Multimedia Tools and Applications (2023) 82:19463–19486
https://doi.org/10.1007/s11042-022-14121-2

A CNN-transformer hybrid approach for an intrusion detection system in advanced metering infrastructure

Ruizhe Yao[1] · Ning Wang[1] · Peng Chen[1] · Di Ma[1] · Xianjun Sheng[1]

Received: 13 July 2021 / Revised: 18 October 2022 / Accepted: 25 October 2022 /
Published online: 1 December 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Bi-directional communication networks are the foundation of advanced metering infrastructure (AMI), but they also expose smart grids to serious intrusion risks. While previous studies have proposed various intrusion detection systems (IDS) for AMI, most have not comprehensively considered the impact of different factors on intrusions. To ensure the security of the bi-directional communication network of AMI, this paper proposes an IDS based on deep learning theory. First, the invalid features are eliminated according to the feature screening strategy based on eXtreme Gradient Boosting (XGBoost), after which the data distribution is balanced by the adaptive synthetic (ADASYN) sampling technique. Next, multi-space feature subsets based on the convolutional neural network (CNN) are constructed to enrich the spatial distribution of samples. Finally, the Transformer is used to construct feature associations and extract crucial traits, such as the temporal and fine-grained characteristics of features, to complete the identification of intrusion behaviors. The proposed IDS is tested on the KDDCup99, NSL-KDD, and CICIDS-2017 datasets, and the results show that it has high performance with accuracy of 97.85%, 91.04%, and 91.06% respectively.

Keywords Smart grids · Advanced metering infrastructure · Intrusion detection systems · Convolutional neural network · Transformer

✉ Ning Wang
ningwang@dlut.edu.cn

Ruizhe Yao
yrzhe@mail.dlut.edu.cn

Peng Chen
PengC@mail.dlut.edu.cn

Di Ma
madi1314@mail.dlut.edu.cn

Xianjun Sheng
sxjun@dlut.edu.cn

[1] Electronic Information and Electrical Engineering, Dalian University of Technology,
No.2 Linggong Road, Ganjingzi District, Dalian, 116024, Liaoning, China

1 Introduction

Among the key components of smart grids, advanced metering infrastructure (AMI) is an important node that connects customers with the power system [5, 7, 14, 50]. It links key equipment, such as smart household multimedia tools, smart meters, data concentrators, and measurement data management centers of the smart distribution grid, with a bi-directional communication network to collect electricity consumption information and provide operational references for real-time tariff strategies and power dispatch. However, bi-directional communication networks also make AMI more vulnerable to intrusions and attacks; intrusions can leak private user information, while attacks can seriously affect the stable and secure operation of the grid [34, 48].

In previous research, passive defense technologies, including encryption [24, 45], authentication [10, 18], and privacy protection [6, 20], have been the main methods used to ensure the security of AMI. However, as it is difficult to dynamically detect intrusion behavior using these techniques, an active defense based on intrusion detection systems (IDSs) is required to accurately assess the risks faced by AMI to reduce and avoid such scenarios. IDSs can be categorized as either misuse or anomaly detection systems according to their detection means. Misuse detection systems mainly identify attacks by building a knowledge base and carrying out pattern matching, but their ability to recognize attacks is limited. Anomaly detection systems identify attacks by comparing differences between actual and normal behavior. As anomaly detection can detect unknown attacks and is more universal, it is more suitable for the complex communication environment of AMI.

With the application of artificial intelligence (AI) technology, anomaly detection methods based on machine learning (ML) have become a popular research topic in the field of IDSs. As a branch of AI, traditional ML has been pioneered in AMI intrusion detection due to its low data requirements, high interpretability, and fast training [23, 29, 38]. However, traditional ML is characterized by low detection accuracy, is prone to fall into the local optimum, and struggles to fit high-dimensional attack scenarios. Therefore, it is difficult to adapt ML to the complex and diverse communication environment of AMI. Deep learning (DL) is an important branch of ML that can fit various complex situations via a deep neural network without feature engineering, and has strong representation ability. In addition, it perfectly overcomes the shortcomings of traditional ML, such as its low accuracy, weak fitting ability, and poor multi-classification effect, and is gradually being employed in AMI intrusion detection research [1, 3, 4, 11, 28].

Nevertheless, most of the current research on AMI intrusion detection based on DL focuses on local and temporal features [26], rather than deeper characteristics, such as correlations and fine-grained features. In addition, datasets are the basis of DL, and the low frequency of intrusion means that most datasets used in IDSs are unbalanced [35] and may contain invalid features that are not conducive to classification [40]. Thus, a more comprehensive and efficient IDS should be designed within the context of AMI.

Considering the problems with datasets and the correlation and fine-grained aspects of AMI communication characteristics, this paper proposes a DL model that incorporates a convolutional neural network (CNN) and Transformer to complete the intrusion detection of AMI. In the proposed model, the impacts of unbalanced data distributions and invalid features are first reduced by adaptive synthetic (ADASYN)-based data augmentation and eXtreme gradient boosting (XGBoost)-based feature screening. The CNN component is then used to map the original data to different subspaces and build multi-space feature subsets to obtain a richer nonlinear representation. Subsequently, the Transformer component is used to mine deeper characteristics, such as fine-grained features, and to complete the construction of feature associations. Finally, the mapping relationships between features and labels are constructed by the softmax function. The proposed model combines the advantages of the CNN and Transformer and has excellent intrusion detection performance. The main contributions of this research are listed as follows.

(1) A CNN-Transformer hybrid network for an AMI IDS is proposed. Compared with models proposed in related work, the proposed model achieves better performance by adopting a Transformer to extract temporal and fine-grained features and construct correlations between arbitrary features in the AMI network. To the best of the authors' knowledge, the proposed model is the first application of the Transformer in the IDS field.

(2) To reduce the impacts caused by invalid features, unbalanced sample distributions, and single feature expression, XGBoost and ADASYN sampling are employed to process the dataset, and a CNN is used to construct multi-space feature subsets of the original data to increase the diversity of samples.

(3) The proposed model is extensively evaluated on the KDDCup99, NSL-KDD, and CICIDS2017 datasets. These three datasets have similar characteristics and attack types as AMI, and the samples are well distributed and abundant. The experimental results demonstrate that the proposed model achieved better performance than models proposed in related work.

The remainder of this paper is arranged as follows. Section 2 provides an overview of the related work. Section 3 presents the components of the CNN-Transformer hybrid network. Section 4 describes feature screening, dataset balancing, and data preprocessing. Finally, the results of experiments are analyzed in Section 5, and conclusions are provided in Section 6.

2 Related work

Since the initial development by Anderson [8], IDSs have been widely used in power systems. Traditional intrusion detection in AMI is achieved via methods such as pattern matching. However, with the application of AI technology, traditional ML and DL algorithms provide new solutions for IDSs.

2.1 IDS in AMI based on traditional ML

Traditional ML methods have a mostly shallow structure, and are widely used in IDSs due to their simplicity, fast training, high interpretability, and generalization ability.

An IDS based on a deep belief network (DBN) was proposed by He et al. [22], and determines the mapping between the input and label via a probability generation model; while this method has strong scalability, it is prone to fall into overfitting. In view of the slow detection speed of traditional methods, an IDS based on an extreme learning machine (ELM) was proposed by Shen et al. [42]; while the detection efficiency of this method is greatly improved, it faces issues with choosing the optimal parameters. To solve this problem, Zhang et al. [54] introduced a genetic algorithm (GA) to obtain the optimal parameters, and the results of experiments showed that the resulting GA-ELM has powerful intrusion detection capabilities. Tian et al. [44] designed an IDS for AMI based on k-means clustering; this model is driven by data and behavioral characteristics to complete the rapid clustering of abnormal electricity users, but its detection precision remains very limited. An IDS model with a two-layer structure was presented by Punmiya et al. [37]. The model first uses a gradient boosting theft detector (GBTD) to complete feature screening, after which the processed data are classified using gradient boosting classifiers (GBCs); this combination has a strong feature processing ability. In view of the low precision and over-fitting of the methods noted previously, Yan et al. [49] proposed an electricity theft detection model based on XGBoost. This model is not only characterized by improved generalization ability, but could also be used in the case of an unbalanced sample distribution. Salman et al. [41] designed an IDS based on a boosted C5.0 decision tree (DT); the proposed model includes the introduction of an adaptive synthetic algorithm, which improves the efficiency of the IDS to ultimately improve the traditional DT. Engelbrecht et al. [17] introduced support vector machines (SVMs) to the study of AMI intrusion detection; in the proposed method, an SVM is used to locate the optimal hyperplane between normal and abnormal users and complete binary classification. To address the shortcomings of the use of a single ML algorithm, Kong et al. [32] proposed a hybrid IDS that incorporates multiple ML algorithms; the IDS integrates the advantages of the *K*-nearest neighbors (KNN), DT, and SVM methods, and is characterized by a stronger detection capability.

In summary, it is clear that traditional DL relies too heavily on feature engineering, making it difficult for it to complete multi-classification. Therefore, it is challenging to apply traditional DL models to AMI intrusion detection on a large scale.

2.2 IDS in AMI based on DL

DL models are mostly deep structures that extract features via feature extraction components, after which they construct the mapping relationship between features and labels via a deep neural network. DL is characterized by high accuracy, multi-classification ability, and no requirements for manual feature design, meaning its performance is better than that of traditional ML in most IDS scenarios.

