Liu *et al. J Wireless Com Network* (2023) 2023:54
https://doi.org/10.1186/s13638-023-02258-z

EURASIP Journal on Wireless Communications and Networking

REVIEW Open Access

A electricity theft detection method through contrastive learning in smart grid

Zijian Liu[1,2], Weilong Ding[1,2*], Tao Chen[3], Maoxiang Sun[1,2], Hongmin Cai[4*] and Chen Liu[1,2]

*Correspondence: dingweilong@ncut.edu.cn; hmcai@scut.edu.cn

[1] School of Information Science and Technology, North China University of Technology, Beijing 100144, China
[2] Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, Beijing 100144, China
[3] Beijing China-Power Information Technology Co.,Ltd, Beijing 100096, China
[4] School of Computer Science and Engineering, South China University of Technology, Guangzhou 518055, China

Abstract

As an important edge device of power grid, smart meters enable the detection of illegal behaviors such as electricity theft by analyzing large-scale electricity consumption data. Electricity theft poses a major threat to the economy and the security of society. Electricity theft detection (ETD) methods can effectively reduce losses and suppress illegal behaviors. On electricity consumption data from smart meters, ETD methods always train deep learning models. However, these methods are limited to extract different electricity consumption characteristics between independent users, and the pattern differences between users cannot be actively learned. Such difficulty prevents ETD further performance improvement. Therefore, a novel ETD method is proposed, which is the first attempt to apply supervised contrastive learning for electricity theft detection. On the one hand, our method allows the detection model to improve its detection performance by actively comparing users' representation vectors. On the other hand, in order to obtain high-quality augmented views, largest triangle three buckets time series downsampling is adopted innovatively to improve model stability through data augment. Experiments on real-world datasets show that our model outperforms state-of-the-art models.

Keywords: Smart grid, Electricity theft detection, Contrastive learning

1 Introduction

The power grid economy has been affected by electricity theft in somewhat degrees [1], which results in an estimated annual worldwide economic loss of $25 billion [2]. Taking Fujian Province, China as an example, the annual loss due to electricity theft exceeds $15 million [3]. Furthermore, electricity theft has resulted in power surges, excessive loads on the power system, and hidden risks to public safety [4], which greatly impacts the stability of the power system. Consequently, electricity theft detection (ETD) was developed. Traditional ETD relies on manual on-site detection, which is not only cumbersome but also expensive [5]. The development of the Internet of Things has accelerated the realization of smart grids, enabling the deployment of sensors for smart meters that require edge computing [6]. Smart meters can monitor users' electricity consumption data in real-time [7] and analyze this data to provide new solutions for electricity theft detection. The successful application of

artificial intelligence, such as deep learning methods in literature [8, 9], has aroused research interest in ETD to detect electricity theft users.

Existing methods learn different electricity usage characteristics from independent users' electricity data. However, the difference in electricity usage characteristics between different users can only be passively learned by iterative training of the detection model, which limits the improvement of detection accuracy. There are six types of electricity theft [10]. The first type is theft of electricity at a fixed ratio during the electricity consumption process. The second type is theft of electricity at a random time-varying ratio. The third type is theft of all electricity at certain moments, resulting in zero electricity consumption. The fourth and fifth types are related to the average electricity consumption, but the fifth type adds a random impact factor. The last type involves transferring data from high-price periods to low-price periods. We can regard those types as different disturbances to normal electricity usage data. The first type can be viewed as scaling normal electricity data, and its electricity usage pattern is relatively simple and single. However, the last five types involve different degrees of reduction (theft) of normal electricity data in the time dimension, resulting in different electricity usage characteristics. Existing ETD methods mainly learn local electricity usage characteristics from independent users' electricity data in the time dimension and detect electricity theft behavior by comparing them with global time electricity usage characteristics. These methods are suitable for the last five types of electricity theft but not for the first type, which only has a single electricity usage pattern. Furthermore, these methods cannot actively compare the electricity usage characteristics of other user samples to improve the detection effect.

