A Novel Approach to Detect Electricity Theft Based on Conv-Attentional Transformer

Junhao Shi[a], Yunpeng Gao[a,*], Dexi Gu[a], Yunfeng Li[a], Kang Chen[a]

[a]*College of Electrical and Information Engineering, Hunan University, 410082, P.R.China*

Abstract

With the establishment of advanced metering infrastructure (AMI), more and more deep learning (DL) methods are being used for electricity theft detection. However there are still problems such as the construction of a DL model that better fits the theft detection task, sample imbalance in the data set and overfitting of the model that limit the potential of DL models. Therefore an novel end-to-end solution based on DL to solve these problems for electricity theft detection is proposed in this paper. First, we construct a model based on Transformer network, which extracts global features from consumption data by calculating the self-attention between each segment obtained by dividing the whole load sequence, and calculates the relative relationship between features for customer classification. Then, we refine the model to further improve its performance. Conv-attentional module is used to embed the input data and capture the local features in each segment. And we minimize the effect of sample imbalance by choosing a suitable loss function, and address the problem of model overfitting by adding normalization layers, dropout regularization and L2 regularization. In addition, grid search is used to determine the optimal values of the model hyper-parameters. Finally, the performance of the proposed model is verified by experiments using the Irish data set. The results show that our method is able to extract features better and thus has higher true positive rate (TPR) with lower false positive rate (FPR) than other state-of-the-art detectors, and it has strong robustness.

Keywords: Electricity theft detection, Transformer, Conv-attentional network, Model overfitting, Feature extraction.

1. Introduction

Electricity theft from customers is the main source of non-technical losses in the operation of the power grid. In addition to the direct loss of net income of power supply companies, illegal operations by customers of electricity theft bring major hidden dangers to the safe and reliable operation of the power system[1]. Currently, the commonly used solutions for electricity theft detection mainly include hardware solutions and data-driven solutions[2]. Hardware-based approaches use hardware devices to monitor variables in the grid such as voltage, current and power to determine whether a customer is stealing electricity[3, 4]. This type of approaches require additional equipment to implement and are therefore costly. Data-driven solutions detect electricity theft customers by mining and analyzing their load profiles and other information. Such methods are more cost-efficient compared to hardware solutions, which can be based on state estimation[5], game theory[6] or machine learning.

With the establishment of advanced metering infrastructure (AMI), attacks on smart meters through digital storage techniques and network communication technologies have made electricity theft more difficult to detect[7]. But at the same time, the massive amount of data provided by smart meters has accelerated the development of data-driven methods for electricity theft detection, especially the most used methods based on machine learning. Single learners including decision tree (DT), and support vector machine (SVM), back propagation neural network (BP) are constructed in [8, 7, 9] for electricity theft

*Corresponding author

Email address: gaoyp@hnu.edu.cn (Yunpeng Gao)

detection, respectively, and the effectiveness of the algorithms is verified by comparative experiments. With the development of machine learning algorithms, ensemble learning algorithms that combine multiple single learners have also been increasingly applied to electricity theft detection. In [10, 11, 12], random forest (RF), Adaboost and XGBoost based electricity theft detection methods are proposed respectively, and experiments prove that ensemble learning algorithms have better performance. Traditional machine learning algorithms are simple to implement and require less data for model training, but it is often necessary to manually select and extract features by applying feature engineering technology, and the workload is large[13]. Punmiya and Choe[14] demonstrate the usefulness of feature engineering for machine learning methods by comparing the impact on the performance of various ensemble learning algorithms when selecting four features i.e. the standard deviation, mean, maximum, and minimum values of daily usage versus selecting raw data as input for electricity theft detection. And this is due to the weakness of machine learning methods for processing high-dimensional data.

DL has a powerful ability to automatically extract features through deep conversion, nonlinearity and abstraction. According to the realization mechanism, it can be divided into unsupervised learning[15, 16], semi-supervised learning[17, 18] and supervised learning[19, 20, 21, 22, 23]. Huang and Xu[15] design a stacked sparse denoising auto-encoder to detect electricity theft customers by extracting abstract features of the data, reconstructing the samples, and comparing the reconstructed error with the size of the set threshold. Fenza*et al.*[16] use a clustering algorithm combined with a long and short term memory (LSTM) network to construct a model that determines whether a customer steals electricity by predicting the customer's next consumption at each instant and comparing its difference with the actual data. In[17], a conditional deep belief network is proposed to detect electricity theft. The model is first trained unsupervised and then fine-tuned using labeled data. Hu *et al.*[18] propose the multi-task feature extracting fraud detector, which combines supervised training with unsupervised training to enable the application of knowledge from both unlabeled and labeled data. Bhat *et al.*[19] design three DL models including convolutional neural network (CNN), LSTM network and stacked auto-encoder (AE), and verify the effectiveness of DL models for electricity theft detection using IEEE123-bus test feeder. In[20], Pereira *et al.* also uses CNN to detect electricity theft, while comparing various oversampling techniques to study the impact of imbalance in the data set. Zheng *et al.*[21] use CNN to extract periodic features of load data transformed into two dimensions and fuse global features of one-dimensional load data captured by a fully connected network for electricity theft detection. In[22], a hybrid neural network of LSTM and multi-layer perceptron (MLP) is proposed which can also extract different types of data features. Ismail,*et al.*[23] consider the problem of electricity theft in the field of distributed generation and propose CNN-GRU hybrid neural network. In[21, 22, 23], the proposed detectors are all hybrid DL models to extract features. The application of deep learning models has improved the effectiveness of electricity theft detection, but there are still problems that limit the potential of the application of the models as follows:

1. *Suitable feature extractor.* Most existing deep learning models are based on CNN or recurrent neural network (RNN). CNN-based models are less likely to capture the global features of the time series data, while RNN-based models cannot compute in parallel and are prone to long-range dependency problem. And they all cannot calculate the relative relationship between the extracted features, leading to their high dependence on the original input data.
2. *Sample imbalance.* The problem of unbalanced data sets is often solved by oversampling in existing electricity theft detection models, which increases the training complexity and aggravates the problem of model overfitting.
3. *Model overfitting.* Theft detection models can suffer from overfitting due to the mismatch between the amount of data in the training set and the complexity of the model, resulting in weak generalization of the model to practical applications. However, there is less research dedicated to

existing electricity theft detection methods.

To solve these problems, we propose a electricity theft detection model based on an improved Transformer network. Transformer is particularly suitable for processing time series, and it captures long-range features in high-dimensional load sequences by calculating the attention coefficients between consumption data at different time periods[24]. The proposed model has a stronger global feature extraction capability compared to CNN and an efficient parallel computing capability compared to RNN. It can also calculate the relative relationship between different features so that it is no longer overly dependent on the original input data when classifying. To further improve the feature extraction capability of the model we use conv-attentional block when embedding load data so that the model can focus on local features in the consumption data. The fusion of two different types of features gives the proposed model excellent performance in electricity theft detection. For the sample imbalance problem in the data set, we compare various loss functions such as weighted cross entropy (WCE), focal loss (FL)[25], gradient harmonizing mechanism classification (GHMC)[26] to deal with this problem and choose the optimal one. And we prevent the model from overfitting by three means. First, we add suitable normalization layers to the conv-attentional block and Transformer blocks, respectively. Second, we add L2 regularization loss to the loss function, which allows the model parameter values to be reduced. Then, dropout regularization is used to randomly stop a certain percentage of neurons at training time.

The main contributions of this paper can be summarized as follows:

1. An electricity theft detection model named CAT based on improved Transformer networks using a conv-attentional module is proposed, which can automatically extract and fuse different types of features in electricity consumption data.
2. The effect of sample imbalance in the data set on the model is reduced by choosing a suitable loss function. The problem of model overfitting is handled by adding normalization layers, dropout regularization and L2 regularization.
3. Experiments of electricity theft detection using Irish data set are conducted to compare the performance with various state-of-the-art methods. Appropriate metrics are selected for evaluation and the effectiveness of the proposed model is verified.
4. The sensitivity of parameters that may change in the actual data set is analyzed to verify the robustness of the proposed model.