Yin et al. [53] exploited the sensitivity of recurrent neural networks (RNNs) to sequence data and applied an RNN to the field of IDS with good results; however, this method still suffers from long-term dependence. To address this issue, an IDS based on long and short-term memory (LSTM) was proposed by Kim et al. [31]; the resulting LSTM-IDS model overcomes the problem of gradient explosion while inheriting the advantages of the RNN-IDS model. Gupta et al. [19] designed an IDS based on LSTM and an improved one-vs-one model. In the proposed two-layer architecture, LSTM is used to classify normal and abnormal situations, and the improved one-vs-one model is used to complete multi-classification. The results of experiments on different datasets showed that this scheme had higher accuracy than traditional methods. Liu et al. [33] proposed an IDS based on a CNN that has a strong ability to extract local features; while it was successfully applied to intrusion detection, a very limited generalization capability and risk of falling into overfitting were found to remain. In response to this problem, Yang et al. [51] presented an improved CNN-IDS model by extracting features through a parallel 1D-CNN architecture, which was found to achieve better performance in a shorter amount of time. To address the problem of the limited feature extraction capabilities of a single DL model, a wide and deep CNN was proposed by Zheng et al. [56] for the detection of AMI intrusions. In this model, the wide component consists of a deep neural network (DNN) for the detection of the periodic features of the data and a deep component consisting of a CNN for the extraction of the local features of the data; together, these two features guide intrusion classification. Hasan et al. [21] proposed a CNN-LSTM-based IDS for electricity theft. The model was found to achieve improved detection ability and a stronger periodic representation of features via the serial combination of the CNN and LSTM. Javaid et al. [27] solved the problem of unbalanced sample distributions faced by the method proposed by [21] via the introduction of

ADASYN to process few-shot samples, and achieved good results. However, this type of serial combination approach is susceptible to feature loss. To address the problems faced by previous methods [21, 27], a cross-layer aggregated CNN-LSTM model was presented by Yao et al. [52]. This model fuses features extracted by CNN and LSTM into a comprehensive feature that contains multi-domain characteristics, which solves the deficiency of the serial CNN-LSTM model; however, the structure of LSTM is complicated and less efficient. To deal with this problem, Ayub et al. [9] introduced gated recurrent units (GRUs) to replace the LSTM component, and proposed a CNN-GRU-based IDS for use in AMI. Compared with LSTM, GRUs have a simpler structure and faster convergence, and require less data. While all these methods extract features directly from the original data, the invalid features are not conducive to better performance implementation. To address this issue, Cosimo et al. [25] proposed an auto-encoder (AE)-based IDS in which the features extracted by the AE replace the raw data as the input to the IDS, and the results of experiments demonstrated that this approach effectively reduces false alarms. Shone et al. [43] combined the advantages of ML and DL and proposed an NDAE-RF-based IDS, which retains the advantages of ML and reduces the training time.

In addition, the DL methods noted previously are also characterized by disadvantages including slow training, weak correlations between features, and poor long-term memory capability. To solve these problems, Vaswani et al. [46] proposed the Transformer structure; this structure uses a parallel approach to process the data, which achieves improved real-time performance. It also allocates limited resources to important areas through a self-attention mechanism, which reinforces the position association of different features. Moreover, its residual architecture alleviates the gradient disappearance problem. The effectiveness of the Transformer has been demonstrated in a variety of tasks in fields including graph matching [16], natural language processing [15], and behavior prediction [47].

Considering the advantages of the transformer, this paper proposes a CNN-Transformer hybrid approach for an IDS in AMI. The approach uses a CNN to construct multi-space feature subsets of the original data, after which the Transformer is used to extract temporal and fine-grained features and learn the correlations between different features to achieve better detection performance.

3 System components

In AMI communication networks, power consumption data have obvious temporal and fine-grained characteristics under normal conditions. Some features are also strongly correlated with each other [56], and those properties will be corrupted when an intrusion occurs. As such, this paper proposes a CNN-Transformer hybrid network-based IDS. The proposed IDS is presented in Fig. 1, and mainly consists of a feature extraction component-based CNN and an intrusion detection component-based Transformer. The CNN is primarily composed of a convolution layer, pooling layer, and a fully connected (FC) layer, and is mainly used to extract local features and build multi-space feature subsets. The Transformer is primarily composed of input and position embedding, multi-head self-attention, a feed-forward neural network, and a residual network. The Transformer is predominantly used for the extraction of temporal and fine-grained features, and as it can establish the associations between different features, it is able to detect small changes in AMI communication networks. Finally, the extracted features are input into the FC layer and the softmax function to complete intrusion classification.

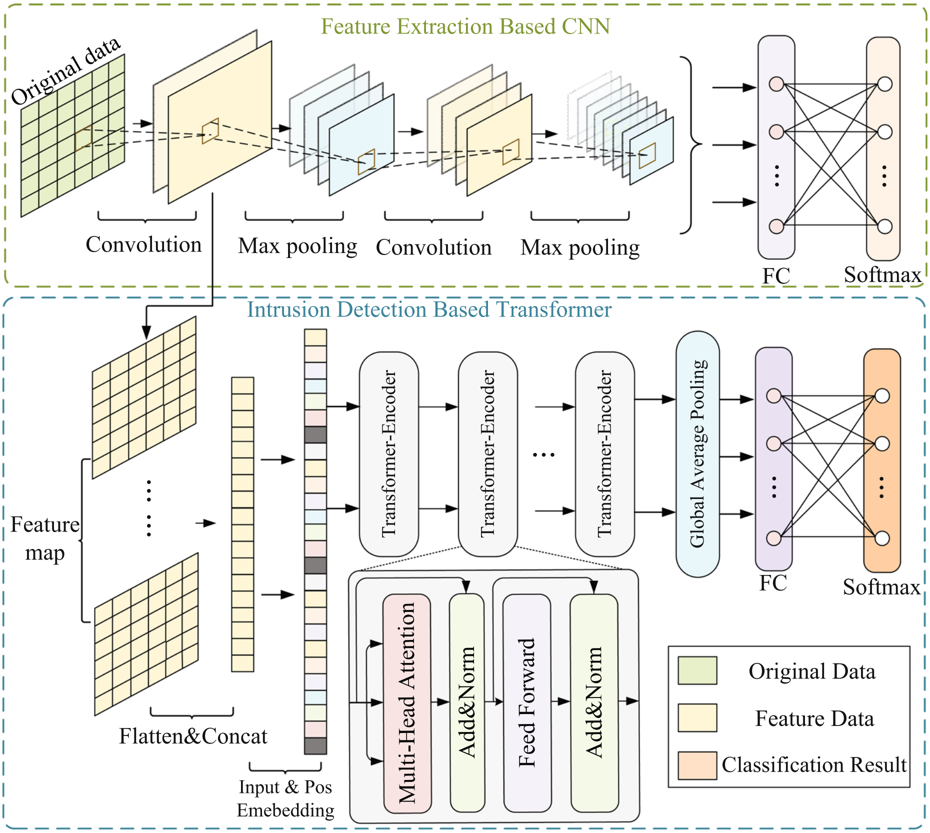


Fig. 1 The architecture of CNN-Transformer hybrid network

3.1 Feature extraction based CNN

The basic CNN is composed of three parts, namely a convolution layer, activation function, and pooling layer. When processing the classification task, an FC layer is also introduced to complete the mapping of the input to the label. The structure of the CNN used in this study mainly consists of two convolutional layers, two pooling layers, and an FC layer.

The convolutional layer is the core of the CNN. It can extract local features via a convolution operation and consists of several convolutional kernels stacked together. Via the convolution operation, the convolution kernels can learn different features in the input data and retain the spatial correspondence of different features. The most important features are then retained by maximum pooling. Compared with other pooling tools, maximum pooling can effectively alleviate the offset of the estimated value caused by the parameter error of the convolution layer. Finally, the features are input to the FC layer with the softmax function to complete the mapping of features to labels. The expression of the CNN is defined as follows.

$$\begin{cases} X_i = f(w_i \otimes X_{i-1} + b_i) \\ Q_j = Max(P_j^0, P_j^1, P_j^2 ... P_j^t) \\ Y_j = f(w_j \otimes y_{j-1} + b_j) \end{cases} \tag{1}$$

where w_i and b_i represent the weight and bias of the i-th convolutional kernel, $\otimes$ stands for the convolution operation, $f(x)$ refers to the activation function, Q_j represents the pooling result of the j-th region, Max stands for the maximum pooling operation, and P_j^t expresses the t-th element of the j-th pooling region, w_j and b_j represent the weights and biases of j-th neurons in the FC layer, X_i, X_{i-1} and y_{j-1}, Y_j indicate the input and output of the convolution kernel and FC respectively.

To enrich the feature space and ensure diversity, the convolutional kernel of the first convolutional layer is selected as the feature extractor, and the original data are re-entered after completing the pre-training of the CNN. The feature map is then flattened and concatenated to obtain a multi-space representation of the original data. This operation allows the original data to be mapped in different spaces, which increases the diversity of samples while avoiding the feature loss problem caused by maximum pooling. Feature extraction-based CNN is defined by (2).

$$x_i = H_{concat}(G_{flatten}(X_1, X_2, ...X_k)) \tag{2}$$

Where X is the feature map extracted by the k-th convolution kernel, k is the number of convolution kernel, $G_{flatten}(x)$ represents the flatten process, $H_{concat}(x)$ means the concatenate operation, and x_i refers to the extracted multi-space feature.

3.2 Transformer-based feature identification

The Transformer mainly consists of stacked encoders and decoders. As encoders predominantly complete the feature extraction function, only the encoder components are employed in the proposed model. The main components of the Transformer encoder are input and position embedding, multi-head self-attention, layer normalization, the forward neural network, and the residual network.

