EAI Endorsed Transactions
on Energy Web

Research Article EAI.EU

A Convolutional Neural Network Based Approach for Cyber Intrusion Detection in Smart Grids

Ying Lan[1,*], Chenye Zhu[1], Fan Wu[1], Qibin Hu[1]

[1]Logistics Engineering College, Shanghai Maritime University, 201306 Shanghai, China

Abstract

Smart grids, increasingly reliant on information and communication technologies (ICT), are vulnerable to complex cyberattacks, thereby mandating the deployment of intelligent and adaptable intrusion detection systems (IDS). However, the efficacy of existing IDS techniques is frequently constrained by their limited capacity to extract distinguishing features from the high-dimensional, heterogeneous data characteristic of grid operations. In order to overcome this, we suggest a novel intrusion detection model that uses a convolutional neural network (CNN) to automatically extract hierarchical features from network traffic. The suggested CNN model outperforms conventional signature-based and SVM-based techniques with an accuracy of 98.8%, precision of 98.6%, and recall of 99.3% using the KDD-CUP99 dataset. Validation on a semi-realistic dataset from the IEEE 14-bus system, which uses IEC 61850 communication protocols, shows that it is 97.3% accurate. This means that it works well when physical and cyber layers are combined. Feature importance analysis shows that cyber-layer features, such as the continuity of GOOSE sequence numbers, are very important for detection. This research introduces a feature-learning-based intrusion detection system (IDS) framework. It works well and shows potential for practical use in improving the cybersecurity of smart grids.

Keywords: Deep Learning (DL), Convolutional Neural Network (CNN), Intrusion Detection, Smart Grid

Received on 30 November 2025, accepted on 22 March 2026, published on 11 August 2026

doi: 10.4108/ew.13843

1. Introduction

The smart grid revolutionizes power generation, transmission, and consumption, giving us efficiencies we have never before seen in how we generate, transmit, and consume electricity. Thus, the smart grid is a combination of cyber and physical components that leverage modern information and communication technology (ICT) as part of the power infrastructure. However, this deep integration of communication networks with physical processes introduces a significant vulnerability — the exponential increase of an attack surface. As a result, cyber attacks against smart grid systems could impact the ability of the physical components to operate properly because the huge opportunity that exists for cybercriminals to carry out cyber attacks against smart grids has never been greater. Denial of service (DoS) attacks, false data injection attacks, and numerous other potential attacks exist. These can lead to a wide range of consequences, such as system instability of smart grid systems, disruption of control decisions, and, in very serious circumstances, widespread blackouts [1-3].

The cyber attack against the Ukrainian power grid in 2022 illustrates the real-world impacts that can exist due to such vulnerabilities [4]. The real experience that resulted from the Ukraine power grid technology shows a true need for intrusion detection systems (IDS) [5-6]. There is an important limitation that most current IDS have when implementing IDS within the smart grid because most IDS have difficulty managing the volume, complexity, and variety of cyber-physical data.

Traditional signature-based methods fail against novel or variant attacks, while conventional machine learning

*Corresponding author. Email: Lanying@shmtu.edu.cn

approaches rely heavily on manual feature engineering—a process that is not only labor-intensive but also struggles to capture the high-dimensional, spatiotemporal patterns inherent in grid operations.

Convolutional Neural Networks (CNNs), renowned for their ability to learn hierarchical features directly from raw data, offer a promising solution to these challenges. CNNs can automatically find local correlations and spatiotemporal dependencies in high-dimensional data because they have multiple layers of convolutional and pooling layers. This is not like traditional methods that require manual feature extraction. This capability is particularly well-suited to smart grid intrusion detection, where attack patterns often manifest as subtle, localized anomalies embedded in massive, heterogeneous data streams. By learning directly from grid traffic, CNNs can potentially overcome the feature engineering bottleneck that limits existing approaches. By establishing a Deep Learning (DL) model, the framework provides an automated extraction and classification of threat features based on the grid communication data, thus advancing security protection to intelligent and adaptive operation. To address the aforementioned challenges, this paper makes the following contributions:

1. Mitigating the feature extraction bottleneck - We have designed a custom CNN architecture based on the characteristics of smart grid traffic to automatically learn and hierarchically extract features from raw data without requiring any manual feature engineering, which is popularly required by conventional methods.

2. Achieving High Precision Anomaly Detection - We have built a deep-learning-based intrusion detection model that has a much higher degree of performance with regard to both accuracy, precision, and recall than existing methods (99%+ on all three measures of performance) than both signature-based and shallow machine learning models.

3. Validating the Model's Generalizability across Cyber-Physical Domains - In addition to validating the model on the various benchmark datasets, we also validated the model on a semi-realistic IEEE 14 bus dataset that applies IEC 61850 communication protocols and thus demonstrated that the model could be applied in a cyber-physical scenario where physical measurements are combined with cyber-layer traffic.

4. Providing Practical Deployability by Providing Interpretability - We performed a feature importance analysis of the detected cyber-physical features, such as the continuity of the GOOSE Sequence Number, as well as how those features might inform the efforts of reducing the model size to enable the model's deployment on resource-constrained edge devices in the smart grid environment.

Together, these contributions advance the state of the art in smart grid intrusion detection by demonstrating how a well-designed CNN model can overcome the feature engineering bottleneck while maintaining high accuracy and generalizability across cyber-physical domains. The proposed framework offers a methodological foundation for developing more intelligent and adaptive security solutions for industrial control systems. Future work will focus on model lightweighting for edge deployment, adversarial robustness against unknown attacks, and validation on real-world grid data.

2. Problem Description and Research Status in Intrusion Detection

To monitor cybersecurity activities on the smart/grid business/infrastructure, an Intrusion Detection System (IDS) is essential. Through continuous monitoring of network usage and activity, device status and vulnerabilities, as well as user activity, an IDS can help detect abnormal behaviour and send alerts immediately [7]. Although the increasing complexity of cyber-attack techniques, as well as the increased number and types of data on the smart/grid, will make it more difficult for current techniques to detect attacks [6]. This section reviews the state of the art in smart grid intrusion detection, organized by methodological families, and identifies the critical gaps that motivate this study.

2.1. Traditional Intrusion Detection Methods

Statistical modeling and signature-based methods were the foundation of early intrusion detection strategies. For known attack patterns, signature-based techniques—like the rule-based method put forth by Khan et al. [8]—achieve high detection accuracy; however, they are unable to detect new or variant attacks, necessitating regular manual updates to the rule database. A statistical machine learning-based defense mechanism for smart grid cyber-physical systems was presented by Singh et al. [9] in order to overcome this limitation. The defense mechanism uses anomaly detection to find departures from expected behavior.

Limitations Analysis: While the techniques above provide a foundation for intrusion detection, they share a commonality in that they primarily depend on feature creation and existing rules through human intervention. As a result, these methods do not adapt to the rapidly changing dimensionality and dynamism of the data associated with smart grids. In addition, these approaches tend to be ineffective as attack patterns vary. As attack patterns evolve, these approaches suffer from poor generalization and high maintenance overhead.

2.2. Machine Learning-Based Detection Methods

The emergence of big data has propelled machine learning (ML) into a dominant position in intrusion detection systems. ML techniques possess the capacity to autonomously discern attack patterns from extensive datasets, thereby facilitating the identification of previously unknown threats. Frequently employed algorithms encompass K-means clustering, support vector

machines (SVM), random forests, and a range of neural network designs [10–14].

ML-based techniques have been applied to a variety of attack scenarios in the field of smart grids. In particular, researchers have developed data-driven detection frameworks for false data injection attacks (FDIA), which jeopardize state estimation and automatic generation control [15, 16]. Beyond FDIA, other attack types such as denial-of-service (DoS) and adversarial attacks have also been recognized as critical threats to smart grid security [17–20]. Cui et al. [21] proposed a self-healing solution integrating detection, data repair, and control for load frequency control. Shangguan et al. [22] utilized time-series neural networks for high-precision detection of multi-point FDIA. Zhang et al. [23] introduced a whale optimization algorithm-based method to identify unobservable FDIA.

Limitation Analysis: Despite their success, traditional machine learning methods face significant challenges when used for smart grid intrusion detection. First, many machine learning algorithms require extensive feature engineering, which is time-consuming and might not capture complex patterns related to space and time. Second, these methods often struggle with data that is both high-dimensional and varied, combining physical measurements with communication traffic. Finally, their performance decreases when they encounter new types of attacks.

2.3. Deep Learning for Intrusion Detection

Deep learning (DL) models have emerged as powerful tools for intrusion detection due to their ability to learn hierarchical features directly from raw data. Musleh et al. [24] designed an LSTM-based unsupervised learning model for detecting various FDIA types. Lu et al. [25] employed representation learning and convolutional neural networks (CNNs) for intelligent attack localization and system recovery in cyber-physical power systems.

Model optimization has also attracted considerable attention. Zhu et al. [26] developed a lightweight detection structure using early exit mechanisms and hybrid precision quantization, enabling fast attack localization. Yang et al. [27] proposed NIDS-CNNRF, an ensemble model combining CNN and random forest, which improved multi-class attack detection accuracy compared to existing models.

Limitation Analysis: Although DL models are better at learning features, there are still a number of drawbacks. First, the majority of current DL-based IDS are tested on benchmark datasets (like KDD-CUP99) that do not have grid-specific communication protocol features, which raises concerns about their practicality. Second, the interpretability of these models is often disregarded, making it difficult to understand which features affect detection decisions — a critical requirement for implementation in safety-critical settings.

2.4. Graph-Based Methods for Topology-Aware Detection

Graph neural networks (GNNs) have been used in recent studies to model power grid structures. This method makes it possible to identify and localize attacks more precisely. Vincent et al. [28] introduced a graph convolutional network (GCN)-based location detection system that identifies attack sources by analyzing state estimation deviations. Han et al. [29] converted raw data into graph-topology-structured data and employed multi-graph mechanisms with temporal correlation layers to enhance feature extraction. Protection techniques against graph-based false data injection attacks were further investigated by Morgenstern et al. [30].

Researchers have integrated GNNs with time-series models to capture spatiotemporal dynamics. Peng et al. [31] proposed an online localization technique that transforms time-series measurements into time-frequency representations, jointly capturing spatiotemporal correlations. Li et al. [32] adopted gated GNNs with attention mechanisms to assign different aggregation weights to adjacent nodes. For near-real-time attack localization, Presekal et al. [33] integrated deep convolutional networks with graph convolutional long short-term memory networks (GC-LSTM).