The paper is organized as follows. Section 2 introduces the AMI system architecture and the attack model of customer theft. Section 3 presents the framework of the proposed model and the concrete implementation of each component module. In Section 4, comparative experiments are conducted and the results and discussions are presented. Finally we conclude in Section 5.

2. Problem analysis

In this section, firstly the architecture of the AMI system is briefly described, with the techniques and characteristics of electricity theft. Then several types of attack models are introduced to simulate electricity theft.

2.1. AMI system architecture

The schematic diagram of the AMI system is shown in Fig. 1. AMI is a complete network and system for measuring, collecting, storing, analyzing, and applying optimized electricity consumption information, consisting of smart meters installed at the customer's side, a measurement data management system located at the utility company, and an in-home network in the customer's home, as well as a communication system linking them. Each customer is equipped with a smart meter, and the concentrator collects and records customer electricity consumption data from the home area network and sends it to the metering data management system. Normally, there is a difference between the total power recorded by the concentrator and the

power sum of each smart meter, mainly due to line loss and abnormal power consumption behavior of customers who steal electricity. The formula is as follows:

$$\begin{aligned} E_t &= \sum_{i=1}^{N} q_t^{(i)} + \eta + \delta \\ \Delta q_t &= E_t - \sum_{i=1}^{N} q_t^{(i)} \end{aligned} \tag{1}$$

where Δq_t represents the difference between the total power recorded by the concentrator and the power sum of each smart meter. E_t represents the sum of metering data collected by a concentrator during time period t. $q_t^{(i)}$ represents the metering data reported by the smart meter for customer i in time period t. η and δ denote the technical losses due to line losses, etc. and measurement errors, respectively.

If Δq_t is not within a reasonable range, there is abnormal electricity usage by customers within the station. This allows the grid company to focus on monitoring and detecting the station area using electricity theft detection models.

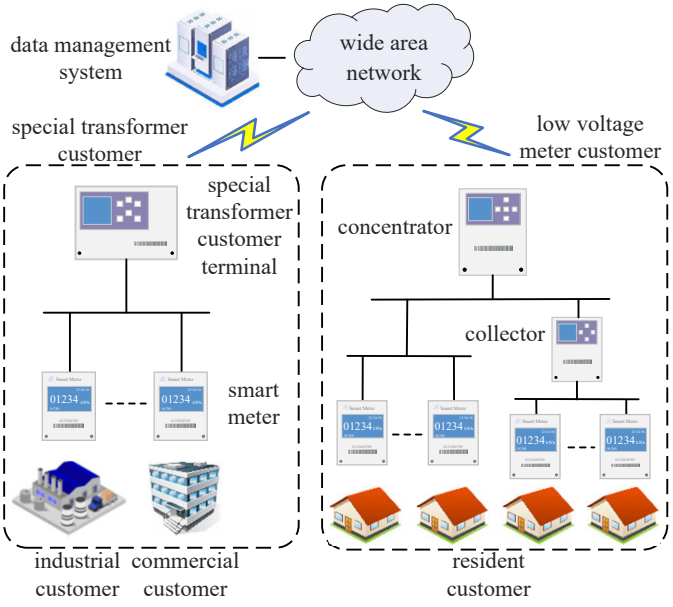


Figure 1: AMI system simplified architecture diagram.

2.2. Attack model

There are many known electricity theft techniques, which can be divided into two major categories in terms of attack methods. That is, physical attacks in which electricity customers bypass or destroy smart meters in a physical way, and network attacks in which customers utilize the network to remotely tamper with the meters readings or use communication technology to invade the data management system[27].

Although electricity theft customers show different patterns of electricity usage due to the specific ways of theft, they all aim to reduce the meter readings. Therefore, different types of False Data Injection (FDI) that reduce meter records can be used to simulate the load data of these electricity theft customers[28]. For a sample x containing m days and n time points per day, we generate six types of FDI as follows:

1. $f_1(x_t^d) = \alpha_t^d x_t^d$, $\alpha_t^d = random(0.2, 0.8)$;
2. $f_2(x_t^d) = max\{x_t^d - \gamma, 0\}$, $\gamma < max\{x_{t=1\ldots n}^{d=1\ldots m}\}$;
3. $f_3(x_t^d) = min\{x_t^d, \gamma\}$, $\gamma < max\{x_{t=1\ldots n}^{d=1\ldots m}\}$;
4. $f_4(x_t^d) = \alpha_t^d x_t^d$,
$$\alpha_t^d = \begin{cases} 0, & t_{start} < t < t_{start} + \Delta t \\ 1, & otherwise \end{cases},$$
$$\Delta t = random(0.2, 0.8) \cdot n,$$
$$t_{start} = random(0, n - \Delta t);$$
5. $f_5(x_t^d) = \alpha mean\{x_{t=1\ldots n}^d\}$, $\alpha = random(0.2, 0.8)$;
6. $f_6(x_t^d) = x_{n-t}^d$.

where x_t^d is the true consumption record at time t of day d; $f_{1\ldots6}$ denotes each of the six FDI types.

In FDI1, the consumption reports are randomly cut by a certain percentage for the entire period of electricity theft. In FDI2, all reports are subtracted by a constant value. In FDI3, consumption reports larger than a constant threshold set randomly are cut off. In FDI4, a random period of time is set in each day when the reports are replaced with zero. In FDI5, the average value of the previous day's consumption is cut by a fixed percentage as the new reports. FDI6 generates new reports by reversing the order of the previous day's consumption. Fig. 2 shows the real consumption reports of an electricity customer for one day with the corresponding six FDI types.

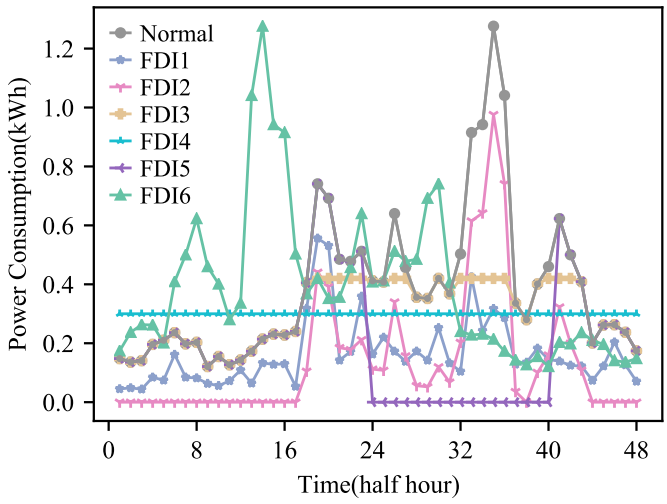


Figure 2: Example of one day consumption and corresponding six types of FDI.

3. Methodology

3.1. Data preprocessing

The electricity consumption data may be null or abnormal. This is because the smart meter may malfunction, such as damage to electrical components, poor contact, and transmission errors.

For some samples with missing data, interpolation is required first. We set the threshold of interpolation to 48, discard samples with missing data points greater than this threshold, and retain samples less than this threshold for interpolation with the following equation:

$$f(t) = \begin{cases} S_i(t), & x_t \in NaN \\ x_t, & otherwise \end{cases} \tag{2}$$

where x_t is the electricity consumption data at time t, NaN represents a null value, $f(t)$ is the value after data processing, $S_i(t)$ is the interpolation function.