The embedding component consists of both input and position embedding. Input embedding reflects the relationship of discrete inputs mapped to the same space, position embedding is used to explain the sequential relationship of different features, and the sum of the two forms is the final input of the Transformer encoder, as expressed by the following equation.

$$\begin{cases} X^i_{input-emb} = W \times G(X^i) \\ X^i_{pos-emb} = sin(pos/1000^{k/d_{model}}) \quad (k = 2i) \\ X^i_{pos-emb} = cos(pos/1000^{k/d_{model}}) \; (k = 2i + 1) \\ X^i_{input} = X^i_{input-emb} + X^i_{pos-emb} \end{cases} \tag{3}$$

where X_i represent i-th feature of the input, W represents the weight matrix, $G(x)$ stands for the one-hot function, pos denotes the position of the feature, d_{model} stands for the input dimension, k represents the position of the input, $X^i_{input-emb}$ and $X^i_{pos-emb}$ means the input and position embedding result.

The multihead self-attention mechanism, which is an extension of the attention mechanism, is the basis of Transformer. In this paper, we adopt the multi-headed self-attention mechanism with a scaled dot product.

Compared with the scaled dot-product attention mechanism, the multi-head scaled dot-product attention mechanism allows the model to focus on the information of different features mapped to various subspaces to obtain diverse attention values. The attention values of each head are calculated independently, which effectively prevents overfitting, and its specific expression is as follows:

$$\begin{cases} Attention(Q,K,V) = softmax((Q \cdot K^T)/\sqrt{d_k})V \\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \\ Multihead(Q,K,V) = concat(head_1, head_2 ... head_h)W^O \end{cases} \tag{4}$$

Where Q, K and V represents the query matrix, key matrix and value matrix respectively, $\cdot$ represents the dot product operation, d_k represents the dimension of k, h is the number of attention heads, and d_v and d_{model} represent the dimension of v and $model$. In addition,$W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_k}$, $W^O \in \mathbb{R}^{d_{model} \times hd_v}$.

The unique structure of the Transformer solves the limitation of traditional RNN that cannot be computed in parallel. Compared with CNN, the spatial distance between different features in Transformer is equal, so it requires fewer operations to compute the association between two features. Other structures like multihead self-attention mechanism and residual network also allow the model to achieve deep architecture while focusing on the temporal and fine-grained features of different subspaces.

3.3 Training process of proposed CNN-transformer hybrid network

Considering the various advantages of the CNN and Transformer, this paper proposes a CNN-Transformer hybrid approach for the detection of intrusions in AMI. ADASYN and XGBoost are also introduced to reduce the interference of an unbalanced data distribution

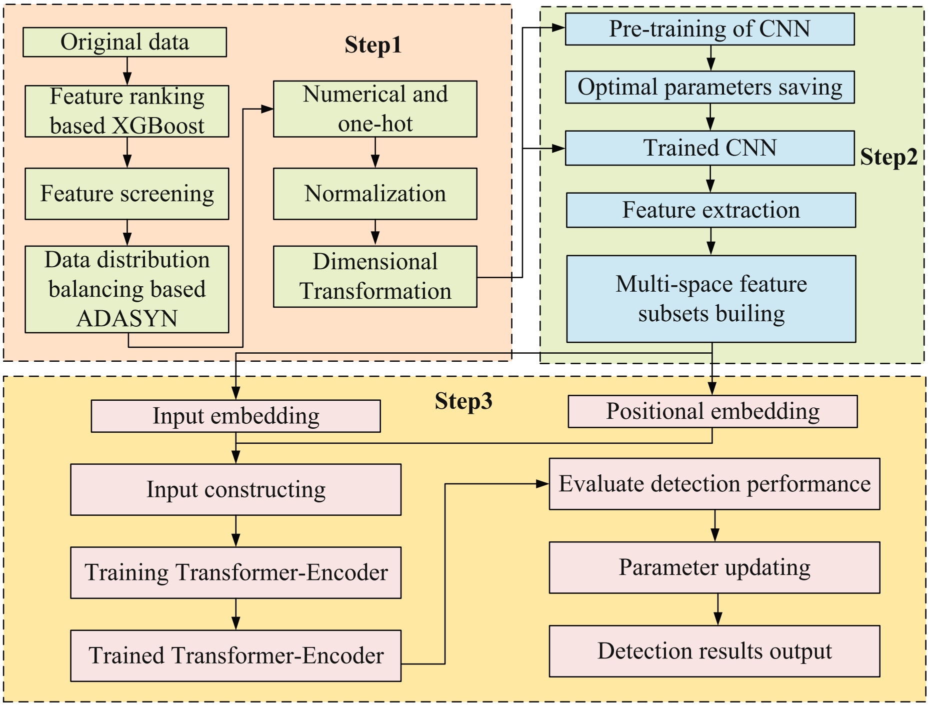

Step1
Original data
Feature ranking based XGBoost
Feature screening
Data distribution balancing based ADASYN
Numerical and one-hot
Normalization
Dimensional Transformation
Step2
Pre-training of CNN
Optimal parameters saving
Trained CNN
Feature extraction
Multi-space feature subsets builing
Step3
Input embedding
Positional embedding
Input constructing
Training Transformer-Encoder
Trained Transformer-Encoder
Evaluate detection performance
Parameter updating
Detection results output

Fig. 2 IDS based on CNN-Transformer hybrid approach

and invalid features. As shown in Fig. 2, the proposed CNN-Transformer hybrid approach mainly consists of the following three steps.

Step 1: Processing of datasets and features. First, XGBoost is selected as the feature screening tool to remove invalid features, and ADASYN is used to balance the dataset distribution. Numerical and one-hot encoding are then adopted to facilitate model processing. Finally, normalization is carried out to eliminate the scale effect.

Step 2: The CNN is pre-trained with the processed dataset, which is then re-input into the CNN, and the convolution kernel is used to complete the mapping of the original data in multiple feature subspaces. The results of each feature map are then extracted and constructed into multi-space feature subsets.

Step 3: Feature extraction and classification-based Transformer. The extracted multi-space feature subsets are processed by input and position embedding to form the final input of the Transformer. The unique structure of the Transformer is used to extract features, and the mapping relationship between features and labels is constructed by the FC layers and softmax function.

4 Dataset selection and data preprocessing

4.1 Datasets selection

The KDDCup99, NSL-KDD, and CICIDS-2017 datasets are selected as the experimental benchmark datasets, and are described as follows.

The KDDCup99 dataset is derived from connectivity and system audit data collected by the Lincoln Laboratory, and simulates the U.S. Air Force LAN system. It contains five major categories and 40 minor categories, with a total of 4,898,431 pieces of data. Each piece of data contains 32 continuous feature attributes and nine discrete feature attributes [13]. The KDDCup99 dataset has a rich sample size, and the attack types are similar to those experienced by AMI [30]. As the original KDDCup99 dataset is large, 10% of the original dataset was selected as the training set. After excluding the data that do not appear in the training set, the sample distribution of the KDDCup99 dataset is shown in Table 1.

The attack and feature types of the NSL-KDD dataset are similar to those of the KDDCup99 dataset. However, the redundant data are eliminated in the NSL-KDD dataset to make the distribution more balanced [12], and the duplicate data in the test set are removed to enable more accurate detection results. In addition, the dataset has a reasonable number of records in the training and test sets, thereby making the results of different studies

Table 1 Distribution of KDDCup 99

Algorithms	Training data		Testing data	
	amount	ratio(%)	amount	ratio(%)
Normal	97277	16.96	60581	20.73
Dos	391458	79.24	223298	76.40
Probe	4107	0.83	2377	0.81
R2L	1126	0.23	5993	2.05
U2R	52	0.01	38	0.01
Total	494020	100	292287	100

Table 2 Distribution of NSL-KDD

Algorithms	Training data		Testing data	
	amount	ratio(%)	amount	ratio(%)
Normal	67343	53.46	9711	51.68
Dos	11656	9.25	5741	30.55
Probe	45927	36.46	1106	5.89
R2L	995	0.79	2198	11.70
U2R	52	0.04	36	0.19
Total	125973	100	18792	100

more comparable. After excluding the data that do not appear in the training set, the sample distribution of the NSL-KDD dataset is shown in Table 2.

The CICIDS-2017 dataset is derived from data collected by the Canadian Institute for Cybersecurity (CIC). It contains the 12 latest common attacks, making the dataset more similar to the real communication environment of AMI [55]. In addition, the CICIDS-2017 dataset overcomes the shortcomings of other datasets, such as a lack of traffic diversity, limited coverage of attack types, and the anonymization of packet payload data. The CICIDS-2017 dataset contains a total of 2,830,744 pieces of data, and its subset is chosen as the benchmark for the experiment. According to previous analyses, the DoS GoldenEye, DoS Hulk, DoS Slow HTTP, DoS Slow Loris, and DDoS attacks can be integrated as DoS attacks. Additionally, brute force, SQL injection, and cross-site scripting (XSS) attacks can be integrated as web attacks. The sample distribution of the integrated CICIDS-2017 dataset is presented in Table 3.

4.2 Data preprocessing

The raw KDDCup99, NSL-KDD, and CICIDS-2017 datasets consist of numerical and characteristic features, and labels, yet some of the features are not helpful for the final classification. Moreover, there are some degree of sample distribution imbalance in these datasets, and the characteristic features cannot be directly processed by the DL model, the high values of some numerical features can also interfere with the final intrusion detection results. Therefore, preprocessing is required to reduce the impacts of such factors.