In this paper, we propose a novel ETD method. The improved contrastive learning (CL) framework is adopted here that allows the model to actively compare the electricity consumption characteristics obtained by multiple samples during training to solve the problem of single electricity consumption mode in detecting electricity theft users. The framework consists of three modules: Encoder, Projector, and Classifier. The Encoder performs feature learning on the augmented views of the samples and obtains representation vectors; the Projector maps the high-dimensional representation vectors to a low-dimensional space and improves the learning ability of the Encoder by calculating the contrastive loss of different view representation vectors; the Classifier analyzes and outputs the detection results based on the sample representation vectors output by the Encoder. The contributions of our work include the following three aspects.

(1) To our knowledge, our work is the first attempt to use supervised CL in ETD, where a joint training mode would improve training effect.
(2) Largest-triangle-three-buckets(LTTB) is combined with time series data augmentation to retain local electricity consumption features in augmented views.
(3) Experiments on real-world datasets show that our method outperforms the state-of-the-art methods. This demonstrates the feasibility and efficiency of our work.

The remainder of the paper is organized as follows: Sect. 2 introduces related research work on ETD and CL techniques. Sections 3 and 4, respectively, show our method in

general and in detail. Section 5 describes the experimental design, and Sect. 6 presents the results and analysis. Section 7 concludes the paper and discusses future work.

2 Related work

2.1 ETD methods

Through large-scale data analysis of smart meters, machine learning and deep learning have become widely used in ETD. While machine learning models such as support vector machines(SVM) [11] and K-nearest neighbors(KNN) [12] are favored by domain experts for their fast training speed and interpretability [13, 14]. Such methods struggle to capture complex latent features from electricity consumption data. Consequently, deep learning has become the mainstream approach, as it enables automatic feature extraction without manual feature engineering [8].

Autoencoder (AE) is one deep learning model that can learn feature representations from unlabeled data [15, 16]. A two-step detection method [17] first uses such convolutional autoencoders to extract and identify abnormal characteristics, and then utilizes improved XGboost for potential theft prediction. Convolutional neural network(CNN) is another popular technique for ETD [18, 19]. To address the limitations of capturing long-term dependencies on one-dimensional(1-D) electricity consumption data, Arif et al. [5] adopted a temporal convolutional network (TCN) to train multiple base models and extract electricity consumption characteristics using ensemble learning. In general, the large time span of electricity consumption data leads to a disadvantage in capturing long-term dependencies. For this reason, the Wide and Deep Convolutional Neural Networks(WDCNN) was proposed [4], which converts 1-D time-series data into a two-dimensional(2-D) matrix and uses CNN to extract periodic features and time dependencies. Moreover, Finardi et al. [20] tried to combine self-attention mechanism with CNN to improve detection accuracy. Additionally, Zhu et al. [21] proposed a hybrid approach that uses self-dependency modeling (SDM) to learn second-order representations after obtaining first-order representations from a CNN-based model.

In fact, data imbalance problem always exists among electricity consumption datasets, where electricity theft users belong to the minority class. Such imbalance affects much of the detection model about predictive accuracy. Some studies have attempted rebalancing techniques, such as random oversampling (ROS), the synthetic minority oversampling technique (SMOTE) [22, 23], and others. However, those methods above only improve model's feature extraction ability in the time dimension, and are limited to learn electricity consumption characteristics from independent users' samples. Therefore, their drawback of abnormal detection for electricity consumption characteristics, limits the ability to further improve detection accuracy.

2.2 Contrastive learning

Contrastive learning (CL) is a representation learning method that learns representations from comparisons between different samples, rather than learning signals from independent sample at a time [24]. One of the earliest applications of CL was in self-supervised learning [25] and many self-supervised models proves good performance are based on CL [26–28]. In addition, CL has been widely used in many fields such as video