The interpolation method we choose is cubic spline interpolation, which can prevent the Runge phenomenon as well as ensure the smoothness of the piecewise connection points of the load sequence. The interpolation equation is as follows:

$$S_i(t) = a_i + b_i t + c_i t^2 + d_i t^3 \tag{3}$$

where a_i, b_i, c_i and d_i are the undetermined coefficients of the interpolation equation of the segmented interval i, which are calculated by the interpolation conditions.

For some samples with incorrect data, we make corrections according to PauTa Criterion with the following equation:

$$f(t) = \begin{cases} \frac{x_{t-1}+x_{t+1}}{2}, & x_t > mean(\mathbf{x}) + 6 \cdot \sigma(\mathbf{x}) \\ x_t, & otherwise \end{cases} \tag{4}$$

where $\mathbf{x}$ is a vector of electricity consumption data over a period of time, $mean(\mathbf{x})$ is the average value of this vector, and $\sigma(\mathbf{x})$ is its standard deviation. We choose $6 \cdot \sigma(\mathbf{x})$ as the limit error instead of $3 \cdot \sigma(\mathbf{x})$, so that we can limit the proportion of outliers to less than 1%, otherwise too much data will be regarded as abnormal data. In addition, since the electricity consumption data are all non-negative numbers, we only consider positive deviations.

3.2. Framework of CAT

This section illustrates the working of the proposed model for electricity theft detection with the model framework as shown in Fig. 3.

As clearly described in this figure, CAT mainly consists of three modules. And the detection process of using this model for electricity theft detection can be summarized as follows:

1. *Sequence Embedding Module:* Given a load sample after data preprocessing, the sequence embedding module first divides it into several segments. Then the conv-attentional module is used to increase the dimension of each segment and the linear layer is used to map the obtained vectors to the selected embedding dimension and add the position encoding.
2. *Feature Extractor:* The resulting encoding vector is input into a feature extractor composed of L-layer Transformer blocks for feature extraction, and a feature vector is obtained.
3. *Classifier:* The classifier gives the probability that the input sample belongs to each category based on the extracted features and thus outputs the specific category.

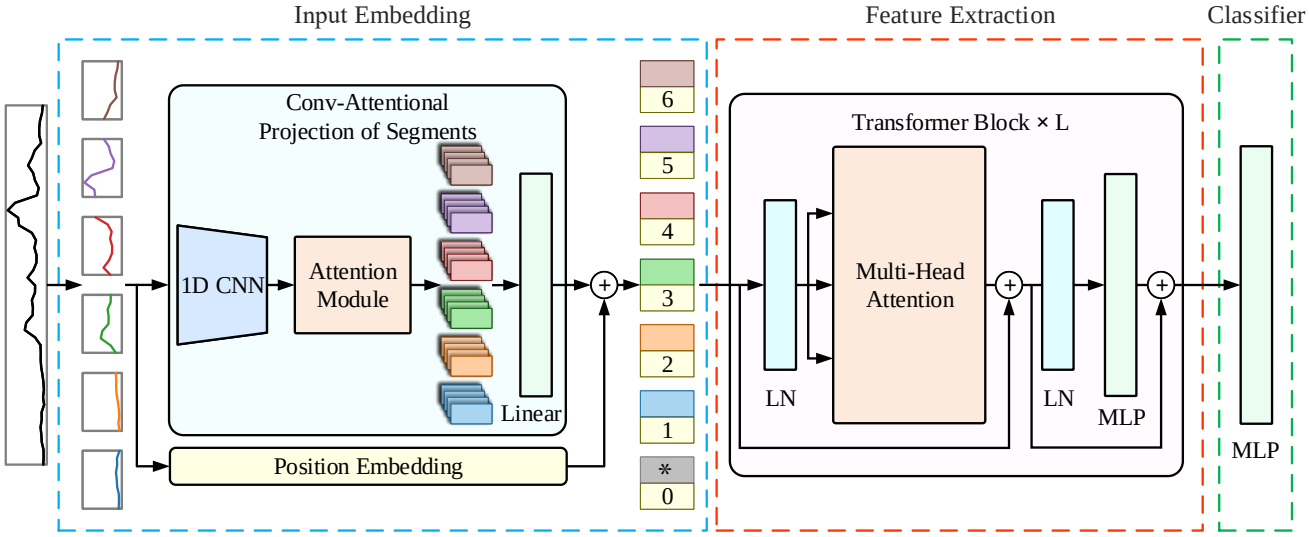


Figure 3: Architecture of the proposed CAT model.

3.3. Conv-attentional sequence embedding

In order for the original load sequence to be input to the Transformer encoder for feature extraction, it needs to be divided into several segments and then converted into a encoding vector. This process is called input embedding. In order to better embed the useful information in the load sequence into the encoding vector, we use conv-attentional projection instead of linear projection. The structure of the conv-attentional module is shown in Fig. 4.

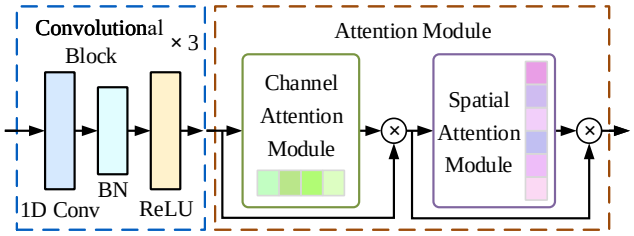


Figure 4: Structure of the conv-attentional module.

As shown in Fig. 4, firstly, m sequences $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}$ of length l obtained by segmenting the load sequence are fed into the 1D-CNN module. This module consists of three stacked convolutional blocks to increase the dimension of each segment and extract local features. The main part of the convolution block is 1D-convolution, which can be expressed as the following equation:

$$\mathbf{h}_j = \sum_{i=1}^{n_i} \mathbf{x}_i * \mathbf{k}_j + \mathbf{b}_j \tag{5}$$

where $\mathbf{k}_j$, $\mathbf{b}_j$ and $\mathbf{h}_j$ represent the jth convolution kernel, bias array and the output map, respectively; $\mathbf{x}_i$ is the input map of the ith channel; $*$ denotes convolution; n_i and n_o are the numbers of input channels and output channels, respectively.

Batch normalization (BN) is to normalize the output of the convolutional layer, thereby speeding up the convergence speed of model training and alleviating overfitting, which can be expressed as:

$$y_b = \gamma \cdot \frac{x_b - \mu(\mathbf{x})}{\sqrt{\sigma^2(\mathbf{x}) + \epsilon}} + \beta \tag{6}$$

where x_b and y_b are the input and output of the ith sample of the current batch respectively; $\mathbf{x}$ represents the row vector formed by b inputs, and b is the size of the batch size; $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ are the mean value and variance of $\mathbf{x}$; γ and β are the scale and shift parameters, respectively.

Rectified linear unit (ReLU) is the activation function, which is used to increase the nonlinear relationship between the layers of the neural network and is expressed as follows:

$$f_{ReLU}(x) = max(0, x) \tag{7}$$

Next, the convolutional block attention module (CBAM) proposed in [29] is used to optimize the feature maps

$\{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m\}$ extracted from the 1D-CNN in both channel and position dimensions. Given a feature map $\mathbf{F} \in \mathbb{R}^{c\times l}$, CBAM first compresses the information of its location dimension to obtain the channel attention vector $\mathbf{V}_c \in \mathbb{R}^{c\times 1}$, and the computation process can be expressed as the following equation:

$$\mathbf{V}_c(\mathbf{F}) = \sigma(f_{MLP}(f^c_{avgp}(\mathbf{F})) + f_{MLP}(f^c_{maxp}(\mathbf{F}))) \tag{8}$$

where $f^c_{avgp}(\cdot)$ and $f^c_{maxp}(\cdot)$ are average-pooling operation and max-pooling operation in the channel dimension, respectively; $\sigma(\cdot)$ denotes sigmoid function; $f_{MLP}(\cdot)$ is the output of the multilayer perceptron (MLP).