Table 3 Distribution of CICIDS-2017

Algorithms	Training data		Testing data	
	amount	ratio(%)	amount	ratio(%)
Normal	71834	53.68	58166	53.11
Patator	7041	5.26	5959	5.46
DoS	40077	29.95	32923	30.17
Web Attack	1114	0.83	886	0.81
Infiltration	16	0.01	14	0.01
Bot	1031	0.77	869	0.80
PortScan	12710	9.50	10290	9.43
Total	133823	100	109107	100

4.2.1 Feature Screening

Not all features in the KDDCup99, NSL-KDD, and CICIDS-2017 datasets are useful for the final intrusion detection. For example, the nineteenth feature (*num_outbound_cmds*) is zero in both the training and test sets of KDDCup99; moreover, in the training set of NSL-KDD, the ninth feature (*urgent*) only has significant meaning in nine data items, and the percentage of valid values is only 0.07‰. In CICIDS2017, features containing the IP addresses and port numbers of source and target hosts are not helpful for the final classification [36], and it is important to remove these invalid features.

To reduce overtraining caused by invalid features [2], XGBoost is introduced to respectively rank the importance values of features in different datasets, and the ranking results are respectively presented in Figs. 3 and 4. Figure 3 illustrates that the importance values of feature 19 (*num_outbound_cmds*) and feature 8 (*urgent*) in the NSL-KDD dataset are 0 and 6, respectively, so these features are of little help for the final classification. This finding is consistent with the authors' previous speculation.

Accuracy is chosen as the evaluation metric, and feature screening experiments based on XGBoost are conducted depending on the ranking results of the importance values. The screening results of different datasets are reported in Tables 4, 5 and 6. It can be seen from Table 4 that the accuracy first increases and then decreases when features are eliminated one by one according to their importance values, and the highest accuracy is achieved when features with an importance value greater than 25 are retained; thus, features with an importance value greater than 25 are screened. Similarly, to reduce the impact of invalid features, the features in the KDDCup99 and CICIDS-2017 datasets with respective importance values of greater than 29 and 41 are screened.

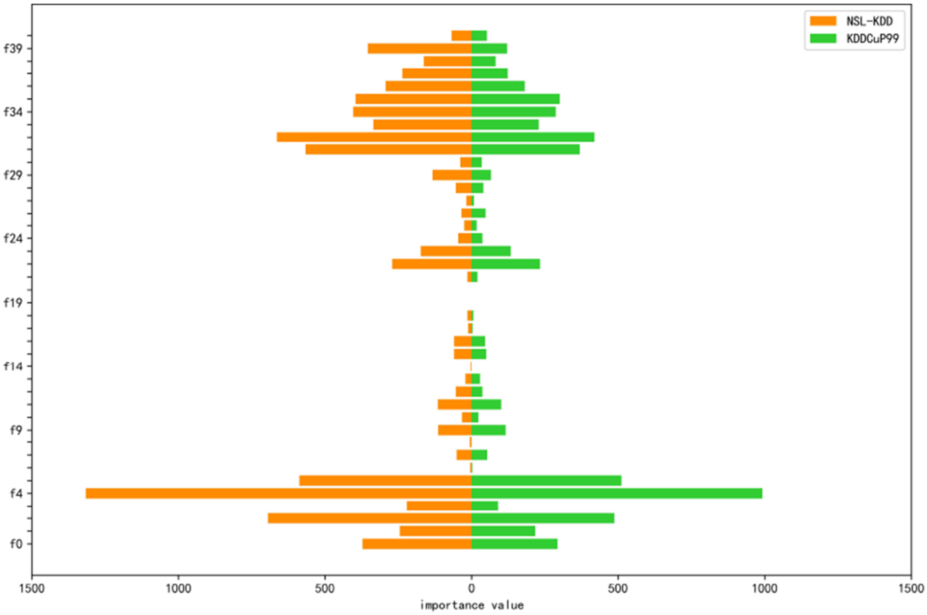


Fig. 3 The importance value of KDDCup99 and NSL-KDD

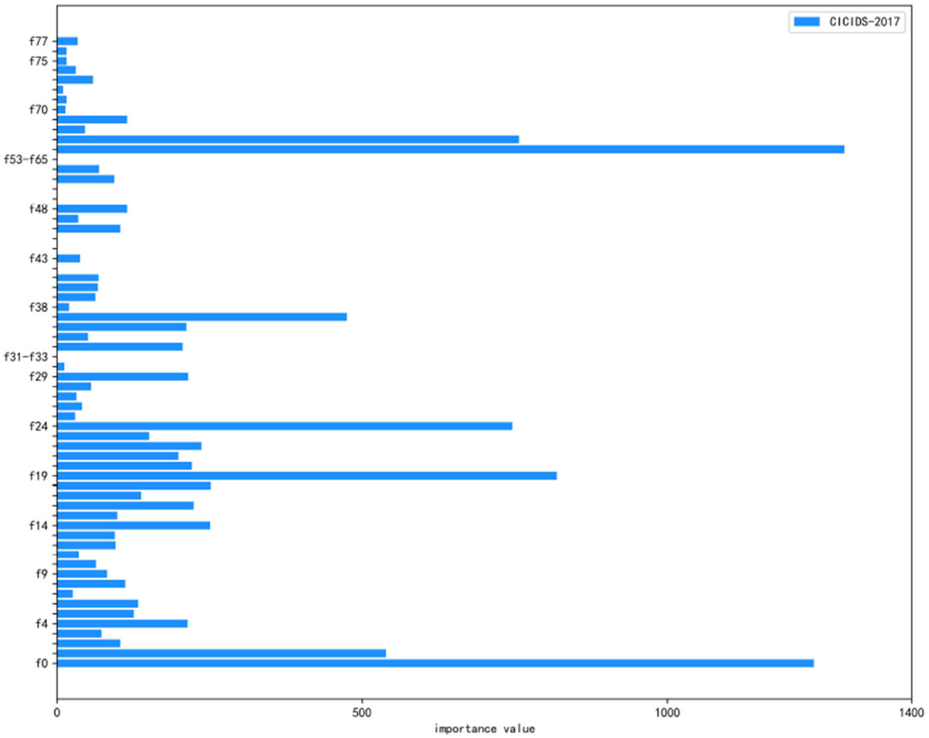


Fig. 4 The importance value of CICIDS-2017

4.2.2 Balanced datasets

Tables 1, 2 and 3 reveal that there are different degrees of distribution imbalance in all three datasets. For example, there are only 52 U2R samples in the KDDCup99 training set, accounting for only 0.01%; this phenomenon is very likely to lead to the underfitting of the model to U2R during the training process. By combining the distributions of samples in these datasets, ADASYN is used to eliminate distribution imbalances. Compared with other techniques, ADASYN can automatically determine the optimal number of samples to be generated, and its generation process incorporates noise, which improves the robustness of the generated data.

4.2.3 Numerical and one-hot

The purpose of numerical and one-hot encoding is to convert the character features into numerical features that can be processed by the DL model. This paper mainly employs numerical and one-hot encoding for features and labels.

Table 4 The feature screening results of KDDCup99

Importance value	>0	>25	>173	>404	>664
Accuracy(%)	75.95	77.16	74.73	66.82	68.13

Table 5 The feature screening results of NSL-KDD

Importance value	>0	>29	>116	>287	>369
Accuracy(%)	92.15	92.69	92.31	90.14	81.03

In KDDCup99 and NSL-KDD, the features that require numerical and one-hot include *Protocol_type*, *Service* and *Flag*. *Protocol_type* consists of three kinds of attributes: TCP, UDP, and ICMP, so its numerical and one-hot results can be represented by 1∗3 dimension vectors (0, 0, 1), (0, 1, 0), and (1, 0, 0). Similarly, Service and Flag contain 70 and 11 attributes, so they can be represented by 1∗70 and 1∗11 dimension vectors separately. In addition, the character labels in these three datasets must also be numerical and one-hot.

4.2.4 Normalization

The main purpose of normalization is to reduce the impact caused by too-strong differences in the same feature. In this study, the max-min normalization method was used to map all features to the interval [0,1], and the specific process is shown in (5):

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{5}$$

where x is the original value, x_{max} and x_{min} represent the maximum and minimum values of the feature, respectively, and x' is the result after normalization. After completing normalization, the original data is mapped to a two-dimensional form for subsequent processing.

5 Experiments and results analysis

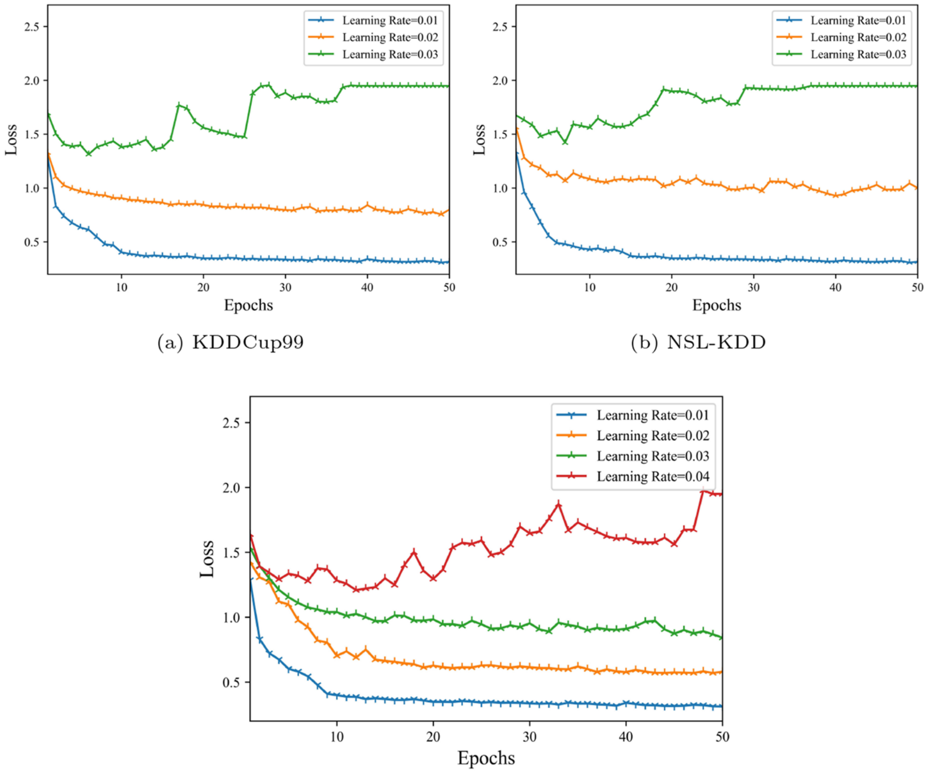
5.1 Experimental environment and hyper-parameters setting

This study is developed on an Intel i7-10700 with Windows 10, and the DL library TensorFlow 2.3-GPU of Python is used to construct the CNN-Transformer hybrid network.