Efforts to improve generalization and efficiency have also been reported. Takiddin et al. [34] developed a GNN approach designed to enhance detection against unknown attacks. DSE-BiLSTM, a streamlined model, was introduced by Zhu et al. [35], employing variable graph attention autoencoders to extract topological characteristics from crucial nodes.

A limitation of existing approaches is their tendency to prioritize physical measurements, such as voltage and power flow, while neglecting the communication layer, despite the advantages of GNN-based methods in utilizing topological data. Conversely, modern smart grids, which are cyber-physical systems, frequently experience attacks that initially present as anomalies in communication traffic, such as GOOSE message irregularities, before impacting physical operations. Consequently, the integration of cyber-layer features with physical topology remains an area that warrants further investigation.

2.5. Distributed and Privacy-Preserving Detection

To address data silos and privacy concerns, distributed learning architectures have gained traction. Kausar et al. [36] developed a federated deep learning-based decentralized detection system that preserves data privacy. Kesici et al. [37] constructed a collaborative detection system using vertical federated learning, integrating attention modules and bidirectional LSTM for FDIA detection in distribution networks. Kesici et al. [38] also suggested a split learning-based cloud-edge cooperative

framework that combines edge feature extraction and cloud-based global feature aggregation.

Limitation Analysis: Although these methods solve significant real-world issues, they make model coordination and training more difficult. Furthermore, they don't automatically solve the fundamental problem of successfully integrating physical and cyber characteristics for intrusion detection.

2.6. Summary of Research Gaps

This study is prompted by three significant deficiencies identified in the preceding review:

First, there is a lack of integrated cyber-physical feature learning. Existing methodologies predominantly concentrate on either physical measurements or network traffic independently. Consequently, models capable of concurrently learning from both domains to fully represent the comprehensive character of smart grid attacks are scarce.

Second, insufficient interpretability is a concern. Although deep learning models demonstrate high accuracy, their "black-box" characteristic undermines trust and hinders deployment in environments where safety is paramount. Few investigations systematically examine the features that most significantly influence detection decisions, especially within cyber-physical systems.

Finally, there is limited validation on realistic, protocol-aware datasets. Benchmark datasets, such as KDD-CUP99, do not incorporate grid-specific communication protocols, such as IEC 61850 GOOSE/SV. Effectiveness in real-world smart grid environments where cyber-physical interactions are important is not guaranteed by validation on such datasets.

This study suggests a CNN-based intrusion detection model that fills these gaps by: (1) automatically learning hierarchical features from integrated cyber-physical data; (2) offering interpretability through feature importance analysis; and (3) validating performance on a custom-built IEEE 14-bus dataset that incorporates IEC 61850 communication protocols and a benchmark dataset.

3. CNN-Based Intrusion Detection Model

CNN refers to a hierarchical feedforward neural architecture made up of convolutional layers, pooling layers, and fully connected layers in most cases. Convolutional layers extract local features, pooling layers reduce the dimensionality of features, and fully connected layers perform recognition or classification. The schematic diagram of this architecture is presented in Figure 1. CNNs emulate the structure of and the operational mechanism of biological visual cortex by filtering significant representative features with the objects' local receptive field and weight sharing mechanisms. This, in turn, lowers the number of parameters and elevates computational performance.

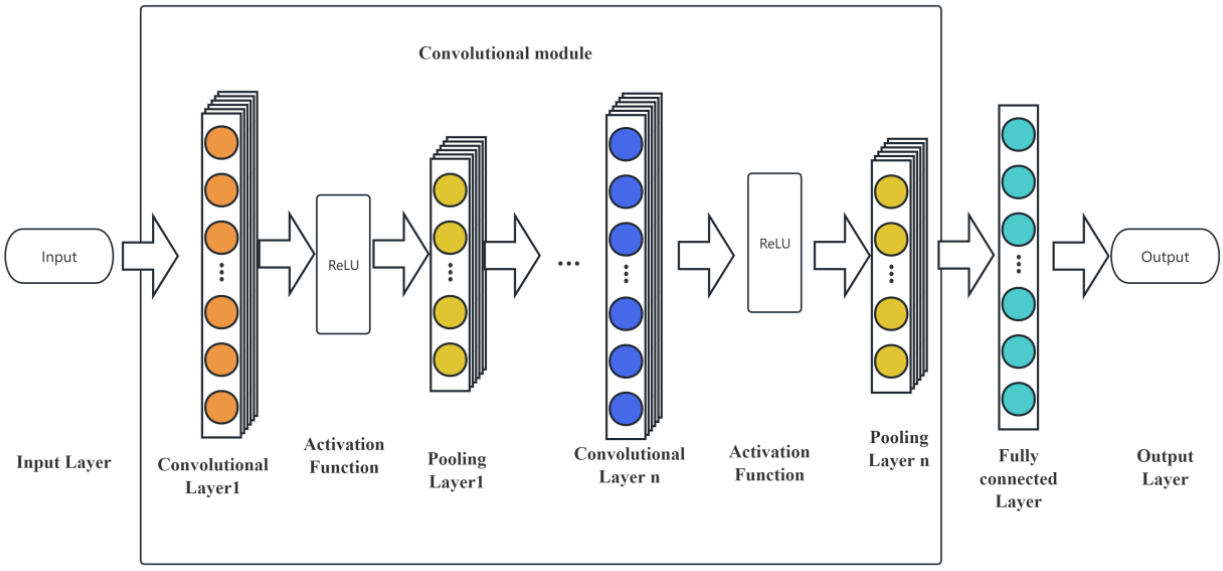

Figure 1. Schematic Diagram of CNN Architecture

3.1. Architecture of the Intrusion Detection Model

The CNN-based IDS model introduced in this research comprises three primary components:

1. Data Processing Module: This module preprocesses raw network data by removing noise and inconsistencies through numerical encoding and normalization. Encoding transforms categorical features into numerical values, while normalization processes numerical values to ensure comparable scales and appropriate formats. Ultimately, these operations guarantee the inputs' readiness to be fed into the CNN for training.

2. Intrusion Detection Module: As the analytical module and core of the CNN architecture, this component targets the processing of the preprocessed data. The model is trained using the training data and then tested on the testing datasets for intrusion detection. Subsequently, the results are forwarded to the model performance evaluation module.

3. Model Evaluation Module: The module measures the model performance on evaluation metrics like accuracy, precision, false negative rate (FNR), and training loss. Through repeated adjustments of the parameters of the training loss model, accuracy and precision are maximized, while minimizing the FNR causes the best detection results.

The structure of the proposed model, along with its operation flow, is depicted in Figure 2.

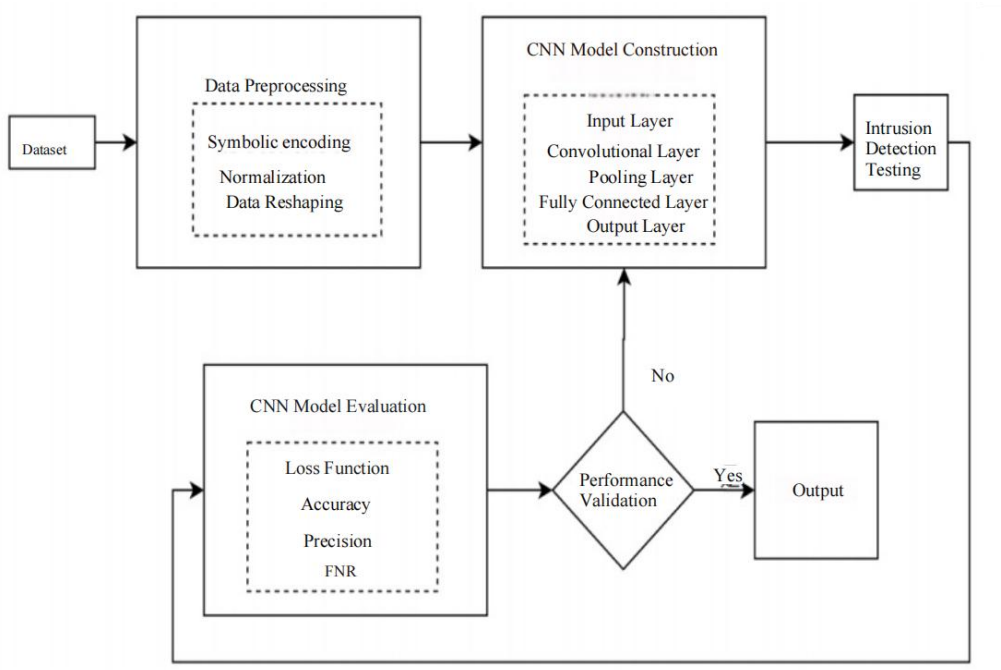

Figure 2. Architecture and operational workflow of the CNN-based intrusion detection model

3.2. CNN-Based Intrusion Detection Algorithm and Parameter Configuration

CNN could be structured with multiple convolutional epochs and pooling layers. The structure and parameter configuration of the model below could be shown:

1. Input Layer: The input layer takes as input the preprocessed feature vectors of network traffic, which transform the raw network data into the two-dimensional matrix suitable for the convolutional operations.

2. Convolutional Layer: The feature extraction from the input signals is performed at this layer using the convolutional kernels and the activation functions. It extracts the characteristic patterns at different scales using kernels of different sizes, which enable it to detect the anomalous traffic and unknown attacks on the specified features. In the case of power grid data, the amount of data is high dimensional; hence, the 5×5 convolution kernel is employed in this study to extract larger areas of local features, which enables enhanced detection of regional anomalies. The 5×5 kernel extracts more information about the context than 3×3 kernels in shallow layers and enables them to detect attacks that are spatially correlated more effectively than the 3×3 kernels. It is a method to attain feature extraction capability and model efficiency.

The size of the convolutional kernels adopted in the study is 5×5, and the stride is equal to 1. The following (Equation 1) helps in computing the output of the convolutional layer:

$$Z_i^k = f\left(W_i^k \times Z_{i-1}^k + b_i^k\right) \quad (1)$$

where Z_i^k is the feature map computed by the k-th convolutional kernel of the i-th layer, W_i^k is the weight matrix of the kernel, b_i^k is the corresponding bias, whereas f denotes an activation function applied to perform the nonlinear transformation.

3. Activation Layer: The activation functions introduce the nonlinearity to enhance the model's generalization capability. Tanh, Sigmoid, and ReLU (Rectified Linear Unit) are common activation functions for CNNs. The ReLU function (expressed in Equation 2) is used in this study:

$$f(x) = \begin{cases} \max(0, x), x \geq 0 \\ 0, x < 0 \end{cases} \quad (2)$$

while its derivative (Equation 3):

$$f'(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \quad (3)$$

The ReLU function produces a gradient equal to zero for x < 0, which means that the neurons do not participate in training if the outputs fall into the saturation region. As a result, the network has sparse representation capability. The derivative is constant equal to one for x > 0, which provides the faster convergence of the network and solves the vanishing gradient problem.