MLP consists of two linear transformations with a ReLU activation in between, which can be expressed as the following equation:

$$f_{MLP}(\mathbf{F}) = \mathbf{W}_1(f_{ReLU}(\mathbf{W}_0\mathbf{F})) \tag{9}$$

where $\mathbf{W}_0 \in \mathbb{R}^{d_0\times c}$ and $\mathbf{W}_1 \in \mathbb{R}^{d_1\times d_0}$ are the MLP weights; c, d_0, d_1 are the dimensions of the input layer, hidden layer, and output layer of MLP, respectively.

CBAM compresses the information of the channel dimension to produce the location attention vector $\mathbf{V}_l \in \mathbb{R}^{l\times 1}$ with the following equation:

$$\mathbf{V}_l(\mathbf{F}) = \sigma(f_{conv}(f_{conc}(f^l_{avgp}(\mathbf{F}), f^l_{maxp}(\mathbf{F})))) \tag{10}$$

where $f^l_{avgp}(\cdot)$ and $f^l_{maxp}(\cdot)$ are average-pooling operation and max-pooling operation in the location dimension, respectively; $f_{conc(\cdot)}$ and $f_{conv}(\cdot)$ are the operations of concatenation and convolution, respectively.

For the input feature map $\mathbf{F}$, the final output $\mathbf{F}'' \in \mathbb{R}^{c\times 1}$ of the entire attention module is obtained by successively weighting it with $\mathbf{V}_c$ and $\mathbf{V}_l$, as follows:

$$\begin{aligned} \mathbf{F}' &= \mathbf{V}_c(\mathbf{F}) \otimes \mathbf{F}, \\ \mathbf{F}'' &= \mathbf{V}_l(\mathbf{F}') \otimes \mathbf{F}' \end{aligned} \tag{11}$$

where $\otimes$ denotes element-wise multiplication.

Then, the linear layer is used to transform the resulting m feature maps into encoding vectors of dimension d_e. In addition, we choose the learnable position encoding used by models such as BERT and ViT to embed it in the encoding vector to capture the input order of the segments. Finally, since there is no decoding process using for the classification task, an additional vector for classification, namely class token, is required to be concatenated with the input. It feeds the learned features into the classifier.

3.4. Feature extractor and classification

The feature extractor consisting of L layers of identical Transformer blocks is used to find the interrelationships between the encoding vectors of each segment, thus extracting the global features in the whole load sequence data. This is combined with the local features focused on by the attentional module and fed to the classifier to classify the input samples.

It can be seen from Fig. 3 that feature extraction is a process in which multi-head attention calculates the correlation of the input encoding vectors. First, a linear layer is used to map the encoding vector $\mathbf{V}_i$ obtained by the input embedding into three vectors $\mathbf{q}_i$, $\mathbf{k}_i$, and $\mathbf{v}_i$. Then, the m weight vectors obtained by calculating the similarity between $\mathbf{q}_i$ and $\mathbf{k}_{j=1\ldots m}$ are weighted and summed with $\mathbf{v}_{j=1\ldots m}$ to obtain the output vector $\mathbf{O}_i$ of the self-attention calculation. This attention calculation method is called Scaled Dot-Product Attention, and can be expressed as the following equation:

$$f_{attn}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = f_{softm}(\frac{\mathbf{q}\mathbf{k}^T}{\sqrt{d_e}})\mathbf{v} \tag{12}$$

where $f_{attn}(\cdot)$ is the output of the self-attention; $\mathbf{k}^T$ is the transpose of $\mathbf{k}$; f_{softm} denotes softmax function.

In order to better allows the model to jointly attend to information from different representation subspaces at different positions, multi-head attention is used instead of the single attention function. Its structure is shown in Fig. 5.

As shown in Fig. 5, First, $\mathbf{q}$, $\mathbf{k}$ and $\mathbf{v}$ are different and learned linear projections h times to perform the attention function in parallel. Then the output values are concatenated and linearly projected again to obtain the final values. The calculation pro-

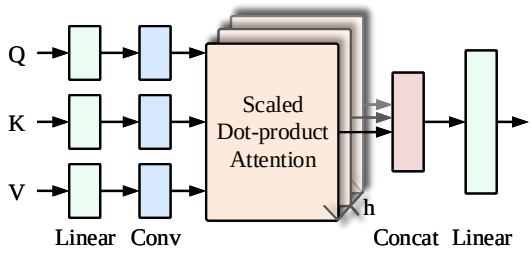


Figure 5: Structure of the multi-head attention.

cess of multi-head attention can be expressed as:

$$\begin{aligned} f_{multihead}(\mathbf{q},\mathbf{k},\mathbf{v}) =& f_{conc}(f_{attn}^{i=1\ldots h}({f_{conv_q}}^{i}(\mathbf{q}\mathbf{W}_q^i), \\ & {f_{conv_k}}^{i}(\mathbf{k}\mathbf{W}_k^i), {f_{conv_v}}^{i}(\mathbf{v}\mathbf{W}_v^i)))\mathbf{W}^o \end{aligned} \tag{13}$$

where $f_{attn}^{i=1\ldots h}(\cdot)$ denotes the attention function performed h times; $\mathbf{W}^i$ and ${f_{conv}}^{i}(\cdot)$ are the weights of the linear layer and the operation of convolution at the ith time.

The convolutional layer is added after the linear layer to reduce the dimensions of $\mathbf{k}$ and $\mathbf{v}$ to save computing resources and reduce the amount of attention calculation parameters, while keeping the $\mathbf{q}$ dimension unchanged to ensure that the dimension of the extracted features does not decrease.

In addition to the attention sub-layer, a feed-forward network sub-layer composed of MLP is also needed. Each sub-layer uses layer normalization (LN) and adds residual connection. Similar to BN, LN can ensure the stability of feature data distribution to accelerate the convergence speed of the model training. Residual connection is to solve the problem of gradient dissipation and weight matrix degradation that occur as the network deepens. The whole process of Transformer block can be expressed as follows:

$$\begin{aligned} \mathbf{I}' &= f_{LN}(\mathbf{I} + f_{multihead}(\mathbf{I})), \\ f_{block}(\mathbf{I}) &= f_{LN}(\mathbf{I}' + f_{MLP}(\mathbf{I}')) \end{aligned} \tag{14}$$

where $f_{LN}(\cdot)$ is the operation of layer normalization; $f_{block}(\cdot)$ is the output of the Transformer block.

The feature extractor consists of L Transformer blocks in series and its output $\mathbf{F}_O \in \mathbb{R}^{1\times d_e}$ are the extracted features. The classifier composed of MLP classifies each sample according to the feature map $\mathbf{F}_O$.

The classifier is composed of an MLP, which consists of two linear layers with a GELU activation in between. Its input is the features extracted by the Transformer Encoder, i.e. the class token used for classification. The number of fully connected neurons in the last layer is 1. The Sigmoid activation function is used to convert the output into a probability value between 0 and 1. Customers with probability values greater than or equal to a threshold of 0.5 are considered as electricity thieves, while the opposite is true for normal customers.

4. Experimental results and discussion

We use real smart metering consumption data to construct electricity theft detection dataset based on the attack models selected in Section 2. Then adequate experiments are used to evaluate the proposed approach. The open-source deep learning framework pytorch is used to build our model and all experiments are conducted on a PC equipped with an i5-6300HQ CPU with 16-GB RAM and an NVIDIA GeForce GTX 950M GPU.