In this paper, the initial interval range of hyper-parameters is first determined according to the characteristics of the datasets, after which the Grid Search method is used to determine the optimal values of different hyper-parameters. For the learning rate as instance, different samples have different categories, but their data characteristics are very similar, if the learning rate is too large, the model will oscillate around the global optimum and converge poorly. Specifically, we used 0.01 as the initial learning rate and step length, and explored the loss variation of the proposed scheme at different learning rates. The experimental results are shown in Fig. 5, the loss value is difficult to decrease and converge to a stable value when the learning rate reaches 0.03 on the KDDCup99 and NSL-KDD datasets, so it can be known that 0.03 is the critical value of the learning rate on these two datasets. Likewise, the critical value is 0.04 on the CICIDS-2017. The reason for having different

Table 6 The feature screening results of CICIDS-2017

Importance value	>0	>16	>41	>82	>151	>252	>539
Accuracy(%)	82.01	84.82	85.11	79.86	73.43	59.39	50.91



(a) KDDCup99

(b) NSL-KDD

(c) CICIDS-2017

Fig. 5 The correlation curve between learning rate and loss

critical values may be that the number of features and the distribution characteristics of these datasets are different. In addition, the AMI intrusion detection system has higher requirements for detection effectiveness, and the smaller learning rate is also beneficial to obtain better detection performance, so a smaller learning rate is appropriate. In this paper, the initial learning rate search range is set from 0.001 to 0.01, and the step length is 0.001. The settings of different hyper-parameters are shown specifically in Table 7.

5.2 Evaluation metrics

Accuracy (*ACC*), precision (*P*), the detection rate (*DR*), the F1-score (*F*), and other indicators are usually employed as evaluation metrics for IDSs [39]. Their definitions are respectively provided by following equations.

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

$$P = \frac{TP}{TP + FN} \tag{7}$$

Table 7 Setting of hyper-parameters

Project	Setting
Conv	4/8
Conv activation function	ReLU
Dense	512
n of Transformer encoder	4
d_{model}	500
Head number	2
Dropout	0.5
Softmax	5/8
Cost function	Cross entropy
Batch size	512
Learning rate	0.001
Optimizer	Adam
Epoch	100

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

$$F = \frac{2 * P * DR}{P + DR} \tag{9}$$

Where *TP* (True Positive) represents the number of normal samples identified correctly, *FP* (False Positive) represents the number of normal samples identified incorrectly, *TN* (True Negative) represents the number of abnormal samples identified correctly, and *FN* (False Negative) represents the number of abnormal samples identified incorrectly.

Among these metrics, *ACC* is the ratio of correctly classified samples to the total sample. While it is the most intuitive evaluation metric, it is not applicable to scenarios with extremely unbalanced sample distributions. *P* is the percentage of true normal samples among the normal samples predicted by the model, and represents the model's detection ability. *DR* is defined as the proportion of actual normal samples that are determined to be normal. *P* and *DR* are mutually inverse, so the model performance can also be determined by *F*, which takes both into account.

To comprehensively evaluate the performance of the proposed IDS, *P*, *DR*, *F* are selected as the main evaluation metrics, and *ACC* is adopted as an auxiliary evaluation metric to judge the behavior of the model in different experiments.

5.3 Experimental design and results

To evaluate the model's performance, four experiments are conducted in KDDCup99, NSL-KDD, and CICIDS-2017.

Experiment 1: Using the above datasets to train the proposed IDS, and its detection ability in different aspect are tested.

We explore the relationship between loss value and epoch firstly, and Fig. 6 shows the experimental results. This figure shows that the CNN-Transformer hybrid network can be stable after at most 10 epochs on different datasets, so the proposed model has good convergence performance.

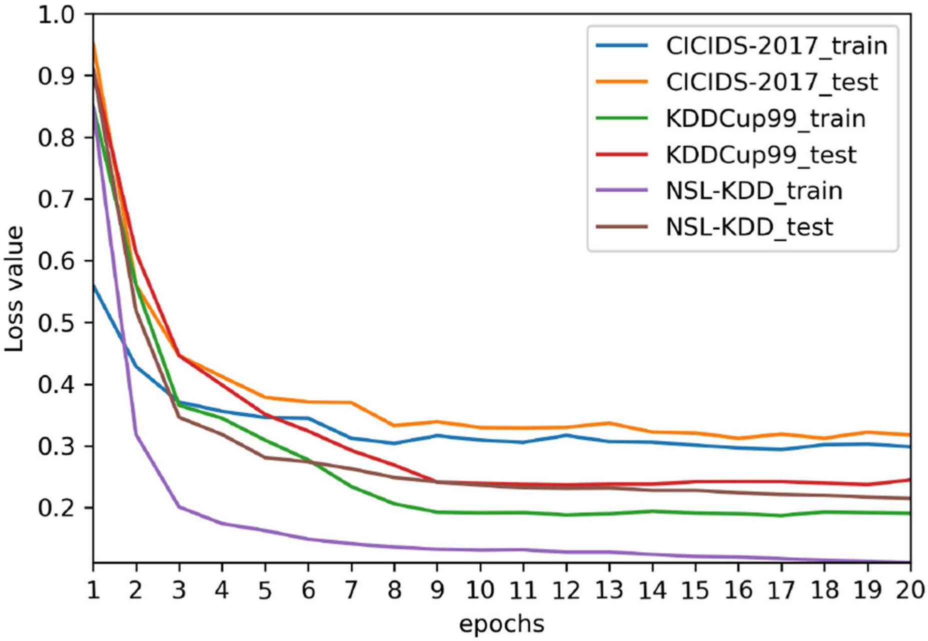


Fig. 6 The relationship of loss and epochs

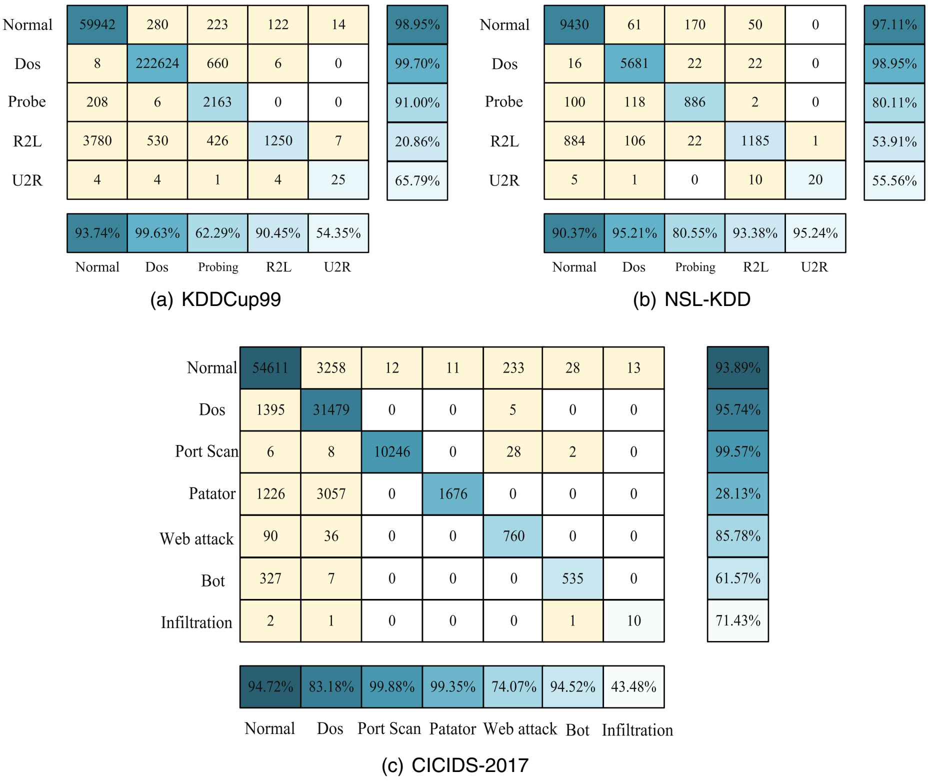
The confusion matrices obtained from different test sets under optimal conditions of parameters are shown in Fig. 7(a), (b), and (c) respectively. It can be seen that the *ACC* of the proposed IDS reaches 97.85%, 91.54% and 91.06% for KDDCup99, NSL-KDD and CICIDS-2017 respectively, and the *P* is greater than 95% for Normal, Dos, Port scan and other types of label. In addition, the *P* of the proposed IDS reaches 85.78% and 71.43% for few-shot attacks, such as Web attack and Infiltration in CICIDS-2017. In KDDCup99 and NSL-KDD, the *P* of few-shot samples, such as U2R, also reaches 55.56% and 65.79% respectively.