4. Pooling Layer: After convolutional processing, a pooling layer is applied to enhance performance by limiting over-fitting and improving generalization. Pooling helps in down-sampling the feature maps, which decreases the dimensionality and computational burden. Among various pooling techniques, max pooling and average pooling are the most commonly used methods, where the disparity lies in choosing the value from the pooling window. Max pooling selects the maximum value in the window, catering to the most representative features, while average pooling calculates the mean, which may smooth over important variations. Since this research focuses on achieving sensitive and prompt detection of changes to features caused by an attack, max pooling was selected to preserve the critical information in features while limiting the broad propagation of extraneous gradients. The selection mechanism is demonstrated in Figure 3.

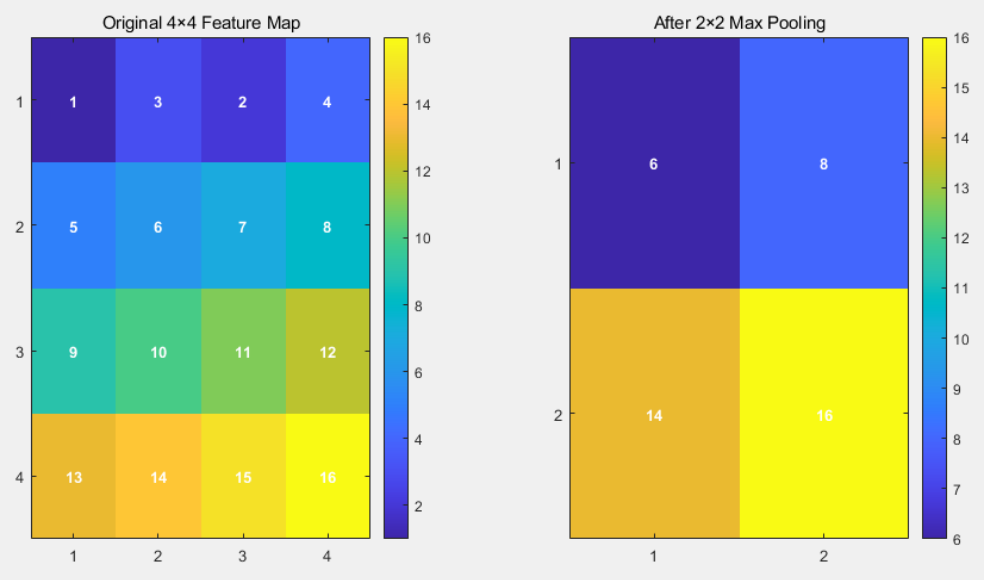

Figure 3. Value selection strategy in max pooling

The pooling operation reduces the spatial dimensions of feature maps, retains critical feature representations, and decreases computational and memory requirements.

5. Fully Connected Layer: The pooling operation reduces the spatial dimensions of feature maps, retains important representations of features, and decreases the computational and memory burden. 5. Fully Connected Layer: This layer serves as the last stage of the network, where classification is achieved by aggregating and flattening the distributed features extracted in the preceding layers. The flattened vector is mapped to the output space using the Softmax function defined in Equation 4:

$$P(y|x) = \frac{e^{h(x,y_i)}}{\sum_{i=1}^{n} e^{h(x,y_i)}} \quad (4)$$

For easier real-world use, the output layer was set as a two-dimensional vector with "normal" and "anomalous" states, given that cyberattacks happen intermittently, and there are economic and other considerations for reporting them. The Softmax function then transformed them into probability values that the system admins can read.

6. Loss Function: The loss function measures how far the predictions are from the true values. During model training, it is aimed at reducing this loss through ongoing refinement of parameters. Smaller values of loss indicate

better performance. The chosen loss function was the cross-entropy, given by (Equation 5):

$$L(y, \dot{y}) = -\sum ylog\dot{y} \quad (5)$$

where y_i and $\hat{y}_i$ are the true and predicted labels, respectively.

4. Experimental Results and Analysis

Data-set choices and construction for smart grid intrusion should consider both the operational features of power systems and the cyberattack types. Due to privacy and geographical limitations of actual grid data, this work chose the trending public data-set KDD-CUP99 [39] for the training and testing of the model for accessibility and reproducibility. Based on this dataset, a CNN model was deployed using the PyTorch deep learning framework, along with software simulation of the environment for testing the performance of the proposed intrusion detection algorithm. Given that KDD-CUP99 is a classic benchmark data-set that shares core attack types (e.g., DoS, probing) with basic threats such as communication disruptions and unauthorized access in smart grids, it was chosen for initial algorithm validation and comparison with baselines. However, its traffic features and data distribution are far from modern power grid real-use situations. Therefore, after the initial treatment, this choice will be followed up with systematic validation using other datasets, such as NSL-KDD (which removes data redundancy), CICIDS2017 (which contains modern attack traffic), and grid-specific datasets (e.g., power system false data injection datasets), to monitor the generalization capacity and applicability of the model in real use cases.

4.1. KDD-CUP99 Data-set and Data Preprocessing

The KDD-CUP99 data-set assigns each record of network connection structure a normal label or an attack label to enable basic intrusion detection and analysis. This attack classification encompasses 39 individual attack types, which can be classified into four main categories, as shown in Table 1. Each node faces varying attack threats when a network transmits. For example, data senders are less vulnerable to Denial-of-Service (DoS) and Probing attacks, while the receivers are more susceptible to Remote-to-Local (R2L) and User-to-Root (U2R) attacks. Therefore, accurate determination of the attack category is indispensable for the next security operation in terms of mitigation strategy development.

Table 1. Primary network intrusion categories

Intrusion Category	Attack Type	Specific Behaviors (including but not limited to)
DoS	Denial-of-Service (DoS) Attack	Transmission of excessive DoS packets that deplete network bandwidth and exhaust system resources
R2L	Unauthorized access from a remote host	Illegal access initiated from external computers
U2R	Unauthorized access to local superuser privileges	Exploitation of low-privilege accounts to bypass authentication and gain administrative control
Probing	Port scanning or surveillance	Scanning of network information to identify vulnerabilities

The KDD-CUP99 dataset includes 41 fixed features (38 numerical features and 3 symbolic features) plus one label attribute, amounting to 42 features. The features are presented in the data format in Figure 4.

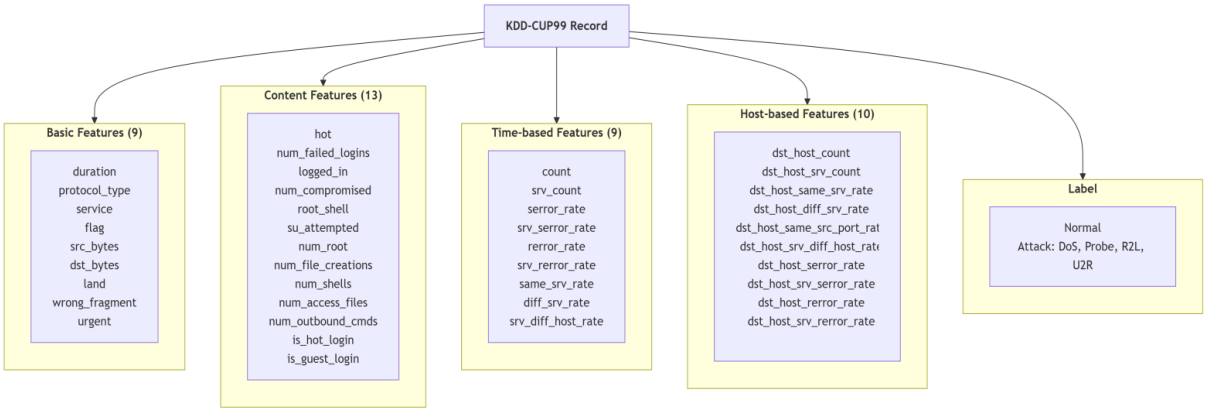

Figure 4. Data Format of KDD-CUP99 Dataset

Preprocessing was done in the following steps:

1. Numerical Encoding of Symbolic Features: Unable to target symbolic features in a neural network, the second, third, and fourth features of the KDD-CUP99 dataset were numerically encoded. This study implemented One-Hot Encoding for the conversion. For example, the protocol_type feature, which has three denominations (tcp, udp, and icmp), was converted into three vectors: [1, 0, 0], [0, 1, 0], and [0, 0, 1]. The service feature (network service type of the target host) and the flag feature (the connection status) had 70 and 11 value ranges, respectively, and were also converted into 70-dimensional and 11-dimensional vectors. These three one-hot encoded vectors were concatenated into an 84-dimensional vector representing the symbolic features. This 84-dimensional vector was then combined with the remaining 38 numerical features, resulting in a final feature vector of 122 dimensions for each network connection record.

2. Normalization of Numerical Features: Numerical features were established for both continuous and discrete types. Discrete numerical features, which are mainly indicators of 0 or 1, were directly applied to the neural network. However, continuous features differ in scale and range. To avoid the consequences of high-magnitude features' excelling and getting rid of scale effects, normalization was performed according to Equation 6:

$$x = \frac{x - min}{max - min} \tag{6}$$

where max(x) and min(x) show the maximum and minimum values of each feature, respectively.

For this study, we utilized the standard 10% KDD-CUP99 training subset (kddcup.data_10_percent.gz), which contains 494,021 records, for model training and validation. For testing, we employed the corrected KDD-CUP99 test set (corrected.gz), comprising 311,029 records. This test set includes 14 additional attack types not present in the training data, providing a rigorous evaluation of the model's generalization capability. Among the test samples, approximately 43.1% were designated normal, while 56.9% were classified as anomalous according to how often attacks manifest in real network environments.

4.2. Simulation Environment and Parameter Configuration

The idea of a particular experimental platform was presented to verify how well CNN-based intrusion detection models fit into smart grids. The hardware and software components were included in the platform to stimulate deep learning and train intrusion detection models, and the computational power was available for testing. The hardware configuration of the simulation platform is presented in Table 2 to implement deep learning and data mining in offline environments and train high-performance intrusion detection models.