4.1. Dataset

The performance of the proposed model is evaluated by using the data set provided by the Irish CER Smart Metering Project[30]. The data set contains load data of more than 5,000 Irish households and business customers for more than 500 days from 2009 to 2010, with 48 collection points per day. The detection window is set to 7 days instead of 1 day in order to improve the accuracy of the detection. To study the electricity theft detection model under sufficient data training, we divide 4232 household customer consumption data to obtain new samples with a division interval of 7 days. Then the resulting data set is divided into a training set, a validation set, and a test set in the ratio of 6:2:2. Finally the same proportion of samples in each data set are randomly selected to generate electricity theft samples, with the same number of FDI samples for each of the six types. We first set the proportion of electricity theft samples to 15% for the experiments.

4.2. Evaluation metrics

The essence of electricity theft detection is a classification task. Positive class samples represent electricity theft customers while negative class samples represent normal customers. The confusion matrix can be constructed to evaluate the performance of the theft detection model, as shown in Table 1.

Table 1: Confusion matrix of electricity theft detection result

Samples	Predicted negative	Predicted positive
Actually negative	TN	FP
Actually positive	FN	TP

Based on the confusion matrix shown in Table 1, we can obtain the evaluation metrics of the results including accuracy (ACC), false positive rate (FPR), ture positive rate (TPR), precision (Pre) and F1-Score (F1), with the following equations:

$$\begin{aligned} FPR &= (FP)/(FP+TN) \\ TPR &= (TP)/(TP+FN) \\ Pre &= (TP)/(TP+FP) \\ F1 &= (2\times TPR\times Pre)/(TPR+Pre) \\ ACC &= (TP+TN)/(TP+TN+FP+FN) \end{aligned} \tag{15}$$

ACC indicates the overall detection accuracy rate of normal and theft customers; FPR indicates the level of misjudging normal customers as theft customers; TPR indicates the ability to detect theft customers without missing them; Pre indicates the ability to detect theft customers accurately; F1 is a comprehensive index of the ability to detect theft customers completely and accurately. In addition, the receiver operating characteristic (ROC) curve describes the relative relationship between FPR and TPR as it varies. It is suitable for evaluating the overall performance of the classifier when the data set is unbalanced. The larger the area under the curve (AUC), the better the model performance.

4.3. Analysis of model hyper-parameters

In the process of building the CAT model, there are some hyper-parameters that need to be set. We determine the optimal hyper-parameters by comparing the results of the validation set.

The number L of Transformer block, the number of heads h of multi-head attention, and the encoding vector dimension d_e jointly determine the complexity of the proposed model. The larger their values are, the more capable the model is, but at the same time the more difficult it is to train and the more likely the model is to be overfitted. The range of these three hyper-parameters grid search and the results of the selection are shown in Table 2.

Table 2: Selection of Model Hyper-Parameters

Hyper-Parameters	Range of Values	Optimal Value
L	4, 5, 6, 7, 8	6
h	4, 6, 8	6
d_e	32, 64	32

The performance of the model also varies when the length l of the segments into which the input sequence is divided varies. The optimal value of l is searched as 8.

The appropriate loss functions for the classification task using sample imbalanced data sets are WCE, FL, and GHMC. WCE imposes different weights to different classes of samples. FL weights samples that are difficult to classify in addition to small classes of samples. GHMC focuses on samples with moderate classification difficulty. To evaluate the performance of the classifier with the stability of its classification results when trained with these three loss functions separately, 5-fold cross-validation is used. That is, for each loss function, there are five sets of ACC and AUC values, as shown in each point in Fig. 6(a), (b). Fig. 6(c), (b) show the mean and standard deviation of the five results.

As shown in Fig. 6, the mean value of the results of the models trained using WCE is the largest, while the maximum value using FL is the largest. However, the distribution of the five results of the models trained with FL is more discrete with a larger standard deviation, while the standard deviation of the results obtained with WCE is the smallest. In summary, the model trained using WCE has a more stable process and the best overall results. And the weight factor of WCE can be flexibly adjusted according to the different degree of imbalance of

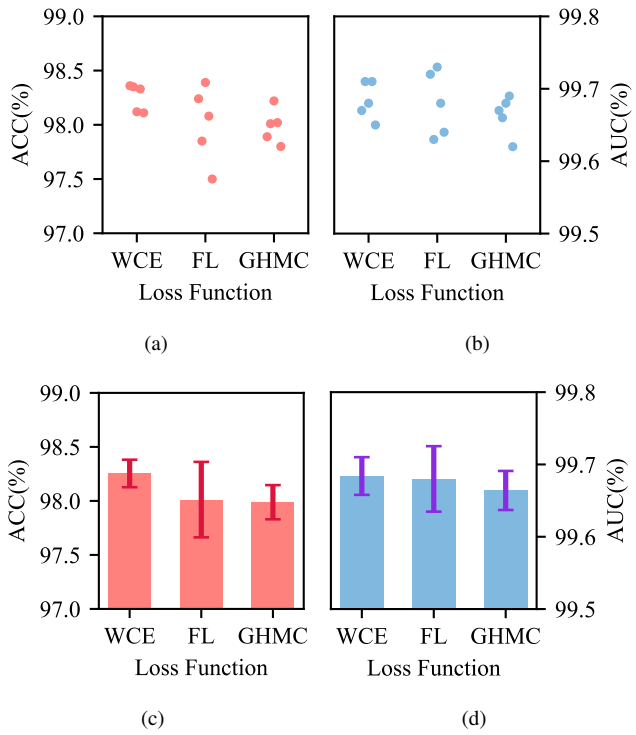


Figure 6: Model performance using cross-validation with different loss functions. (a) Scatter plot of ACC. (b) Scatter plot of AUC. (c) Histogram with error bars of ACC. (d) Histogram with error bars of AUC.

data sets. Using FL as the loss function, the model classification results can achieve high AUC and ACC, but the results are more volatile. And since two interacting parameters need to be tuned when the sample imbalance changes, its optimization search cost is high. The performance of GHMC is worse than the other two loss functions. For comprehensive comparison, we choose WCE as the loss function.

In addition, to solve the overfitting problem of the model, we add L2 regularization loss to the loss function and add dropout layer to the model. Then Adam optimizer is selected to update the weights of the model. We search for the optimal values of the regularization coefficient, dropout rate, and learning rate to be 0.001, 0.5, and 0.002, respectively. After determining each hyperparameter of the model, the loss and ACC of the training process of the model are shown in Fig. 7. It can be seen that the training process is smooth. Moreover, the loss and ACC of both training and validation sets are close to each other, indicating that the model does not have the problem of underfitting or overfitting.

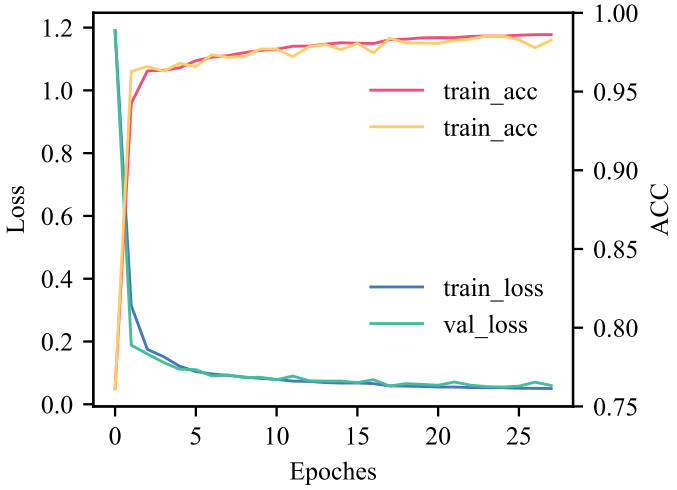


Figure 7: The ACC and loss of the CAT model during training.

4.4. Performance comparison with other methods

To show the superiority of the proposed method, we compare the performance of the proposed CAT model with nine other machine learning algorithms that perform well in electricity theft detection. These methods can be categorized into three groups: individual learners (IL), ensemble learning (EL) and deep learning (DL).