To demonstrate the validity of the experimental results, a 10-fold cross-validation experiment is also conducted on the proposed IDS. The confusion matrices obtained from different datasets are shown in Fig. 8(a), (b), and (c) respectively. It can be seen that the *ACC* of the proposed IDS reaches 99.90%, 99.42% and 92.15% separately, and the *P* for Normal, Dos, Probe, and Port Sacn is greater than 99%. In addition, the *DR* and *F* are also generally consistent with the test sets results.

The results of Experiment 1 show that the proposed IDS is more general and has excellent *ACC*, *P* and *DR* performance in different datasets. Moreover, it can focus on subtle feature differences to obtain a better intrusion detection capability in few-shot sample scenes and performs better in convergence.

Experiment 2: The CNN, Transformer, CNN-LSTM, and CNN-Transformer are trained and tested, and the necessity of multi-space feature subsets constructed by CNN and the fine-grained and association constructed by Transformer are verified.

Tables 8, 9 and 10 show the *P* and *DR* of different models in different datasets, respectively. As can be seen from Tables 8 and 9, the single Transformer has deficiencies on both *P* and *DR* when not using CNN to construct multi-feature subsets compared with the proposed CNN-Transformer. For instance, the Transformer has a 41% difference in *P* for U2R



Normal	59942	280	223	122	14	98.95%
Dos	8	222624	660	6	0	99.70%
Probe	208	6	2163	0	0	91.00%
R2L	3780	530	426	1250	7	20.86%
U2R	4	4	1	4	25	65.79%
	93.74%	99.63%	62.29%	90.45%	54.35%	
	Normal	Dos	Probing	R2L	U2R	

(a) KDDCup99

Normal	9430	61	170	50	0	97.11%
Dos	16	5681	22	22	0	98.95%
Probe	100	118	886	2	0	80.11%
R2L	884	106	22	1185	1	53.91%
U2R	5	1	0	10	20	55.56%
	90.37%	95.21%	80.55%	93.38%	95.24%	
	Normal	Dos	Probing	R2L	U2R	

(b) NSL-KDD

Normal	54611	3258	12	11	233	28	13	93.89%
Dos	1395	31479	0	0	5	0	0	95.74%
Port Scan	6	8	10246	0	28	2	0	99.57%
Patator	1226	3057	0	1676	0	0	0	28.13%
Web attack	90	36	0	0	760	0	0	85.78%
Bot	327	7	0	0	0	535	0	61.57%
Infiltration	2	1	0	0	0	1	10	71.43%
	94.72%	83.18%	99.88%	99.35%	74.07%	94.52%	43.48%	
	Normal	Dos	Port Scan	Patator	Web attack	Bot	Infiltration	

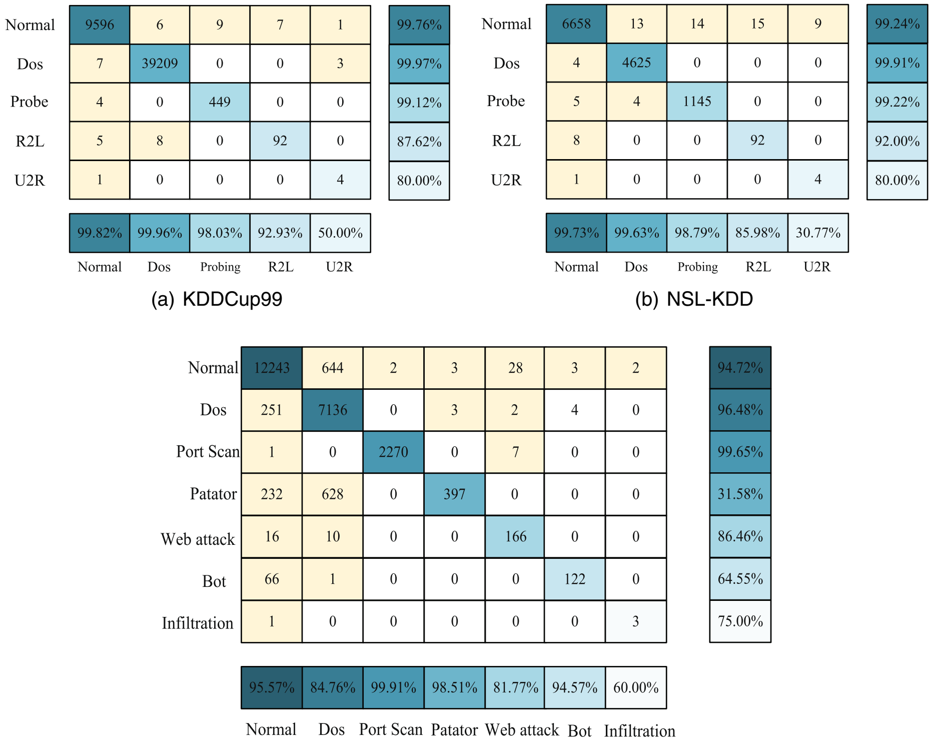
(c) CICIDS-2017

Fig. 7 The confusion matrix for test sets

attacks, compared to the proposed IDS in the KDDCup99 datasets, and this finding demonstrates the necessity of CNN to sufficiently construct multiple feature subsets. On the other hand, it has a maximum 66%(U2R of KDDCup99) and 95%(U2R of NSL-KDD) gap in *P* and *DR*, respectively, compared with the proposed IDS when there is no Transformer. In addition, we replace Transformer with LSTM, which also has a memory function but cannot extract fine-grained features or construct feature association. It also has a difference in*P* and *DR* from 1% to 95% compared to the proposed IDS. All these results prove that the Transformer's ability to conduct fine-grained feature extraction and feature association construction is very helpful for intrusion detection.

The results of Experiment 2, which is an ablation experiment, show that compared with a single Transformer, the proposed scheme effectively enriches the feature space, ensures the diversity of features, and enlarges the feature boundaries through the ability of constructing multiple spatial feature subsets of CNN. Compared with single CNN and CNN-LSTM, the proposed scheme enhances the capability of temporal feature and fine-grained feature extraction by introducing Transformer, and effectively enhances the global association of features by input and position embedding mechanism. The effective improvement and combination of different components make the proposed scheme have better *P* and *DR* capabilities.

Experiment 3: Traditional models, including CNN, LSTM, GRU, Bayes Network, RF, and KNN are trained and tested in different datasets, and the performance of the proposed IDS is compared with the above models.



	Normal	Dos	Probing	R2L	U2R	
Normal	9596	6	9	7	1	99.76%
Dos	7	39209	0	0	3	99.97%
Probe	4	0	449	0	0	99.12%
R2L	5	8	0	92	0	87.62%
U2R	1	0	0	0	4	80.00%
	99.82%	99.96%	98.03%	92.93%	50.00%	

(a) KDDCup99

	Normal	Dos	Probing	R2L	U2R	
Normal	6658	13	14	15	9	99.24%
Dos	4	4625	0	0	0	99.91%
Probe	5	4	1145	0	0	99.22%
R2L	8	0	0	92	0	92.00%
U2R	1	0	0	0	4	80.00%
	99.73%	99.63%	98.79%	85.98%	30.77%	

(b) NSL-KDD

	Normal	Dos	Port Scan	Patator	Web attack	Bot	Infiltration	
Normal	12243	644	2	3	28	3	2	94.72%
Dos	251	7136	0	3	2	4	0	96.48%
Port Scan	1	0	2270	0	7	0	0	99.65%
Patator	232	628	0	397	0	0	0	31.58%
Web attack	16	10	0	0	166	0	0	86.46%
Bot	66	1	0	0	0	122	0	64.55%
Infiltration	1	0	0	0	0	0	3	75.00%
	95.57%	84.76%	99.91%	98.51%	81.77%	94.57%	60.00%	

(c) CICIDS-2017

Fig. 8 The confusion matrix for 10-fold cross-validation

Table 8 The *P* and *DR* of different model in KDDCup99

Metrics	*P*/*DR*				
Attack type	Normal	DoS	Probe	R2L	U2R
CNN	0.99/0.72	0.97/0.99	0.71/0.78	0/0.07	0/0
Trans	0.99/0.81	0.96/0.99	0.78/0.70	0.12 /0.79	0.25/0.34
CNN-LSTM	0.99/0.72	0.97/0.99	0.69/0.81	0.01/0.03	0/0
CNN-Trans	0.99/0.94	1.00/1.00	0.91/ 0.62	0.21/0.90	0.66/0.54