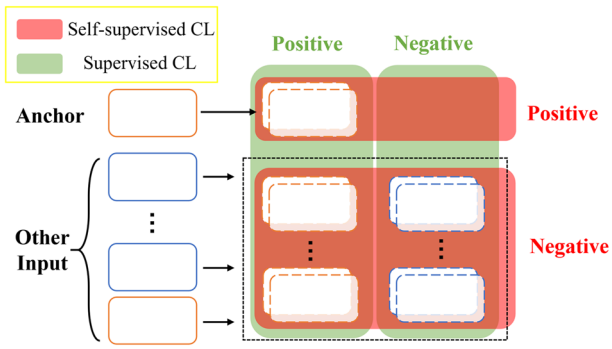

Fig. 1 Self-supervised versus supervised CL: the small rectangles on the left side of the figure represents a batch of input samples, and their color represents the class. The small dashed rectangles on the right are the augmented views generated from the input. The large red boxes indicate the positive and negative sample pairs of self-supervised CL, while the large green ones indicate supervised CL

processing [29], music classification [30], recommendation system [31], and natural language processing(NLP) [32].

The basic idea of CL is to maximize the agreement between representations of "similar" samples(i.e., positive pairs), and minimize those "dissimilar" samples(i.e., negative pairs) [33]. How to define positive and negative sample pairs is the key to distinguish CL types. On the one hand for unlabeled data, the usual way to get positive pairs is to apply data augmentation to generate two augmented views of the same input (i.e., anchor) [34, 35], and negative pairs are formed between anchor and other input's augmented views in the same batch. This method is called self-supervised CL. On the other hand for labeled data, Khosla et al. [36] proposed supervised CL, using label information to contrast all the samples from the same class as positives against the negatives from the remainder of the batch. The positive and negative pairs of self-supervised CL and supervised CL are shown in Fig. 1.

Although CL has been adopted in many domains, there is very few studies exist on ETD. Recently, Fei et al. [37] attempted to use self-supervised CL for ETD. Specifically, on an unbalanced dataset, we selected normal samples with the same number as abnormal samples to construct a balanced dataset. The remaining normal samples are used for pre-training to obtain an encoder, and then the balanced dataset is used to fine-tune the classification. However, like other self-supervised CL methods, such method cannot utilize the prior information of the labels, and its detection performance cannot be further improved [36]. Therefore, all those inspires us to adopt supervised CL instead of self-supervised CL in this paper.

2.3 Data augmentation

As an important part of CL, data augmentation can not only generate positive pairs but also improve the stability of the model [38, 39]. One of the simplest data augmentation methods is to apply dropout on the data, which randomly removes some information to obtain a new augmented view [40]. Similar methods include cutting [41], mix-up [42], erasing [43], etc.

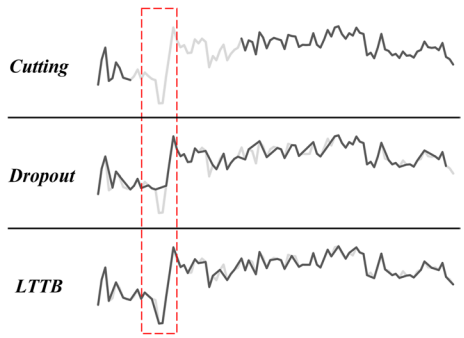

Fig. 2 Comparison of data augmentation methods: the gray line represents the original electricity consumption data, and the black line represents the augmented data with 30% information removed

Although data augmentation can improve robustness, its randomness may lose important information from original data. Taking electricity consumption data as an example, the random disturbance may affect local electricity theft detection. So data augmentation should maintain the integrity of task-related information on the data [44], especially in ETD tasks.

Inspired by [40], we find that time-series down-sampling can also be exploited for data augmentation. Steinarsson et al. [45] proposed the Largest-Triangle-Three-Buckets(LTTB), which is a time-series down-sampling method widely used in industry [46, 47]. The basic idea of LTTB is to remove redundant data from the time series so that the downsampled data can keep the original feature information as much as possible. As shown in Fig. 2, random augmentation methods such as cutting and dropping will lose important local information, while LTTB can well maintain the characteristics of the original data. Therefore, LTTB can be used for time series data augmentation.