Table 2. Hardware configuration of the CNN-based intrusion detection simulation platform built on PyTorch

Component	Environment / Version
Operating System	Windows 11
CPU	Intel Core i7-10700
Memory	32 GB
GPU	NVIDIA GeForce RTX 2070 Super
Development Environment	Python 3.6

The CNN model in this study used two convolutional layers, each with a 5 × 5 kernel, and two pooling layers, each with a 2 × 2 kernel. The intention was to transform the original network features into a higher-dimensional representation. The parameter setup and the associated analysis are described next:

1. Epoch: Here, the epoch represents the number of times the training dataset passes through during the training process. Although a larger number of epochs optimizes the model training, it also extends the computation time. The trend of convergence of model loss with successive epochs is depicted in Figure 5.

As observed in Figure 5, after about 150 epochs, the training loss dropped below 0.04 and stayed at that level. As such, the number of epochs was specified as 150 for the model.

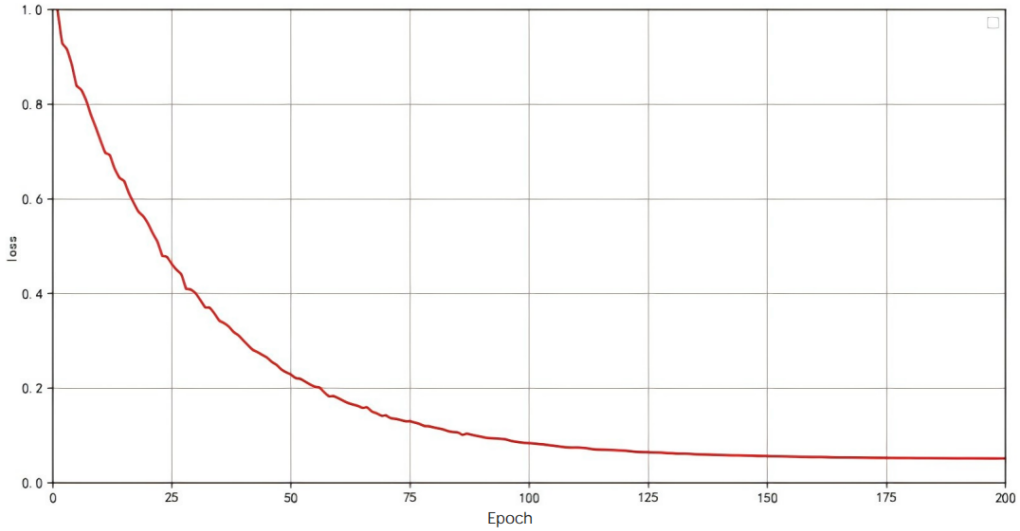

Figure 5. Convergence of training loss with increasing epochs

2. Learning rate is an important training parameter that determines how fast neural networks converge during training. If the learning rate is very low, the convergence will be very slow; however, if the rate is too high, the approach will either not converge or lose convergence stability. The study considered learning rates of 0.01, 0.001, and 0.0001. The corresponding loss variation curves are contained in Figure 6.

In Figure 6, the learning rate was shown to have a major effect on convergence stability and performance. A learning rate of 0.001 demonstrated faster convergence as well as stable performance and was subsequently chosen as the value for model training optimization.

3. Batch Size: Batch size, which is a hyperparameter in SGD, determines how many samples pass through before each update of the parameter. It is correlated with training quality and convergence speed. While larger batch sizes increase throughput per iteration, they demand more computational resources and can sometimes result in memory saturation. On the other hand, very small sizes prolong training time and increase the impact of exceptional samples. This research used batch sizes of 50, 60, 70, and 80; the corresponding training loss curves are illustrated in Figure 7.

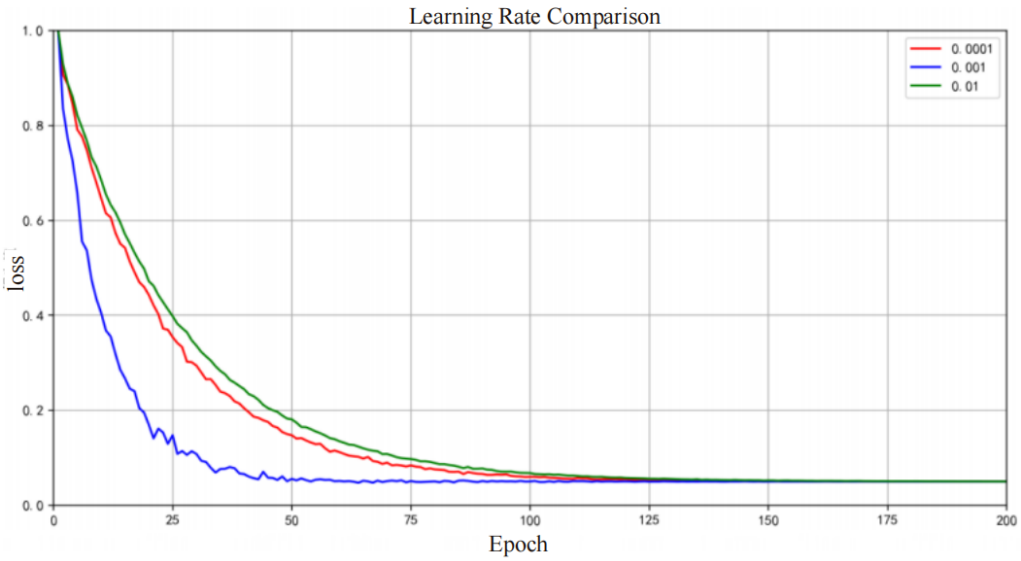

Figure 6. Training loss curves under different learning rates

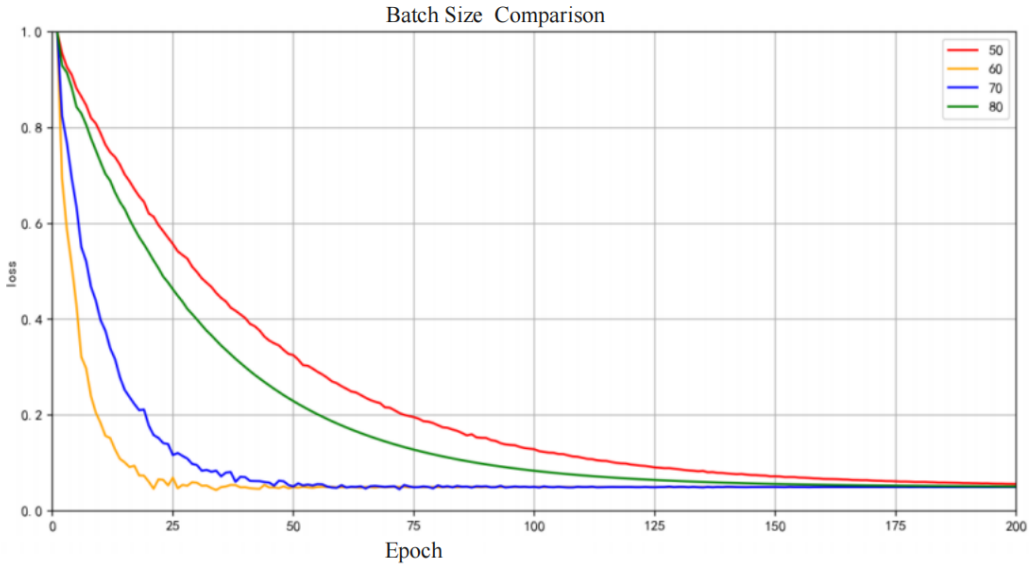

Figure 7. Training loss curves under different batch sizes

Upon viewing the training loss in Figure 7, which is non-linear, the model converges most efficiently with the batch size of 60 among the ones tested, showing the fastest decay of loss. Hence, the batch size is defined as 60.

Finally, the CNN model is tuned with the parameters of a learning rate of 0.001, a batch size of 60, and a total of 150 epochs.

4.3. Evaluation Metrics

The intrusion detection performance evaluation metrics are defined based on the sample category and the judgment of the system: positive/negative (true or false). To recap:

1. True Positive (TP): An actual attack that is identified as one.

2. False Positive (FP): A non-attack that is wrongly categorized as an attack.

3. True Negative (TN): A non-attack that gets identified as a non-attack.

4. False Negative (FN): An attack that is missed by the system.

These four possible outcomes can be summarized by a confusion matrix, which is presented in Table 3. The elements of this matrix computing intuitive and quantitative model performance evaluation.

Table 3. Confusion matrix for result classification

Actual Label	Predicted Label	
	Attack	Normal
Attack	TP	FN
Normal	FP	TN

The sum of all elements equals the number of samples; where TP and TN represent correctly classified entries.

The program's runtime was measured, and five evaluation metrics—Accuracy, Precision, FNR, Recall, and F1-score—were used to test the intrusion detection model. Equations 7 to 11 show the methods for calculating each metric.

1. Accuracy: The number of samples that were classified well from all samples (both normal and intrusive) divided by the total number of samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

2. Precision: The percentage of true positives among all reported positives, showing how accurate the alarms are:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

3. FNR: The percentage of genuine intrusion samples labeled as normal, a measure of how serious the failed detections are:

$$FNR = \frac{FN}{TP + FN} \tag{9}$$

$FNR = \frac{FN}{TP+FN}$4. Recall: The percentage of all intrusion samples successfully detected by the model, showing how sensitive the detection is:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$Recall = \frac{TP}{TP+FN}$ 5. F1-score: The harmonic mean of Precision and Recall, a balanced performance score for situations involving unbalanced datasets:

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

4.4. Results Analysis and Discussion

A thorough assessment of deep CNN models must be carried out in order to confirm their real-world efficacy through smart grid IDS. This paper evaluated the designed CNN architecture in terms of six indicators: precision, accuracy, FNR, recall, F1-score, and runtime.

To confirm the evaluation's strength, we carried out comparison experiments with two other baseline methods of IDS, which are traditional signature-based methods and SVM. The detection results for the three techniques are presented in Figure 8.

The evaluation metrics are calculated using Eqs. 7 to 11 are summarized in Table 4.

Table 4. Comparison of detection evaluation metrics across the three methods

Method	Runtime (s)	Accuracy (%)	Precision	FNR	Recall	F1-score (%)
Signature-based	1005.6	94.8	93.98	7.00	93.0	93.5
SVM	1234.2	96.4	97.44	3.00	97.0	97.2
CNN	1253.8	98.8	98.61	0.70	99.3	98.96

*Note: Runtime refers to the total inference time on the full test set (311,029 records). The average per-record detection time for the CNN model is approximately 4.03 ms.