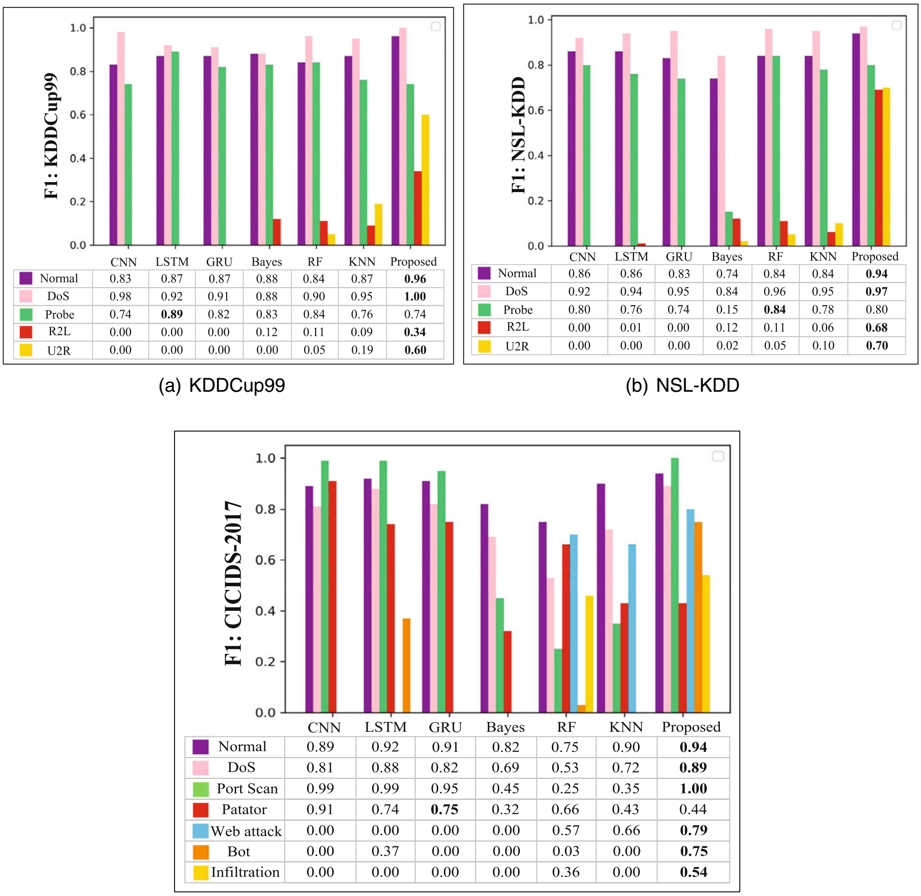
Table 9 The *P* and *DR* of different model in NSL-KDD

Metrics	*P*/*DR*				
Attack type	Normal	DoS	Probe	R2L	U2R
CNN	0.95/0.79	0.91/0.94	0.86/0.74	0/0	0/0
Trans	0.96/0.82	0.93/0.95	0.77/0.57	0.18 /0.91	0.29/0.41
CNN-LSTM	0.95/0.81	0.96/0.95	0.89/0.70	0/0.13	0/0
CNN-Trans	0.97/0.90	0.99/0.95	0.80/ 0.81	0.54/0.93	0.56/0.95

Table 10 The *P* and *DR* of different model in CICIDS-2017

Metrics	*P*/*DR*						
Attack type	Normal	DoS	Port Scan	Patator	Web Attack	Bot	Infiltration
CNN	0.91/0.88	0.83/0.79	1.00/0.99	0.85/0.99	0/0	0/0	0/0
Trans	0.88/0.81	0.76/0.81	1.00/0.99	0.28/0.52	0.81/0.73	0.61/0.84	0.69/0.02
CNN-LSTM	0.92/0.90	0.81/0.72	1.00/0.99	0.89/0.99	0.02/0	0/0	0/0
CNN-Trans	0.94/ 0.95	0.96/ 0.83	1.00/ 1.00	0.28/ 0.99	0.86/ 0.74	0.62/ 0.95	0.71/ 0.43

Considering the contradictory nature of *P* and *DR*, we select *F* as the performance evaluation metric. Figure 9(a), (b) and (c) show the *F* of different models, respectively, where it can be seen that the performance of the proposed IDS is mostly optimal across different datasets. The benefits of the proposed IDS is still evident in few-shot attack detection; for example, the *F* of Infiltration, Bot, and Web attack is improved by 18%, 38%, and 13% in the CICIDS-2017, seperately.



	CNN	LSTM	GRU	Bayes	RF	KNN	Proposed
🟪 Normal	0.83	0.87	0.87	0.88	0.84	0.87	**0.96**
🟪 DoS	0.98	0.92	0.91	0.88	0.90	0.95	**1.00**
🟩 Probe	0.74	**0.89**	0.82	0.83	0.84	0.76	0.74
🟥 R2L	0.00	0.00	0.00	0.12	0.11	0.09	**0.34**
🟨 U2R	0.00	0.00	0.00	0.00	0.05	0.19	**0.60**

(a) KDDCup99

	CNN	LSTM	GRU	Bayes	RF	KNN	Proposed
🟪 Normal	0.86	0.86	0.83	0.74	0.84	0.84	**0.94**
🟪 DoS	0.92	0.94	0.95	0.84	0.96	0.95	**0.97**
🟩 Probe	0.80	0.76	0.74	0.15	**0.84**	0.78	0.80
🟥 R2L	0.00	0.01	0.00	0.12	0.11	0.06	**0.68**
🟨 U2R	0.00	0.00	0.00	0.02	0.05	0.10	**0.70**

(b) NSL-KDD

	CNN	LSTM	GRU	Bayes	RF	KNN	Proposed
🟪 Normal	0.89	0.92	0.91	0.82	0.75	0.90	**0.94**
🟪 DoS	0.81	0.88	0.82	0.69	0.53	0.72	**0.89**
🟩 Port Scan	0.99	0.99	0.95	0.45	0.25	0.35	**1.00**
🟥 Patator	0.91	0.74	**0.75**	0.32	0.66	0.43	0.44
🟦 Web attack	0.00	0.00	0.00	0.00	0.57	0.66	**0.79**
🟧 Bot	0.00	0.37	0.00	0.00	0.03	0.00	**0.75**
🟨 Infiltration	0.00	0.00	0.00	0.00	0.36	0.00	**0.54**

(c) CICIDS-2017

Fig. 9 The *F* of different model

Table 11 The *ACC* of related work in Test set

Dataset	Literature	*ACC*
KDDCup99	Kim et al. [31](LSTM)	96.93%
	Shone et al. [43](NDAE)	97.85%
	Kim et al. [31](KNN)	90.74%
	Kim et al. [31](SVM)	90.40%
	Kim et al. [31](Bayesain)	88.46%
NSL-KDD	Ieracitano et al. [25](AE-DNN)	87.00%
	Shone et al. [43](NDAE)	85.42%
	Zhang et al. [53](RNN)	81.29%
	Ieracitano et al. [25](Q-SVM)	83.65%
	Ieracitano et al. [25](LDA)	83.17%
CICIDS-2017	Gupta et al. [19](CNN)	85.00%
	Gupta et al. [19](DNN)	88.00%
	Gupta et al. [19](LSTM)	86.00%
	Gupta et al. [19](XGBoost)	76.00%
All	Proposed	97.85%/91.54%/91.06%

The experimental results of Experiment 3 show that the proposed scheme effectively reduces the interference of non-technical factors on the intrusion results through feature engineering and data pre-processing, The proposed scheme enriches the feature space and obtains the representation of intrusion samples in different feature spaces through the multi-space feature subset construction capability of CNN. With the Transformer component, the proposed scheme extracts the temporal and fine-grained features more efficiently and constructs the global association of features by input and position embedding. Therefore, the proposed scheme has better performance than other DL and ML models in terms of *F*.

Table 12 The *ACC* of related work in 10-fold cross-validation

Dataset	Literature	*ACC*
KDDCup99	Liu et al. [33](CNN)	98.02%
	Shen et al. [42](ELM)	98.94%
	Zhang et al. [54](GA-ELM)	98.90%
NSL-KDD	Liu et al. [33](CNN)	97.07%
	Shen et al. [42](ELM)	97.58%
CICIDS-2017	Gupta et al. [19](CNN)	(89.68%)
	Gupta et al. [19](DNN)	(90.71%)
	Gupta et al. [19](LSTM)	(91.36%)
	Gupta et al. [19](XGBoost)	(84.56%)
All	Proposed	99.90%/99.42%/92.15%

Experiment 4: The*ACC* of the proposed IDS is compared with related work in different datasets.

The comparison of the testsets is shown in Table 11. Taking the NSL-KDD dataset as an example, as shown in Table 11, the proposed IDS still has at least 4.54% improvement in *ACC* while retaining noisy data, so the proposed CNN-Transformer hybrid network not only has better *ACC* performance but also is more robust. The comparison of 10-fold cross-validation is shown in Table 12. Since there is a lack of studies on 10-fold cross-validation on the CICIDS-2017 dataset in related work, we replicated part of the models based on the hyper-parameter settings in related work and used the CICIDS-2017 dataset for 10-fold cross-validation. In Table 12, the values in parentheses are the *ACC* of the replicated model, and it can be seen that the *ACC* of the proposed scheme is still optimal, so the same conclusion as the testsets can be drawn.

The results of Experiment 4 show that the proposed IDS achieves better *ACC* performance and is more robust than related work in different datasets, and this result is confirmed in both the testsets and the 10-fold cross-validation.

6 Conclusion

A CNN-Transformer hybrid deep neural network model to detect intrusions in AMI is proposed in this paper. The model consist of a cascaded combination of CNN and Transformer, which can focus on the multi-space characteristics of AMI while paying attention to the association and fine-grained characteristics of features. XGBoost-based feature screening and ADASYN-based sample enhancement strategies are also included to reduce the impact of invalid features and sample distribution imbalance. The experimental results on KDD-Cup99, NSL-KDD, and CICIDS-2017 show that the proposed IDS enriches the feature subset by CNN and effectively extracts fine-grained and temporal features by Transformer, and the accuracy of the proposed IDS is 97.85%, 91.04%, and 91.06% respectively. Subsequent research should focus on IDS research in zero-day attack scenarios, which is a typical few-shot classification problem. Although the detection capability of the proposed IDS is significantly improved in this work compared with other research, further improvement is still required. In addition, AMI intrusion detection based on a real communication environment is also a key issue for subsequent research.

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interests The authors declare that there are no conflicts of interest.