The Scikit-learn library is used to implement the IL and EL algorithms, and the opensource framework Pytorch is used to build the DL models. The optimal hyper-parameters of the models are obtained by the random search, as described in Table 3.

In the first comparison experiment, we compare and analyze the electricity theft detection capabilities of various methods, whose ROC curves and AUC values are shown in Fig. 8. It can be seen that the proposed method has the steepest ROC curve. The area under the curve is the AUC of the model, and the length of the bar indicates its size. The various metrics for each model are shown in Table 4.

As can be seen in Table 4, our method achieved the highest ACC of 98.23%. The TPR and FPR of our method are 94.15% and 1.05%, respectively. This indicates that 94.15% of the samples that are actually electricity theft customers are detected and

Table 3: Brief description of comparison method and hyper-parameters selection

Methods		Brief Description and Hyper-Parameters Selection
IL	DT[8]	DT implements sample classification based on tree theory. The maximum depth of the tree is 5, and the minimum number of samples required for nodes subdivision and leaf nodes are 10 and 2, respectively.
	SVM[7]	SVM separates the different classes of samples using a hyperplane. The kernel function is the radial basis function, and the penalty factor is set as $C = 20$.
	BP[9]	BP is the simplest neural network, containing a hidden layer and an output layer. The number of neurons in the hidden layer is 128 and the learning rate is 0.01.
EL	RF[10]	RF builds a decision tree forest based on the idea of bagging, and uses majority voting to obtain the final result. AdaBoost is based on the idea of boosting, which is a linear combination of multiple weak learners. XGBoost is an efficient system implementation of gradient boosting tree. Their weak learners are all DT with a number of 50. And the learning rate is 0.1.
	AdaBoost[11]	
	XGBoost[12]	
DL	LSTM[19]	The network contains 3 LSTM layers, each of that has 64 neurons and is followed by a dropout layer with 25% dropout rate. linear layers are used for classification.
	CNN[19]	CNN stacks two 1D-convolutional layers followed by a max-pooling layer twice to extract features and uses linear layers for classification. The kernels are all of size 3 and the number is 64, 32, 16 and 16, respectively.
	CNN-GRU[23]	CNN-GRU uses a 1D-convolutional layer with 64 convolution kernels of size 3 followed by 4 GRU layers with 64 neurons to extract features, and then uses a linear layer for classification.

1.05% of the samples that are actually normal customers are incorrectly predicted. These two metrics show that the model is able to check out more electricity theft customers while still ensuring that normal customers are not mispredicted. Pre and TPR are a pair of tradeoff quantities, and F1 is the harmonic average of these two metrics, reflecting the ability of the model to detect more and more accurately electricity theft customers. The F1 of our method is 94.11%, which is higher than other methods. LSTM achieves the highest Pre, but the TPR is lower, so its F1 is not high. AUC is a metric that combines TPR and FPR and considers all classification thresholds. It is not affected by the imbalance of the dataset and indicates the comprehensive performance of the model. The proposed method achieves the highest value of AUC of 99.71%, which is higher than other deep learning methods by 0.7%-2.6%, higher than ensemble learning methods by 4.5%-6.3%, and higher than single classifiers by 6.3%-10%.

In the second comparison experiment, we analyze the detection ability of the proposed method for six types of electricity theft customers, and the confusion matrix of the detection results is shown in Fig. 9. The confusion matrix can visualize the percentage of confusion between each type of customers. Each

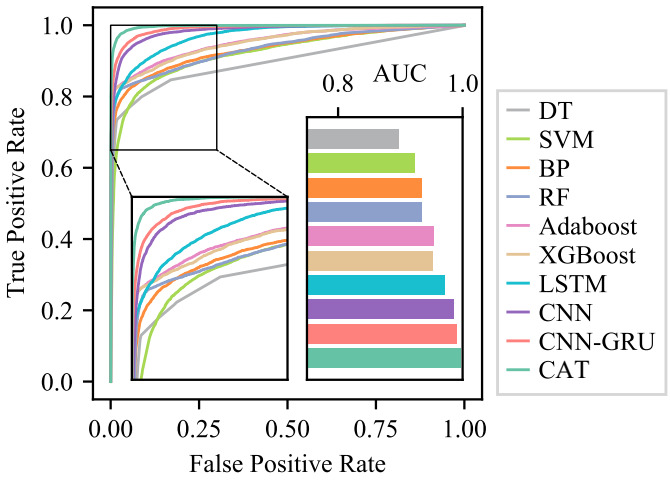


Figure 8: ROC curves of the proposed method and other methods and the value of the area under each curve AUC.

value in the matrix represents the proportion of customers who are actually of the type shown in the vertical coordinate and are predicted to be of the type shown in the horizontal coordinate. Based on the confusion matrix, other evaluation metrics can be obtained. The proposed method is compared with the best performing CNN-GRU among other methods, and the results are shown in Table 5.

As can be seen from Fig. 9, the most easily confused of the

Table 4: Comparison of performance metrics of various methods

Methods	Metrics(%)					
	AUC	ACC	FPR	TPR	Pre	F1
DT	89.78	94.49	1.82	73.55	87.71	80.01
SVM	92.22	93.27	2.14	67.27	84.72	74.99
BP	93.39	95.03	0.94	72.18	93.11	81.32
RF	93.36	95.58	1.59	79.54	89.84	84.37
Adaboost	95.34	95.86	1.79	82.53	89.06	85.67
XGBoost	95.23	96.54	0.48	79.69	96.67	87.36
LSTM	97.12	95.89	**0.37**	74.69	**97.27**	84.5
CNN	98.52	96.51	1.66	86.19	90.14	88.12
CNN-GRU	99.04	97.05	1.58	89.32	90.88	90.09
CAT	**99.71**	**98.23**	1.05	**94.15**	94.08	**94.11**

seven types of customers is the first type of electricity theft customers and normal customers, with 11% of the customers who are actually the first type of electricity theft customers being incorrectly predicted as normal customers. In addition, 7% of the customers who are actually the sixth type of electricity theft are incorrectly predicted to be normal customers. The proportion of confusion among the remaining types is relatively small, all below 2%.

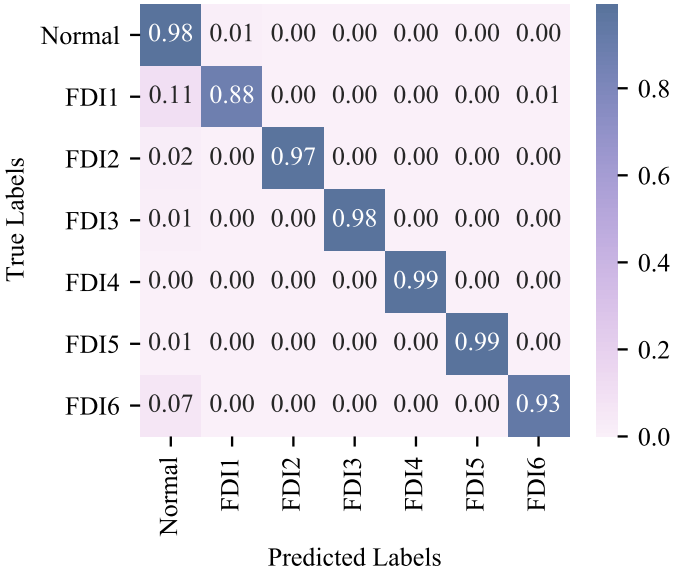


Figure 9: Confusion matrix of classification results for different categories of samples using CAT model.