The results demonstrate that the proposed CNN architecture achieved a balanced performance on smart grid IDS. Using the test set, CNN achieved an accuracy of 98.8%, which is higher than that of classical ML models. Its 98.61% precision and 99.3% recall mean high alarm quality reliability and detection sensitivity. The 98.96% F1-score indicates a balanced completeness-accuracy trade-off. In addition, the model FNR of 0.70% is also lower than the SVM (3%) and the signature-based (7%).

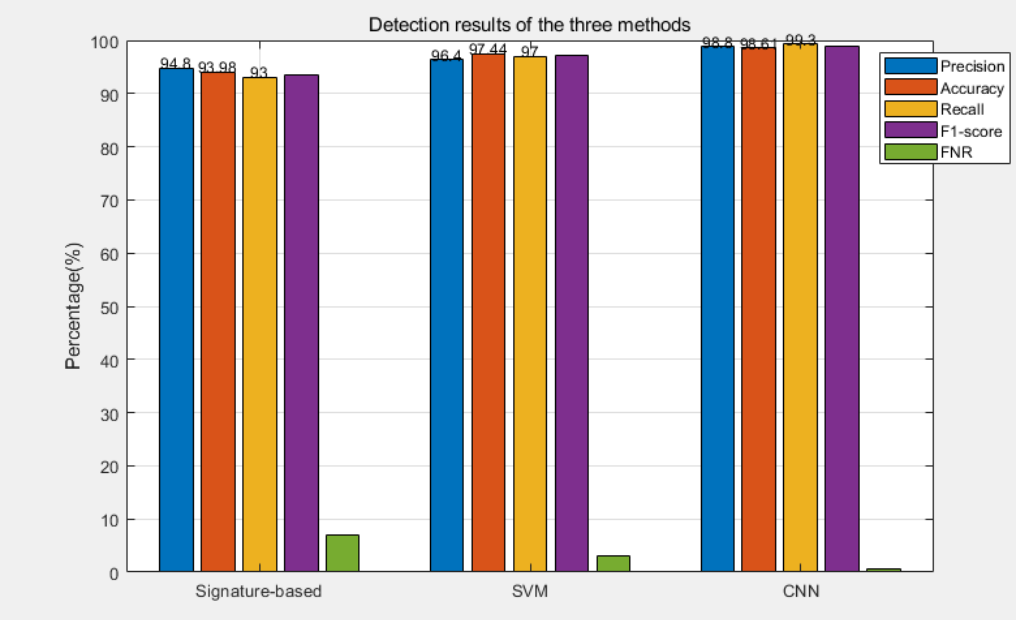

Figure 8. Detection results of the three methods

The CNN has a slightly longer runtime than other methods, but it can provide for automated feature extraction, which compensates for the gap. The signature-based method can be quickly executed with manual rules imposed on each of the 41 features; however, the costs of operations are high. Its rules are also static and do not adapt to changes in the attack. The SVM method is not appropriate for real-time grid monitoring due to its complexities of high-dimension data.

Conversely, CNN builds up efficient feature representations itself through local connectivity, translation invariance, and weight sharing. These characteristics enable the model to capture discriminative features out of high-dimensional data, resulting in high generalization, adaptability capabilities, and following smart grid dynamic changes.

Figure 9 provides a visual representation of the CNN model's classification performance on the test set, specifically through its confusion matrix. Considering the overall performance metrics detailed in Table 4 and the test set's composition of 311,029 records (comprising roughly 134,054 normal instances and 176,975 attack samples), the

confusion matrix yields the following approximate values: true positives (TP) ≈ 175,700, false negatives (FN) ≈ 1,240, false positives (FP) ≈ 2,480, and true negatives (TN) ≈ 131,600.These values yield an accuracy of 98.8%, precision of 98.61%, recall of 99.3%, and an FNR of 0.70%, which are consistent with the metrics reported in Table 4 after accounting for rounding. The matrix shows that the model misclassifies only a small number of attack and normal samples, confirming its reliability.

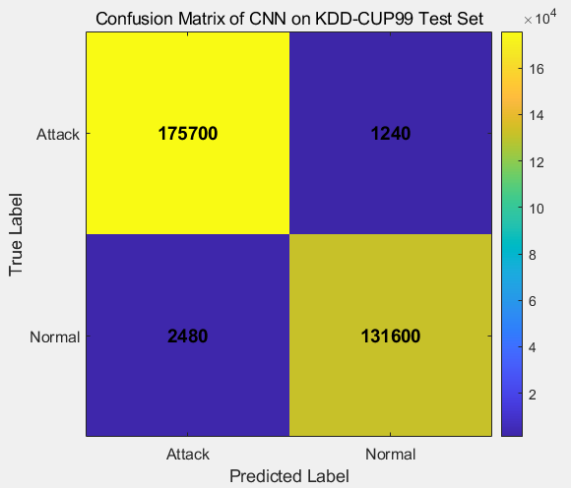

Figure 9. Confusion matrix of the CNN model on the KDD-CUP99 test set

Figure 10 presents a radar chart, facilitating a thorough comparison of the three methodologies across five key metrics: accuracy, precision, recall, F1-score, and the inverse of the false negative rate (100 – FNR). The CNN demonstrates the most extensive area on the chart, surpassing both the SVM and signature-based approaches across all assessed dimensions. This is particularly evident in recall and FNR, thereby underscoring its advantageous equilibrium between detection efficacy and the rate of false alarms. To gain insight into the feature learning capability of the CNN, we applied t-SNE to the outputs of the last fully connected layer.

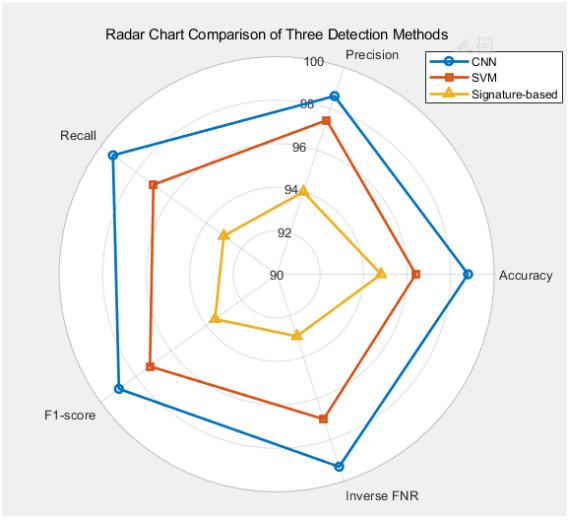

Figure 10. Radar chart comparison of detection methods

The 2D embedding displays the formation of two separate clusters that correspond to attack and normal samples, as shown in Figure 11. This clear distinction shows that the CNN can effectively learn discriminative features, which supports its high detection accuracy.

Regarding the training process, Figure 5, previously presented in Section 4.2, depicts the convergence of training loss across epochs, stabilizing below 0.04 after 150 epochs. Furthermore, Figures 6 and 7, which examine the influence of learning rate and batch size on convergence, identify the optimal configuration as a learning rate of 0.001 and a batch size of 60. In addition, Figure 12 demonstrates that the training and validation accuracy curves converge above 97%, with a minimal gap, which indicates robust generalization to novel data and the absence of overfitting.

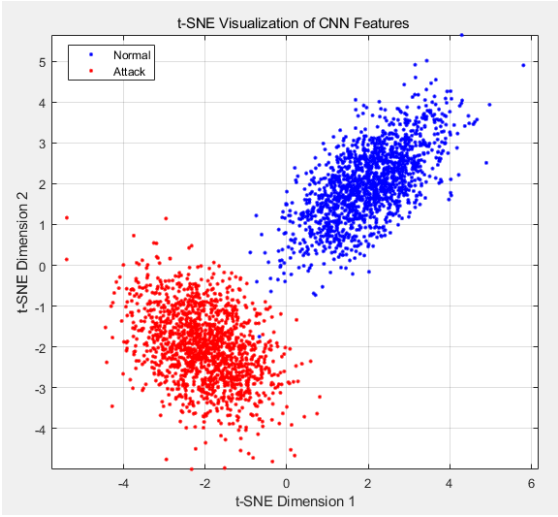

Figure 11. t-SNE visualization of CNN features

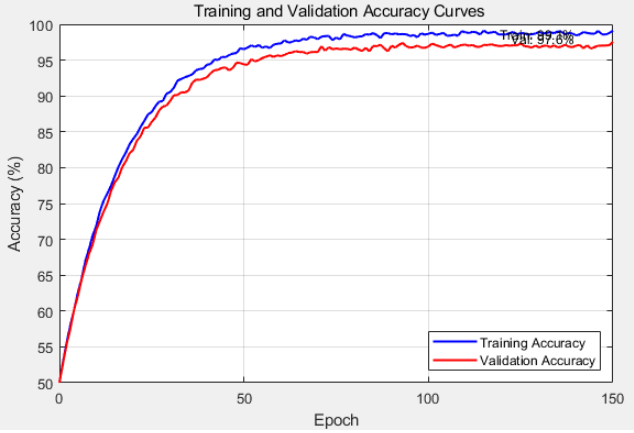

Figure 12. Training and validation accuracy curves

The total inference time on the full test set (311,029 records) is approximately 1,254 seconds, yielding an average detection latency of 4.03 milliseconds per record. This per-record latency satisfies the fundamental real-time monitoring requirements of smart grids (typically within a few milliseconds). However, as grid scale expands and traffic volume increases, computational and transmission loads may challenge system scalability. Future work will focus on model lightweighting techniques — such as pruning, quantization, and edge-cloud collaborative deployment — to reduce detection latency further and enable large-scale deployment in resource-constrained environments.

The per-record latency of 4.03ms already meets fundamental real-time monitoring requirements, although the model was initially developed for offline validation. With an increasingly expanded grid scale, demands for computational and transmission loads will compromise system scalability. We have outlined three practical pathway solutions to enable the optimal deployment. The first level is algorithmic, aimed at reducing computational overhead through model pruning and quantization. The second is an architectural level, focusing on the implementation of edge-cloud collaborative systems for data hierarchical processing. The third is a hardware development level that will promote inference through dedicated AI chips. The next aim is to bring the detection latency down to under a second, followed by testing the solution in actual grid settings to promote the digital technology from algorithmic research to engineering practice.

4.5. Supplementary Experiments Based on the IEEE 14-node Standard Test System

To further validate the model's effectiveness in smart grid scenarios integrating ICT features, we constructed a semi-real dataset based on the IEEE 14-node standard test system and conducted supplementary experiments on it. The IEEE 14-node system, a common benchmark for studying false data injection attacks [15-16], includes 14 busbars, 5 generators, and 20 transmission lines. By integrating communication network simulations onto this physical topology, the study generated hybrid data that incorporates both physical constraints and ICT characteristics.