References

1. Aa A, Yz A, Mz B (2021) Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks. Inf Sci 577:852–870
2. Aldaweri MS, Ariffin KZ, Abdullah S, Senan MM (2020) An analysis of the kdd99 and unsw-nb15 datasets for the intrusion detection system. Symmetry 12:1666

3. Ali A, Zhu Y, Zakarya M (2021) A data aggregation based approach to exploit dynamic spatio-temporal correlations for citywide crowd flows prediction in fog computing. Multimed Tools Appl 80:31401–31433
4. Ali A, Zhu Y, Zakarya M (2022) Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. Neural Netw 145:233–247
5. Alsharif A, Nabil M, Tonyali S, Mohammed H, Mahmoud M, Akkaya K (2018) Epic: efficient privacy-preserving scheme with e2e data integrity and authenticity for ami networks. IEEE Internet Things J 6:3309–3321
6. Alsharif A, Nabil M, Mahmoud M, Abdallah M (2019) Epda: efficient and privacy-preserving data collection and access control scheme for multi-recipient ami networks. IEEE Access 7:27829–27845
7. Alsharif A, Nabil M, Sherif A, Mahmoud M, Song M (2019) Mdms: efficient and privacy-preserving multidimension and multisubset data collection for ami networks. IEEE Internet Things J 6(6):10363–10374
8. Anderson JP (1980) Computer security threat monitoring and surveillance. James P. Anderson Co., Washington, pp 1–46
9. Ayub N, Aurangzeb K, Awais M, Ali U (2020) Electricity theft detection using cnn-gru and manta ray foraging optimization algorithm. In: 2020 IEEE 23Rd international multitopic conference (INMIC), pp 1–6
10. Benmalek M, Challal Y, Derhab A (2019) Authentication for smart grid ami systems: threat models, solutions, and challenges. In: 2019 IEEE 28Th international conference on enabling technologies: infrastructure for collaborative enterprises (WETICE), pp 208–213
11. Biswas R, Roy S (2021) Botnet traffic identification using neural networks. Multimed Tools Appl 80:24147–24171
12. Choudhary S, Kesswani N (2020) Analysis of kdd-cup'99, nsl-kdd and unsw-nb15 datasets using deep learning in iot. Proc Comput Sci 167:1561–1573
13. Das U, Namboodiri V (2018) A quality-aware multi-level data aggregation approach to manage smart grid ami traffic. IEEE Trans Parallel Distrib Syst PP(2):245–256
14. Das U, Namboodiri V (2019) A quality-aware multi-level data aggregation approach to manage smart grid ami traffic. IEEE Trans Parallel Distrib Syst 30(2):245–256
15. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding: 4171–4186
16. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An image is worth 16x16 words: transformers for image recognition at scale. arXiv:2010.11929
17. Engelbrecht J, Hancke GP, Osifeko MO (2019) Design and implementation of an electrical tamper detection system. In: IECON 2019 - 45Th annual conference of the IEEE industrial electronics society, vol 1. pp 2952–2957
18. Gope P (2020) Pmake: privacy-aware multi-factor authenticated key establishment scheme for advance metering infrastructure in smart grid. Comput Commun 152:338–344
19. Gupta N, Jindal V, Bedi P (2021) LIO-IDS: handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system. Comput Netw 192(19):108076
20. Haddad Z, Mahmoud M, Taha S, Saroit IA (2015) Secure and privacy-preserving ami-utility communications via lte-a networks. :748–755
21. Hasan MN, Toma RN, Nahid AA, Islam M, Kim JM (2019) Electricity theft detection in smart grid systems: a cnn-lstm based approach. Energies 12:3310
22. He Y, Mendis GJ, Wei J (2017) Real-time detection of false data injection attacks in smart grid: a deep learning-based intelligent mechanism. IEEE Trans Smart Grid 8(5):2505–2516
23. Hsu C, Wang S (2021) Hffpnn classifier: a hybrid approach for intrusion detection based opso and hybridization of feed forward neural network (ffnn) and probabilistic neural network (pnn). Multimed Tools Appl
24. Ibrahem MI, Badr MM, Fouda MM, Mahmoud M, Alasmary W, Fadlullah ZM (2020) Pmbfe: efficient and privacy-preserving monitoring and billing using functional encryption for ami networks. In: 2020 international symposium on networks, computers and communications (ISNCC), pp 1–7
25. Ieracitano C, Adeel A, Morabito FC, Hussain A (2020) A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing 387:51–62
26. Ismail M, Shaaban MF, Naidu M, Serpedin E (2020) Deep learning detection of electricity theft cyber-attacks in renewable distributed generation. IEEE Trans Smart Grid 11(4):3428–3437

27. Javaid N, Jan N, Javed MU (2021) An adaptive synthesis to handle imbalanced big data with deep siamese network for electricity theft detection in smart grids - sciencedirect. J Parallel Distrib Comput 153:44–52
28. Jeong JH, Kwon S, Hong MP, Kwak J, Shon T (2019) Adversarial attack-based security vulnerability verification using deep learning library for multimedia video surveillance. Multimed Tools Appl 79:16077–16091
29. Kala TS, Christy A (2020) Hffpnn classifier: a hybrid approach for intrusion detection based opso and hybridization of feed forward neural network (ffnn) and probabilistic neural network (pnn). Multimed Tools Appl 80:6457–6478
30. Khammassi C, Krichen S (2017) A ga-lr wrapper approach for feature selection in network intrusion detection. Comput Secur 70:255–277
31. Kim J, Kim J, Thu H, Kim H (2016) Long short term memory recurrent neural network classifier for intrusion detection. In: International conference on platform technology & service, pp 1–5
32. Kong X, Zhao X, Liu C, Li Q, Li Y (2021) Electricity theft detection in low-voltage stations based on similarity measure and dt-ksvm. Int J Electr Power Energy Syst 125(3):106544
33. Liu G, Zhang J (2020) Cnid: research of network intrusion detection based on convolutional neural network. Discret Dyn Nat Soc 2020:1–11
34. Nabil M, Ismail M, Mahmoud M, Shahin M, Qaraqe K, Serpedin E (2018) Deep recurrent electricity theft detection in ami networks with random tuning of hyper-parameters,740–745
35. Pereira J, Saraiva F (2020) A comparative analysis of unbalanced data handling techniques for machine learning algorithms to electricity theft detection. In: 2020 IEEE congress on evolutionary computation (CEC), pp 1–8
36. Prasad M, Tripathi S, Dahal K (2019) An efficient feature selection based bayesian and rough set approach for intrusion detection. Appl Soft Comput 87(9):105980
37. Punmiya R, Choe S (2019) Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. IEEE Trans Smart Grid 10(2):2326–2329
38. Rahman MA, Asyhari AT, Wen OW, Ajra H, Ahmed Y, Anwar F (2021) Effective combining of feature selection techniques for machine learning-enabled iot intrusion detection. Multimed Tools Appl 80:31381–31399
39. Rajesh Kanna P, Santhi P (2021) Unified deep learning approach for efficient intrusion detection system using integrated spatial–temporal features. Knowl-Based Syst 226:107132
40. Razavi R, Gharipour A, Fleury M, Akpan IJ (2019) A practical feature-engineering framework for electricity theft detection in smart grids. Appl Energy 238:481–494
41. Saeed MS, Mustafa MW, Sheikh UU, Jumani TA, Khan I, Atawneh S, HamadneH NN (2020) An efficient boosted c5.0 decision-tree-based classification approach for detecting non-technical losses in power utilities. Energies 13:3242
42. Shen Y, Zheng K, Wu C, Zhang M, Niu X, Yang Y, Furnell S (2018) An ensemble method based on selection using bat algorithm for intrusion detection. Comput J 61(4):526–538
43. Shone N, Ngoc TN, Phai VD, Shi Q (2018) A deep learning approach to network intrusion detection. IEEE Trans Emerg Top Comput Intell 2(1):41–50
44. Tian C, Su C, Yang C, Zheng Y (2021) Big data analytics for cyber-physical system in smart city, Springer, pp 714–721. In: Atiquzzaman M, Yen N, Xu Z (eds)
45. Tonyali S, Akkaya K, Saputro N, Uluagac AS (2016) A reliable data aggregation mechanism with homomorphic encryption in smart grid ami networks. In: 2016 13Th IEEE annual consumer communications networking conference (CCNC), pp 550–555
46. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser U, Polosukhin I (2017)
47. Wang W, Peng X, Su Y, Qiao Y, Cheng J (2021) Ttpp: temporal transformer with progressive prediction for efficient action anticipation. Neurocomputing 438:270–279
48. Yan Z, Wen H (2020) Electricity theft detection base on extreme gradient boosting in ami. In: 2020 IEEE international instrumentation and measurement technology conference (i2MTC), pp 1–6
49. Yan Z, Wen H (2021) Electricity theft detection base on extreme gradient boosting in ami. IEEE Trans Instrum Meas 70:1–9
50. Yang Z, Ping S, Aijaz A, Aghvami AH (2016) A global optimization-based routing protocol for cognitive-radio-enabled smart grid ami networks. IEEE Syst J 12(1):1015–1023
51. Yang H, Wang F (2019) Wireless network intrusion detection based on improved convolutional neural network. IEEE Access 7:64366–64374
52. Yao R, Wang N, Liu Z, Chen P, Sheng X (2021) Intrusion detection system in the advanced metering infrastructure: a cross-layer feature-fusion cnn-lstm-based approach. Sensors 21(2):626

53. Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:21954–21961
54. Zhang K, Hu Z, Zhan Y, Wang X, Guo K (2020) A smart grid ami intrusion detection strategy based on extreme learning machine. Energies 13:4907
55. Zhang H, Huang L, Wu CQ, Li Z (2020) An effective convolutional neural network based on smote and gaussian mixture model for intrusion detection in imbalanced dataset. Comput Netw 177:107315
56. Zheng Z, Yang Y, Niu X, Dai H-N, Zhou Y (2018) Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. IEEE Trans Ind Inform 14(4):1606–1615

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.