol. 82, no. 103861, 2022.

[19] O. Boyaci, A. Umunnakwe, and A. Sahu, et al, "Graph neural networks based detection of stealth false data injection attacks in smart grids," IEEE Syst. J., vol. 16, pp. 2946-2957, 2022.

[20] Y. Li, X. Wei, Y. Li, et al., "Detection of false data injection attacks in smart grid: A secure federated deep learning approach," IEEE Trans. Smart Grid, vol. 13, pp. 4862-4872, 2022.

[21] C. Nathwani, "Online signature verification using bidirectional recurrent neural network,"The 4th International Conference on Intelligent Computing and Control Systems, pp. 1076-1078, 2020.

[22] S. Wang, S. Bi, and Y. J. A. Zhang, "Locational detection of the false data injection attack in a smart grid: A multilabel classification approach," IEEE Internet Things J., vol. 7, pp. 8218-8227, 2020.

[23] K. O'Shea, and R. Nash, "An introduction to convolutional neural networks,"arXiv preprint arXiv:1511.08458, 2015.

[24] H. Hettiarachchi, and T. Ranasinghe, "Emoji powered capsule network to detect type and target of offensive posts in social media,"Proceedings of the International Conference on Recent Advances in Natural Language Processing, pp. 474-480, 2019.

[25] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example,"2020 International Workshop on Electronic Communication and Artificial Intelligence, pp. 98-101, 2020.

[26] Y. Deng, L. Wang, H. Jia, et al., "A sequence-to-sequence deep learning architecture based on bidirectional GRU for type recognition and time location of combined power quality disturbance," IEEE Trans. Ind. Inform., vol. 18, p. 4481-4493, 2019.

[27] H. Pan, X. Feng, C. Na, et al., "A model for detecting false data injection attacks in smart grids based on the method utilized for image coding," IEEE Syst. J., vol. 17, pp. 6181-6191, 2023.

[28] R. C. B. Hink, J. M. Beaver, M. A. Buckner, et al., "Machine learning for power system disturbance and cyber-attack discrimination," The 7th International Symposium on Resilient Control Systems, pp. 1-8, 2014.

[29] A. Ashok, M. Govindarasu, V. Ajjarapu, "Online detection of stealthy false data injection attacks in power system state estimation," IEEE Trans. Smart Grid, vol. 9, pp. 1636-1646, 2016.