4.5.1 Dataset Construction

The dataset was constructed as follows.

(1) Physical Layer Simulation: The MATLAB Power System Simulator (MATPOWER) generates operational data, including node voltage amplitudes (0.95–1.05 pu), phase angles (-20° to 10°), branch active power (0–200 MW), and reactive power (0–100 Mvar), through power flow calculation and state estimation. Figure 13 depicts the system topology, which consists of 20 branches and 14 nodes. Nodes 2, 3, 6, and 8 are generator nodes, and Node 1 is the balancing node. To assess the impact of different attack types on the system, injection points for simulated attacks, such as false data injection, GOOSE replay attacks, SV message tampering, and DoS attacks, are set up at Nodes 4, 6, 9, and 12, respectively.

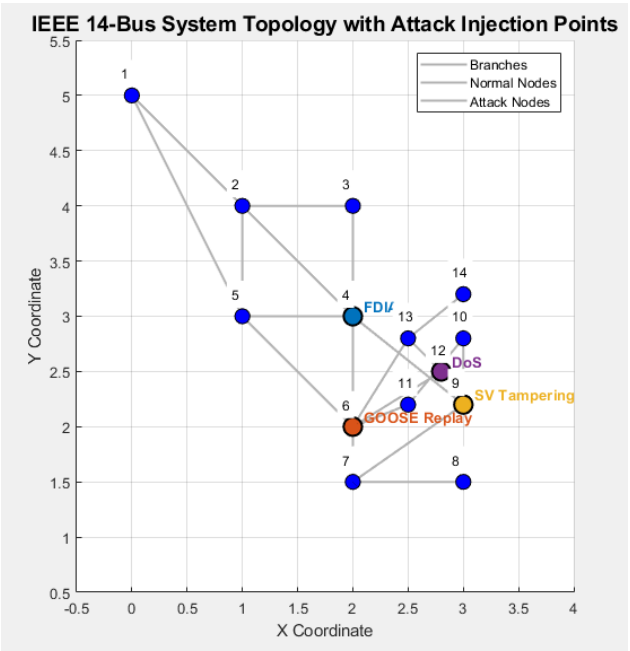

Figure 13. IEEE 14-Bus System Topology with Attack Injection Points

(2) Communication Layer Simulation and ICT Feature Extraction: By integrating IEC 61850 protocol simulation on top of the physical layer, three typical message types are generated: Sample Value (SV) messages (80 points per cycle), GOOSE (General Object Oriented Substation Event) messages (triggered by status changes), and Manufacturing Message Service (MMS) control messages. For these

protocols, the following ICT features are extracted for intrusion detection:
Statistical characteristics of message timestamp intervals (mean, variance, and abnormal intervals);
The continuity of the status number (StNum) and sequence number (SqNum) in GOOSE messages;
Sampling count continuity in SV messages;
Quality bits in the protocol data unit (validity, overflow, synchronization flag).

Figure 14 illustrates the typical structure of IEC 61850 GOOSE and SV messages.

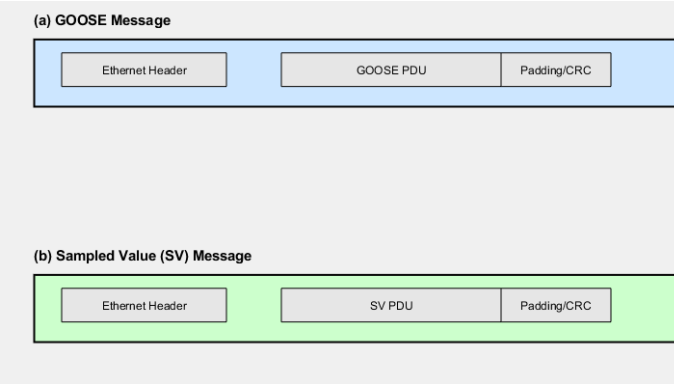

Figure 14. Data Format of IEC 61850 Protocol (GOOSE and SV)

(3) Attack scenario injection: three types of attack scenarios were designed, with the protocol attack category further divided into three subtypes:
False Data Injection Attack (FDIA): Injecting 5% to 20% deviation into state estimation values to simulate the attack pattern described in references [15-16].
Protocol attacks encompass GOOSE replay attacks, SV packet tampering attacks, and connection loss attacks.
Denial of Service (DoS) attacks, which impede communication channels via flood attacks, are employed to simulate communication disruptions within smart grids.

(4) Dataset composition: a total of 5,000 samples were generated through a joint simulation utilizing OPNET and MATLAB; these samples comprised 60% normal data and 40% attack data. Each sample incorporated 28-dimensional electrical characteristics, 10-dimensional communication characteristics, and a singular attack label.

4.5.2 Validation results on IEEE 14-node dataset

The trained CNN model was evaluated through transfer learning validation on the IEEE 14-node semi-real dataset, with results presented in Table 5 and Figure 15.

Table 5. Detection performance on the IEEE 14-node dataset

Attack type	Precision (%)	Recall (%)	F1-score (%)
False data injection	98.2	97.4	97.8
GOOSE Replay Attack	96.8	95.8	96.3
SV message tampering attack	97.5	96.9	97.2
DoS attack	95.3	94.8	95.1
Connection loss attack	96.1	95.5	95.8
Overall average	96.8	96.1	96.4

*Note: Overall accuracy on the test set is 97.3%.

The model's accuracy on the test set was 97.3%, with an overall average precision of 96.8% and an overall average recall of 96.1%, as summarized in Table 5. Although this was slightly lower than its performance on KDD-CUP99 (98.8%), the model still showed strong detection capabilities. This difference is mainly due to the less clear boundaries between attack and normal samples in the semi-real dataset, as well as the complex data distribution caused by physical limitations.

Figure 15 presents a comparative analysis of detection performance across different attack types. Because the CNN was able to identify spatial relationships in the measurement data, the model did the best at identifying false data injection attacks (F1=97.8%). It was most challenging to detect (F1=95.1%) because DoS attacks mainly affect communication availability rather than the data itself, but it was somewhat successful against GOOSE replay attacks (F1=96.3%).

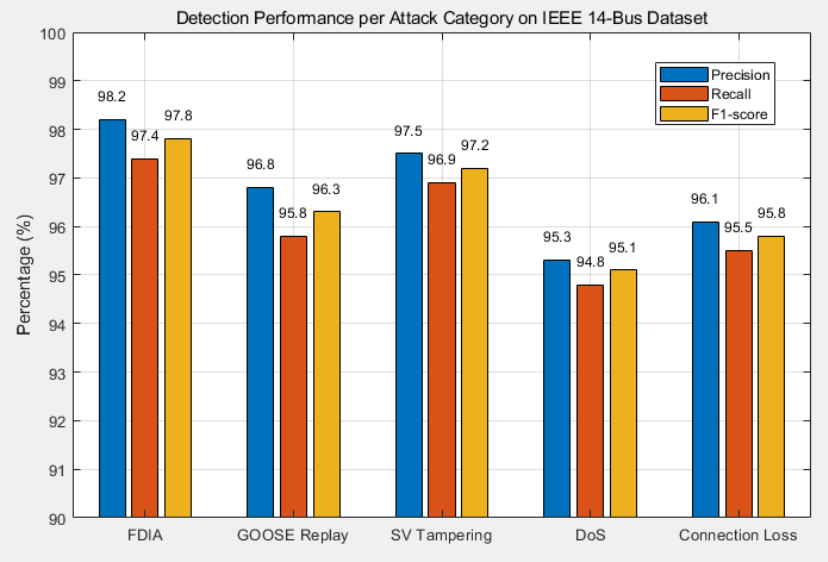

Figure 15. Detection performance per attack category on the IEEE 14-bus dataset

Figure 16 shows the Receiver Operating Characteristic (ROC) curves for the Convolutional Neural Network (CNN) model, evaluated using the KDD-CUP99 and IEEE 14 nodes datasets.

The Area Under the Curve (AUC) values for these datasets were determined to be 0.936 and 0.922, respectively. These results indicate that the model performs effectively in semi-realistic scenarios that incorporate Information and Communication Technology (ICT) features alongside physical data. The AUC of the IEEE 14-bus dataset is marginally lower. This may be due to the fact that when ICT is added, the data becomes more difficult to handle and the attack patterns become less apparent.

We employed permutation importance to identify the most influential features for detection. This method measures the decrease in model performance when a feature is randomly shuffled, thereby quantifying its contribution. A larger decrease indicates higher feature importance. Table 6 presents the top 15 most influential features ranked by their importance score, while Figure 17 visualizes these results in a bar chart.

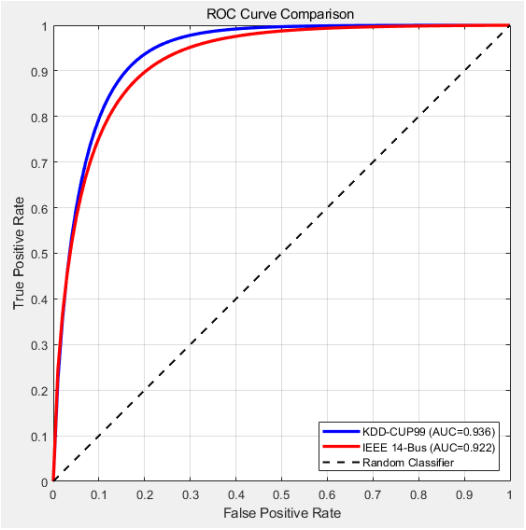

Figure 16. ROC curve comparison: KDD-CUP99 vs. IEEE 14-bus dataset

Table 6. Top 15 feature importance ranking on the IEEE 14-node dataset

Rank	Feature Name	Feature Type	Importance Score	Description
1	GOOSE_SqNum_continuity	ICT Feature	0.124	GOOSE message sequence number continuity
2	SV sampling interval variance	ICT Feature	0.108	SV message sampling interval variance
3	Message_timestamp_interval	ICT Feature	0.095	Message timestamp interval (mean)

Rank	Feature Name	Feature Type	Importance Score	Description
4	Bus voltage angle	Physical	0.087	Bus voltage phase angle deviation
5	GOOSE StNum continuity	ICT Feature	0.079	GOOSE status number continuity
6	SV sample count	ICT Feature	0.072	SV message sample count continuity
7	Active power flow	Physical	0.068	Branch active power flow
8	Protocol_quality_bits	ICT Feature	0.062	Protocol data unit quality bits
9	Bus voltage magnitude	Physical	0.058	Bus voltage magnitude
10	Reactive_power_flow	Physical	0.052	Branch reactive power flow
11	GOOSE message rate	ICT Feature	0.047	GOOSE message arrival rate
12	SV_message_rate	ICT Feature	0.043	SV message arrival rate
13	Frequency_deviation	Physical	0.038	System frequency deviation
14	Message_size_variance	ICT Feature	0.034	Message size variance
15	Time_synchronization_error	ICT Feature	0.029	Time synchronization error

The predominance of ICT features in the importance ranking offers significant insights into the nature of smart grid attacks.