As can be seen from Table 5, for various types of customers,

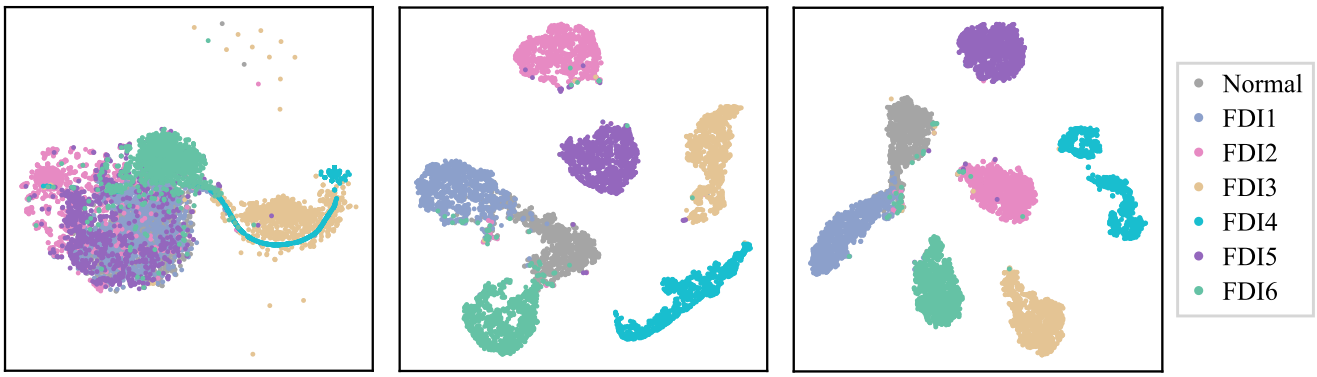
Table 5: Comparison of multi-classification results by different methods

Sample Types	Methods	Metrics(%)					
		AUC	ACC	FPR	TPR	Pre	F1
Normal	CNN-GRU	99.28	97.34	11.3	98.86	98.02	98.44
	CAT	99.76	98.16	3.67	98.48	99.35	98.91
FDI1	CNN-GRU	98.42	98.28	0.66	57.31	68.85	62.55
	CAT	99.51	98.75	0.98	88.33	69.79	77.97
FDI2	CNN-GRU	99.87	99.79	0.12	96.11	95.4	95.75
	CAT	99.96	99.82	0.12	97.41	95.38	96.38
FDI3	CNN-GRU	99.99	99.82	0.16	99.07	94.02	96.48
	CAT	99.99	99.85	0.11	98.43	95.68	97.04
FDI4	CNN-GRU	100.0	99.95	0.03	99.07	98.8	98.93
	CAT	100.0	99.98	0.0	99.35	100.0	99.67
FDI5	CNN-GRU	99.98	99.8	0.01	92.59	99.4	95.87
	CAT	99.99	99.96	0.02	99.26	99.08	99.17
FDI6	CNN-GRU	99.59	99.54	0.09	84.81	96.22	90.16
	CAT	99.84	99.68	0.14	92.87	94.36	93.61

the AUC and ACC of the proposed method are greater than or equal to the results of the CNN-GRU model, indicating that our method has a good comprehensive detection capability for each type of electricity theft customers.The TPR of our method for detecting the third type of electricity theft customers is slightly lower than that of the CNN-GRU model by 0.6%, but relatively, its Pre is 1.7% higher and FPR is 0.05% lower. This indicates that CNN-GRU detects a little more customers in this type but at the same time more customers not in this category are misclassified as such. The similarity is also observed for the fifth and the sixth type of electricity theft customers. For these two types of customers, CNN-GRU has higher Pre and lower FPR, but relatively its TPR is lower than our method. And the F1 of our method is higher than that of CNN-GRU for all types of customers.

4.5. Feature visualization

To show the superiority of our method for feature extraction of electricity consumption data, the t-Distributed Stochas-



(a) (b) (c)

Figure 10: Visualization of feature extraction results of different models based on t-SNE. (a) Original input data. (b) Features extracted by CNN-GRU. (c) Features extracted by CAT.

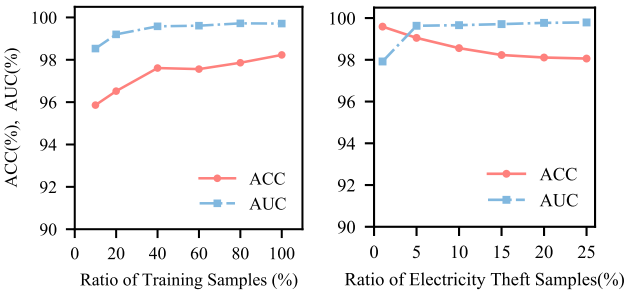
tic Neighbor Embedding (t-SNE) algorithm is used to visualize the features. The complexity and the number of iterations of the t-SNE algorithm are set to 50 and 1500, respectively. As shown in Fig. 10, seven different colors indicate seven types of customers, and each type shows 1000 customers. Fig. 10(a) shows the results of the visualization of the raw load data. It can be seen that the samples of different types overlap severely, which indicates that the classification performance using the raw load data is poor. Fig. 10(b) shows the result of visualizing the features extracted by the CNN-GRU model, and it can be observed that the normal customers are not well separated from the first type of electricity theft customers, and there is also a lot of overlap with the sixth type of electricity theft customers. Fig. 10(c) shows the result of the visualization of 32 features extracted by our method. It can be seen from the figure that customers of the same type are clustered very well. There is only little overlap of samples between normal customers and the first type of electricity theft customers that are difficult to separate, and the separation effect among other types of customers is better.

In summary, the proposed model embeds customers electricity consumption data and captures local features in each segment through the constructed conv-attentional module, and then extracts sequence global features using Transformer's self-attentive structure, which shows strong feature extraction capability.

4.6. Sensitivity analysis

In order to verify the robustness of the proposed model when the data situation is different in reality, we conduct sensitivity experiments to analyze the impact of different sample sizes and different imbalances in the data sets on the model performance.

First, we train the model using 10%, 20%, 40%, 60% and 80% samples of the original training set to analyze the effect of the training sample size on the performance of our model. The performance metrics are shown in Table 6, and the visualization of ACC and AUC is shown in Fig. 11(a).



(a) (b)

Figure 11: Performance of the proposed model with varying parameters of the data set. (a) Performance with different ratios of training sets. (b) Performance with different ratios of electricity theft samples.

As shown in Table 6 and Fig. 11(a), the performance of the proposed model decreases slightly when the ratio of training samples used in the original training set is reduced. And when the ratio of training samples is higher than 40%, the perfor-

Table 6: Model performance with different ratios of training samples

Training Samples Ratio (%)	Metrics(%)					
	AUC	ACC	FPR	TPR	Pre	F1
10	98.53	95.86	2.94	89.06	84.24	86.58
20	99.2	96.52	2.96	93.61	84.79	88.98
40	99.58	97.61	1.87	94.66	89.95	92.25
60	99.61	97.56	2.11	95.74	88.88	92.18
80	99.72	97.86	1.95	96.76	89.76	93.13
100	99.71	98.23	1.05	94.15	94.08	94.11

mance of the model is almost constant and the decrease of AUC is less than 0.2%. As the number of training samples decreases, the AUC decreases by 1.2% when 10% of the samples are used for training. The other evaluation metrics of the model show similar trends. The results show that our model performance reaches its optimal value when a certain number of samples is reached, i.e., 40% of the original training set size.

Then, we analyze the impact of the proportion of electricity theft samples in the original dataset, i.e., the degree of imbalance, on the performance of our model. The degree of imbalance is set to 1%, 5%, 10%, 15%, 20% and 25% respectively. The test results are shown in Table 7, and the visualization of ACC and AUC is shown in Fig. 11(b).