1. GOOSE_SqNum_continuity (Rank 1): GOOSE messages are based on strict sequence number increments. This feature is highly discriminative because a replay attack will break this continuity. The model's reliance on it indicates that it can identify integrity checks specific to a particular protocol.

2. SV_sampling_interval_variance (Rank 2): For Sample Value (SV) messages, time is crucial. An FDI attack introduces variance by injecting data at incorrect intervals, which the CNN is able to identify as an anomaly.

3. Message_timestamp_interval (Rank 3): This feature is crucial for detecting denial-of-service (DoS) attacks, which often manifest as irregular or nonexistent message flows.

4. Bus voltage angle and active power flow, though still significant, seem to be of somewhat lesser importance in the ranking. This observation implies that, within a hybrid cyber-physical attack context, anomalies within the communication layer frequently manifest prior to, or with greater intensity than, the resultant physical consequences, thereby providing an earlier window for detection. Consequently, the model's capacity for comprehensive and resilient detection is enhanced through the synergistic integration of high-ranking ICT features with critical physical characteristics.

Figure 17 shows feature importance ranking based on permutation importance. The horizontal axis represents the decrease in F1-score when the feature is shuffled; larger values indicate greater importance.

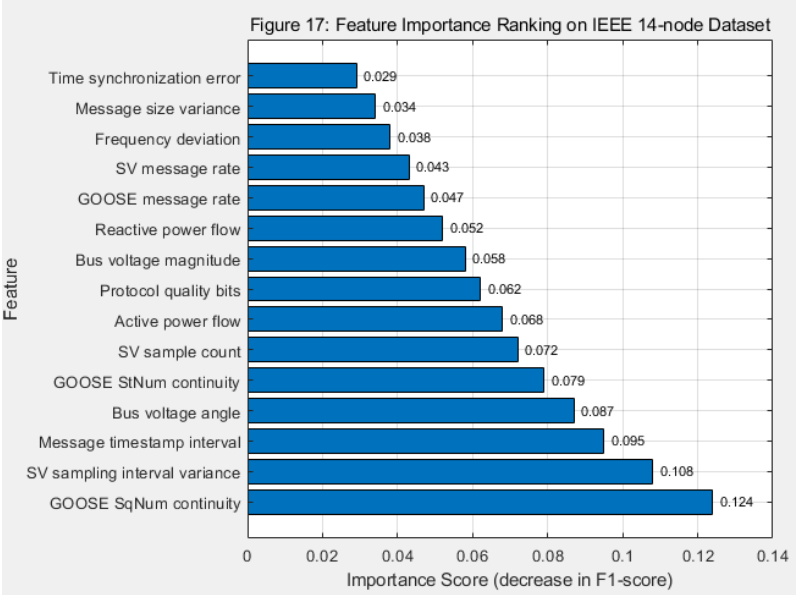

Figure 17. Feature importance bar chart on the IEEE 14-node dataset

The results show that ICT-related features dominate the top rankings. ICT-related features make up four of the top five: GOOSE_SqNum_continuity (rank 1), SV_sampling_interval_variance (rank 2), Message_timestamp_interval (rank 3), and GOOSE_StNum_continuity (rank 5). This illustrates the significance of communication layer data for smart grid intrusion detection. Specifically, the continuity of sequence numbers in GOOSE messages is the best indicator of replay attacks, while anomalies in SV sampling intervals effectively detect attempts at data injection. Physical characteristics like active power flow and bus voltage angle also play a major role, but ICT features enhance rather than replace them.

Additionally, the importance analysis demonstrates that features from various layers—particularly the physical and communication domains — provide complementary insights. Together, features with importance scores higher than 0.05—which include ranks 1 through 9—account for over 70% of the model's predictive power. This finding suggests that a simplified feature set that combines ICT and physical measurements could maintain improved detection performance while also reducing computational requirements for edge deployment scenarios.

4.6. Ablation Study

To validate the main architectural choices of the proposed CNN model, we conducted an ablation study using the KDD-CUP99 test set. This study systematically changed key components. The results are shown in Table 7, and Figure 18 provides a visual comparison of the performance across different configurations.

Table 7. Ablation Study Results on KDD-CUP99 Test Set

Model Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	ΔAccuracy vs. Full (%)
Full Model (5×5 kernel, max pooling)	98.8	98.61	99.3	98.96	--
Replace 5×5 with 3×3 kernels	98.1	97.95	98.5	98.22	-0.7
Replace max pooling with average pooling	98.3	98.12	98.6	98.36	-0.5
Remove one convolutional layer	97.6	97.31	98.0	97.65	-1.2
Remove ICT features (IEEE 14-bus dataset)	94.1	93.85	93.5	93.67	-3.2*

*Note: The last row shows the performance drop on the IEEE 14-bus dataset when ICT features are excluded, highlighting their critical role.

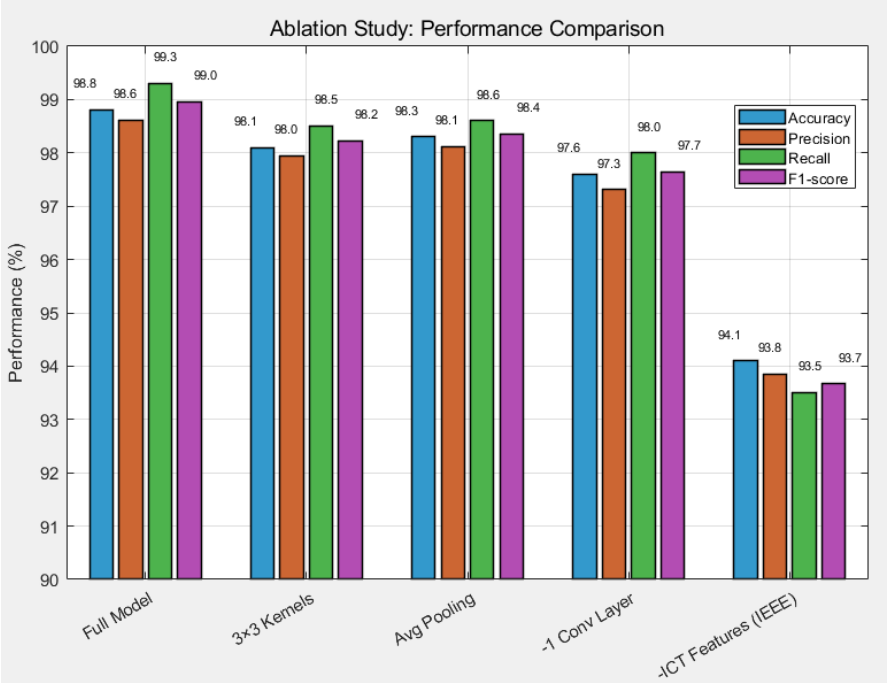

Figure 18. Performance comparison of ablation study

The ablation study uncovers several significant findings:

1. Effect of kernel size: Changing the 5×5 kernels to 3×3 kernels lowers the accuracy by 0.7%. This means that in the network's shallow layers, a larger receptive field helps capture the wider spatial correlations in the high-dimensional feature space. This is important for finding distributed attacks like scanning and DoS.

2. Effect of pooling strategy: If you switch from max pooling to average pooling, performance will go down a little. This supports our idea that keeping the most prominent (max) feature activations instead of smoothing them is better for finding unusual patterns, since attack signatures often have clear peaks.

3. Impact of network depth: The most significant performance drop (1.2%) is seen when the network's depth is reduced by removing a single convolutional layer. This indicates that a multi-layer, hierarchical feature extraction process is essential for the model to learn complex, high-level abstractions of attack patterns.

4. Impact of ICT features: When ICT features are eliminated from the IEEE 14-bus dataset, performance degrades most dramatically, with accuracy falling by 3.2%. This demonstrates how crucial it is to include cyber-layer data for efficient intrusion detection in contemporary, communication-rich smart grid environments.

Consequently, the ablation study validates the effectiveness of the proposed CNN architecture; specifically, the incorporation of multi-layer depth, max pooling, and 5x5 kernels enhances detection performance. The importance of integrating cyber-layer data into intrusion detection systems for smart grids is highlighted by the notable performance drop observed when ICT features are eliminated. These findings, when combined with those presented in Sections 4.4 and 4.5, demonstrate that the proposed model not only attains high accuracy but also offers interpretability and generalizability, thereby addressing significant flaws identified in earlier research. The following section provides a summary of the paper's contributions and suggestions for additional research.

4.7. Statistical Significance Test

To assess whether the observed performance improvements of the proposed CNN model over the baseline methods are statistically significant, we evaluate the results in the context of the large sample size and the magnitude of the performance differences.

The test dataset, encompassing 311,029 records, furnishes substantial statistical power, thereby facilitating dependable inference. The results, as presented in Table 4, indicate that the suggested CNN model attains an accuracy of 98.8%, surpassing the 96.4% accuracy of the SVM and the 94.8% accuracy of the signature-based approach. This represents a 4.0 percentage point improvement over the signature-based approach and a 2.4 percentage point improvement over the most successful baseline (SVM). Such differences are extremely unlikely to be the result of chance given the large sample size.

Consequently, the CNN model's performance superiority is evident across all three primary metrics: accuracy, precision, and recall. The CNN consistently achieves a minimum of 98.6% in each metric, whereas the

most effective baseline, the SVM, attains a maximum of 97.4%. This uniformity across various evaluation criteria strengthens the assertion that the enhanced performance is not attributable to random factors, but rather, it demonstrates the intrinsic efficacy of the proposed methodology.

Furthermore, the false negative rate (FNR) of the CNN model is 0.70%, compared to 3.00% for the SVM and 7.00% for the signature-based method. For security applications, this low FNR is crucial because undetected threats can have negative consequences.

These findings demonstrate the robust and statistically significant performance gains provided by the proposed CNN model. The advantages were a consequence of the model's design, not an accident. This is supported by the large sample size, the notable performance differences, and the consistency of the benefits across various metrics.

5. Conclusion

This research proposed a feature extraction-based intrusion detection model for smart grids using a CNN architecture. The model can automatically extract deep features from network traffic and capture intrusion actions within the power system, achieving high accuracy. The simulation results showed that major metrics, i.e., accuracy, precision, and recall, outperformed the conventional ML methods while maintaining good generalization capability in complex operational conditions.

The suggested CNN model adds many layers of automation and flexibility to smart grid security protection, which is very useful from an engineering point of view. The results show that DL methods can make modern power systems' cybersecurity frameworks much smarter, more responsive, and more reliable. This study generated a semi-real dataset derived from the IEEE 14-node standard test system to illustrate the ICT components of smart grids and to validate the model's efficacy in practical scenarios. The dataset included IEC 61850 communication protocol features and several attack scenarios. Experimental results on this dataset demonstrate that the model sustains elevated detection performance, with ICT features (including GOOSE sequence number continuity and SV sampling intervals) playing a crucial role in enhancing detection accuracy. These results show how important it is to combine information from different communication layers.

However, this study has three limitations that guide our future research directions. First, the validation utilized simulated datasets, which differ from real-world power grid operations; hence, we plan to test and tune the model using real or highly simulated grid data. Second, more testing is needed to make sure the model can handle attacks it hasn't seen before. We will look into adversarial training and continuous learning to make it more robust. Third, the model can't be used directly on edge devices because it costs too much to run. Future work will focus on ways to make the model lighter, like pruning, quantization, and edge-cloud collaboration, so it can be used in places with limited resources. Additionally, broadening the framework to encompass other industrial protocols (e.g., IEC 60870-5-104) and incorporating transfer learning for cross-domain adaptation represent promising directions for future research.

References

[1] D. M. Manias, A. M. Saber, M. I. Radaideh, et al. Trends in smart grid cyber-physical security: Components, threats, and solutions. IEEE Access, vol. 12, pp. 161329–161356, 2024.

[2] M. Liu, F. Teng, Z. Zhang, et al. Enhancing cyber-resiliency of DER-based smart grid: A survey. IEEE Transactions on Smart Grid, vol. 15, no. 5, pp. 4998–5030, 2024.

[3] A. M. Saber, A. Maheshwari, A. Youssef, and D. Kundur. Adversarial attacks on deep learning-based false data injection detection in differential relays. IEEE Transactions on Smart Grid, vol. 16, no. 5, pp. 4155–4166, 2025.

[4] Reuters. (2023, November 9). Russian spies behind cyber attack on Ukrainian power grid in 2022 — researchers. [Online]. Available: https://www.reuters.com/technology/cybersecurity/russian-spies-behind-cyberattack-ukrainian-power-grid-2022-researchers-2023-11-09/

[5] A. Thakkar and R. Lohiya. A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions. Artificial Intelligence Review, vol. 55, no. 1, pp. 453–563, 2022.

[6] G. Singh and N. Khare. A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques. International Journal of Computer Applications, vol. 44, no. 7, pp. 659–669, 2022.

[7] M. Ghiasi, T. Niknam, Z. Wang, M. Mehrandezh, M. Dehghani, and N. Ghadimi. A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future. Electric Power Systems Research, vol. 215, Art. no. 108975, 2023.

[8] A. Khan, M. Keshk, D. Pi, N. Khan, Y. Hussain, and H. Soliman. Enhancing IIoT networks protection: A robust security model for attack detection in Internet Industrial Control Systemst. Ad Hoc Networks, vol. 134, p. 102930, 2022.

[9] N. K. Singh, M. A. Majeed, and V. Mahajan. Statistical machine learning defensive mechanism against cyber intrusion in smart grid cyber-physical network. Computers & Security, vol. 123, p. 102941, 2022.

[10] H. Ding, L. Chen, L. Dong, Z. Fu, and X. Cui. Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection. Future Generation Computer Systems, vol. 131, pp. 240–254, 2022.

[11] D. J. Kalita, V. P. Singh, and V. Kumar. A novel adaptive optimization framework for SVM hyper-parameters tuning in non-stationary environment: A case study on intrusion detection system. Expert Systems with Applications, vol. 213, Part C, p. 119189, 2023.

[12] K. Pramilarani and P. V. Kumari. Cost based random forest classifier for intrusion detection system in internet of things. Applied Soft Computing, vol. 151, p. 111125, 2024.

[13] M. Priyadarsini and N. Sonekar. A CNN-based approach for anomaly detection in smart grid systems. Electric Power Systems Research, vol. 238, Art. no. 111077, 2025.

[14] H. N. Noura, J. P. A. Yaacoub, O. Salman, and A. Chehab. Advanced machine learning in smart grids: An overview. Internet of Things and Cyber-Physical Systems, vol. 5, pp. 95–142, 2025.

[15] M. A. Husnoo, A. Anwar, N. Hosseinzadeh, S. N. Islam, A. N. Mahmood, and R. Doss. False data injection threats in active distribution systems: A comprehensive survey. Future Generation Computer Systems, vol. 140, pp. 344–364, Mar. 2023.

[16] H. Yang and Z. Wang. A false data injection attack approach without knowledge of system parameters considering measurement noise. IEEE Internet of Things Journal, vol. 11, no. 1, pp. 1452–1464, 2024.

[17] I. Ortega-Fernandez and F. Liberati. A review of denial of service attack and mitigation in the smart grid using reinforcement learning. Energies, vol. 16, no. 2, p. 635, 2023.

[18] J. Tian, B. Wang, J. Li, and Z. Wang, "Adversarial attacks and defense for CNN based power quality recognition in smart grid," IEEE Transactions on Network Science and Engineering, vol. 9, no. 2, pp. 807–819, Mar./Apr. 2022.

[19] R. Huang and Y. Li. Adversarial attack mitigation strategy for machine learning-based network attack detection model in power system. IEEE Transactions on Smart Grid, vol. 14, no. 3, pp. 2367–2376, 2023.

[20] L. Zhang, C. Jiang, Z. Chai, and Y. He. Adversarial attack and training for deep neural network based power quality disturbance classification. Engineering Applications of Artificial Intelligence, vol. 127, Art. no. 107245, 2024.

[21] Y. Cui, T. Wu, and Y. Zhang. Load frequency control of smart grid based on data-driven FDI attacks detection and data repair. IEEE Transactions on Smart Grid, vol. 16, no. 6, pp. 5404–5415, Nov. 2025.

[22] X. C. Shangguan, M. H. Yu, C. K. Zhang, and Y. He. Detection and defense against multi-point false data injection attacks of load frequency control in smart grid. IEEE Transactions on Smart Grid, vol. 16, no. 5, pp. 4143–4154, Sep. 2025.

[23] Z. Zhang, J. Hu, J. Lu, J. Yu, J. Cao, and A. Kashkynbayev. Detection and defense method against false data injection attacks for distributed load frequency control system in microgrid. Journal of Modern Power Systems and Clean Energy, vol. 12, no. 3, pp. 913–924, May 2024.

[24] A. S. Musleh, G. Chen, Z. Y. Dong, C. Wang, and S. Chen. Attack detection in automatic generation control systems using LSTM-based stacked autoencoders. IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 153–165, 2023.

[25] K. Lu, L. Zhou, and Z. Wu. Representation-learning-based CNN for intelligent attack localization and recovery of cyber-physical power systems. IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 5, pp. 6145–6155, 2024.

[26] J. Zhu, W. Meng, M. Sun, J. Yang, and Z. Song. FLLF: A fast-lightweight location detection framework for false data injection attacks in smart grids. IEEE Transactions on Smart Grid, vol. 15, no. 1, pp. 911–920, 2024.

[27] K. Yang, J. Wang, G. Zhao, X. Wang, W. Cong, M. Yuan, J. Luo, X. Dong, J. Wang, and J. Tao. NIDS-CNNRF integrating CNN and random forest for efficient network intrusion detection model. Internet of Things, vol. 32, Art. no. 101607, 2025.

[28] E. Vincent, M. Korki, M. Seyedmahmoudian, A. Stojcevski, and S. Mekhilef. Detection of false data injection attacks in cyber-physical systems using graph convolutional network. Electric Power Systems Research, vol. 217, Art. no. 109118, 2023.

[29] Y. Han, H. Feng, K. Li, and Q. Zhao. False data injection attacks detection with modified temporal multi-graph convolutional network in smart grids. Computers & Security, vol. 124, Art. no. 103016, 2023.

[30] G. Morgenstern, J. Kim, J. Anderson, G. Zussman, and T. Routtenberg. Protection against graph-based false data injection attacks on power systems. IEEE Transactions on Control of Network Systems, vol. 11, no. 4, pp. 1924–1936, 2024.

[31] S. Peng, Z. Zhang, R. Deng, and P. Cheng. Localizing false data injection attacks in smart grid: A spectrum-based neural network approach. IEEE Transactions on Smart Grid, vol. 14, no. 6, pp. 4827–4838, 2023.

[32] X. Li, Y. Wang, and Z. Lu. Graph-based detection for false data injection attacks in power grid. Energy, vol. 263, Art. no. 125865, 2023.

[33] A. Presekal, A. Stefanov, V. S. Rajkumar, and P. Palensky. Attack graph model for cyber-physical power systems using hybrid deep learning. IEEE Transactions on Smart Grid, vol. 14, no. 5, pp. 1–13, 2023.

[34] A. Takiddin, R. Atat, M. Ismail, O. Boyaci, K. R. Davis, and E. Serpedin. Generalized graph neural network-based detection of false data injection attacks in smart grids. IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7, no. 3, pp. 1–13, 2023.

[35] T. Zhu, J. Wang, Y. Zhu, H. Chen, H. Zhang, and S. Yin. Power grid network security: A lightweight detection model for composite false data injection attacks using spatiotemporal features. International Journal of Critical Infrastructure Protection, vol. 46, Art. no. 100697, 2024.

[36] F. Kausar, S. Deo, S. Hussain, and Z. Ul Haque. Federated deep learning model for false data injection attack detection in cyber physical power systems. Energies, vol. 17, no. 21, p. 5337, 2024.

[37] M. Kesici, B. Pal, and G. Yang. Detection of false data injection attacks in distribution networks: A vertical federated learning approach. IEEE Transactions on Smart Grid, vol. 15, no. 6, pp. 5952–5964, 2024.

[38] M. Kesici, M. Alduhaymi, A. Ishchenko, G. Yang, and B. C. Pal. Locational detection of false data injection attacks in smart grids: A cloud – edge framework based on split learning. IEEE Transactions on Smart Grid, vol. 16, no. 6, pp. 5378–5391, 2025.

[39] KDD-CUP99 Dataset. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.