Table 7: Model performance at different levels of sample imbalance

Electricity Theft Sample Ratio (%)	Metrics(%)					
	AUC	ACC	FPR	TPR	Pre	F1
1	97.92	99.59	0.14	72.45	84.37	77.96
5	99.63	99.05	0.27	86.11	94.32	90.03
10	99.66	98.56	0.5	90.12	95.25	92.61
15	99.71	98.23	1.05	94.15	94.08	94.11
20	99.77	98.11	1.55	96.77	93.96	95.35
25	99.79	98.06	1.42	96.51	95.78	96.14

As can be seen from Table 7 and Fig. 11(b), when the proportion of electricity theft samples is above 1%, the AUC is basically unchanged as the proportion decreases, although the degree of sample imbalance increases. When the proportion decreases to 1%, the model performance decreases faster and the AUC decreases by 2%. In addition to AUC, ACC increases slightly, TPR and FPR decrease, Pre remains the same, and F1 decreases as the ratio of theft samples decreases, but the changes of each metric are small when the ratio is above 1%. It can be concluded that in most cases the overall performance of our model changes very little as the percentage of customers who steal electricity decreases.

In summary, our model has good robustness. The model performance can remain stable when the training set size and the sample imbalance degree vary within a certain range.

5. Conclusion

In this paper, we propose an electricity theft detection model based on improved Transformer networks. To the best of our knowledge, this is the first time that Transformer network is applied to the field of electricity theft detection, and we embed raw electricity consumption data using a conv-attentional module to improve the method. Therefore the proposed model can extract the global features of the long-range load sequence with the local features in each segment of the sequence obtained by division, and calculate the relative relationship between the features. The grid search method is used to determine the optimal values of the model hyper-parameters. We conduct experiments using the Irish data set to verify the strong feature extraction capability of the proposed model through feature visualization. The experimental results also show that the model can handle unbalanced data set well and has strong generalization ability without overfitting. The comparison with other state-of-the-art detectors shows that the performance is better than other DL models, ensemble learning models and single classifiers. In addition, the results of sensitivity analysis show that our proposed model is robust to the degree of imbalance and the size of the training set that may change in the actual data set.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant NO.51777061 and the Science and Technology Project of China Southern

Power Grid Guangxi Electric Power Company under Grant NO.CXKJXM20200020.

References

[1] A. Nizar, Z. Dong, Y. Wang, Power utility nontechnical loss analysis with extreme learning machine method, IEEE Trans. Power Syst. 23 (2008) 946–955.

[2] J. L. Viegas, S. M. Vieira, R. Melício, V. Mendes, J. M. Sousa, Classification of new electricity customers based on surveys and smart metering data, Energy 107 (2016) 804–817.

[3] Y. Zhou, Y. Liu, S. Hu, Energy theft detection in multi-tenant data centers with digital protective relay deployment, IEEE Transactions on Sustainable Computing 3 (2017) 16–29.

[4] N. V. Patil, R. S. Kanase, D. R. Bondar, P. Bamane, Intelligent energy meter with advanced billing system and electricity theft detection, in: 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), IEEE, 2017, pp. 36–41.

[5] J. B. Leite, J. R. S. Mantovani, Detecting and locating non-technical losses in modern distribution networks, IEEE Trans. Smart Grid 9 (2016) 1023–1032.

[6] C.-H. Lin, S.-J. Chen, C.-L. Kuo, J.-L. Chen, Non-cooperative game model applied to an advanced metering infrastructure for non-technical loss screening in micro-distribution systems, IEEE Trans. Smart Grid 5 (2014) 2468–2469.

[7] P. Jokar, N. Arianpoo, V. C. Leung, Electricity theft detection in ami using customers' consumption patterns, IEEE Trans. Smart Grid 7 (2015) 216–226.

[8] I. Monedero, F. Biscarri, C. Leon, J. I. Guerrero, J. Biscarri, R. Millan, Detection of frauds and other non-technical losses in a power utility using pearson coefficient, bayesian networks and decision trees, Int. J. Electr. Power Energy Syst. 34 (2012) 90–98.

[9] S. S. S. R. Depuru, L. Wang, V. Devabhaktuni, P. Nelapati, A hybrid neural network model and encoding technique for enhanced classification of energy consumption data, in: 2011 IEEE Power and Energy Society General Meeting, IEEE, 2011, pp. 1–8.

[10] J. A. Meira, P. Glauner, R. State, P. Valtchev, L. Dolberg, F. Bettinger, D. Duarte, Distilling provider-independent data for general detection of non-technical losses, in: 2017 IEEE Power and Energy Conference at Illinois (PECI), IEEE, 2017, pp. 1–5.

[11] N. F. Avila, G. Figueroa, C.-C. Chu, Ntl detection in electric distribution systems using the maximal overlap discrete wavelet-packet transform and random undersampling boosting, IEEE Trans. Power Syst. 33 (2018) 7171–7180.

[12] M. M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, A. Gómez-Expósito, Detection of non-technical losses using smart meter data and supervised learning, IEEE Trans. Smart Grid 10 (2018) 2661–2670.

[13] J. Heaton, An empirical analysis of feature engineering for predictive modeling, in: SoutheastCon 2016, IEEE, 2016, pp. 1–6.

[14] R. Punmiya, S. Choe, Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing, IEEE Trans. Smart Grid 10 (2019) 2326–2329.

[15] Y. Huang, Q. Xu, Electricity theft detection based on stacked sparse denoising autoencoder, Int. J. Electr. Power Energy Syst. 125 (2021) 106448.

[16] G. Fenza, M. Gallo, V. Loia, Drift-aware methodology for anomaly detection in smart grid, IEEE Access 7 (2019) 9645–9657.

[17] Y. He, G. J. Mendis, J. Wei, Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism, IEEE Trans. Smart Grid 8 (2017) 2505–2516.

[18] T. Hu, Q. Guo, X. Shen, H. Sun, R. Wu, H. Xi, Utilizing unlabeled data to detect electricity fraud in ami: A semisupervised deep learning approach, IEEE Trans. Neural Netw. Learn. Syst. 30 (2019) 3287–3299.

[19] R. R. Bhat, R. D. Trevizan, R. Sengupta, X. Li, A. Bretas, Identifying nontechnical power loss via spatial and temporal deep learning, in: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2016, pp. 272–279.

[20] J. Pereira, F. Saraiva, Convolutional neural network applied to detect electricity theft: A comparative study on unbalanced data handling techniques, Int. J. Electr. Power Energy Syst. 131 (2021) 107085.

[21] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, Y. Zhou, Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids, IEEE Transactions on Industrial Informatics 14 (2017) 1606–1615.

[22] M.-M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, A. Gómez-Expósito, Hybrid deep neural networks for detection of non-technical losses in electricity smart meters, IEEE Trans. Power Syst. 35 (2019) 1254–1263.

[23] M. Ismail, M. F. Shaaban, M. Naidu, E. Serpedin, Deep learning detection of electricity theft cyber-attacks in renewable distributed generation, IEEE Trans. Smart Grid 11 (2020) 3428–3437.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2020) 318–327.

[26] B. Li, Y. Liu, X. Wang, Gradient harmonized single-stage detector, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 8577–8584.

[27] S. McLaughlin, B. Holbert, A. Fawaz, R. Berthier, S. Zonouz, A multi-sensor energy theft detection framework for advanced metering infrastructures, IEEE J. Sel. Areas Commun. 31 (2013) 1319–1330.

[28] K. Zheng, Q. Chen, Y. Wang, C. Kang, Q. Xia, A novel combined data-driven approach for electricity theft detection, IEEE Trans. Ind. Inform. 15 (2019) 1809–1819.

[29] S. Woo, J. Park, J.-Y. Lee, I. S. Kweon, Cbam: Convolutional block at-

tention module, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.

[30] Commission for Energy Regulation, Cer smart metering project—electricity customer behaviour trial, 2009–2010 (Irish Soc. Sci. Data Arch., Dublin, Ireland, SN: 0012-00, 2012).