3 Problem statement

Suppose the electricity consumption dataset of N users is $\mathcal{X} = (X_1, \ldots, X_i, \ldots, X_N) \in \mathbb{R}^{N \times T}$, where $X_i = (x_1, ..., x_T) \in \mathbb{R}^T$ represents the data of the i-th user, during a time span of T. The ETD task is defined as using the detection model $\mathcal{M}$ to judge whether a user is as an electricity theft among abnormal user. This process can be defined as formula (1), and the detection model is a function that takes the user's electricity consumption data as input and outputs the detection result. Here, $\hat{y}_i$ represents the detection result: 0 is normal users; otherwise 1 represents abnormal users. The optimization objective of the model is to minimize the difference between real result y_i and detection result $\hat{y}_i$, as shown in formula (2).

$$\hat{y}_i = \mathcal{M}(X_i) \tag{1}$$

$$\min_{\mathcal{M}} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2}$$

4 Methods

In this section, we introduce our method in detail which includes two parts: data preparation and contrastive representation learning.

4.1 Data preparation

2-D time series feature In order to extract periodic features on electricity consumption data, we refer to [4] and convert 1-D data X_i into a 2-D matrix $X^{2D} = \{x_{i,j}\} \in \mathbb{R}^{H \times W} (H = T/W)$ with a weekly frequency ($W = 7$) as formula (3). At the same time, a binary mask matrix $M = \{m_{i,j}\} \in \mathbb{R}^{H \times W}$ is added to supplement the missing information of the data, which can improve the effect of model [20, 21]. The two matrices are stacked to get a dual-channel 2-D time series $X^{input} \in \mathbb{R}^{2 \times H \times W}$ as input according to formula (5).

$$X_i \in \mathbb{R}^T \xrightarrow{\text{Reshape}} X^{2D} \in \mathbb{R}^{H \times W} \tag{3}$$

$$m_{i,j} = \begin{cases} 0, & x_{i,j} \text{ is missing} \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

$$X^{input} = [X^{2D}, M] \in \mathbb{R}^{2 \times H \times W} \tag{5}$$

Time series data augmentation The randomness of data augmentation may eliminate important information among data. In order to obtain high-quality augmented views, we propose a time-series data augmentation version of LTTB, which preserves important information as much as possible. Unlike the original LTTB, we do not need to output sampled data, we only need to output a mask matrix $M^{LTTB} \in \mathbb{R}^T$ with retention information. The pseudo-code of LTTB for time-series data augmentation is shown as Algorithm 1.

Assume that n time points need to be retained, then generate n buckets, and put the first and last points into the first and last buckets, respectively. The remaining points are evenly divided into $n - 2$ buckets. Each bucket retains one time point and eliminates the others. Whether the point is retained or not depends on its importance. The technique used to measure such importance of points is called effective area(EA). The EA of a time point is represented by the largest triangle area formed by the current time point and the previous bucket's time point and the next bucket's time point. To reduce computation, the previous bucket selects the retained time point and the next bucket generates a virtual time point with a value equal to the average values of all the time points in that bucket. Thus, one calculation can obtain the EA of that time point, as shown in Fig. 3. The larger the EA is, the more the point fluctuates compared to its neighboring points, and the more time-series trend information it owns. For each bucket, we retain the point with maximum EA. After obtaining M^{LTTB}, we converted it into a 2-D form $M^{Aug1} \in \mathbb{R}^{H \times M}$ in a similar way to formula (3). Finally, use formula (6) to obtain the first augmented view X^{Aug1}, where $\odot$ means element-wise multiplication of matrices.

$$X^{Aug1} = [X^{2D} \odot M^{Aug1}, M \odot M^{Aug1}] \tag{6}$$

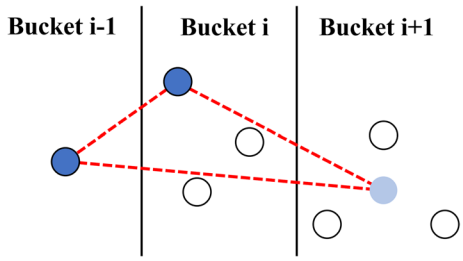

Fig. 3 Calculate effective area: the deep blue point represents the retained data point, and the light blue point represents the virtual data point of the bucket

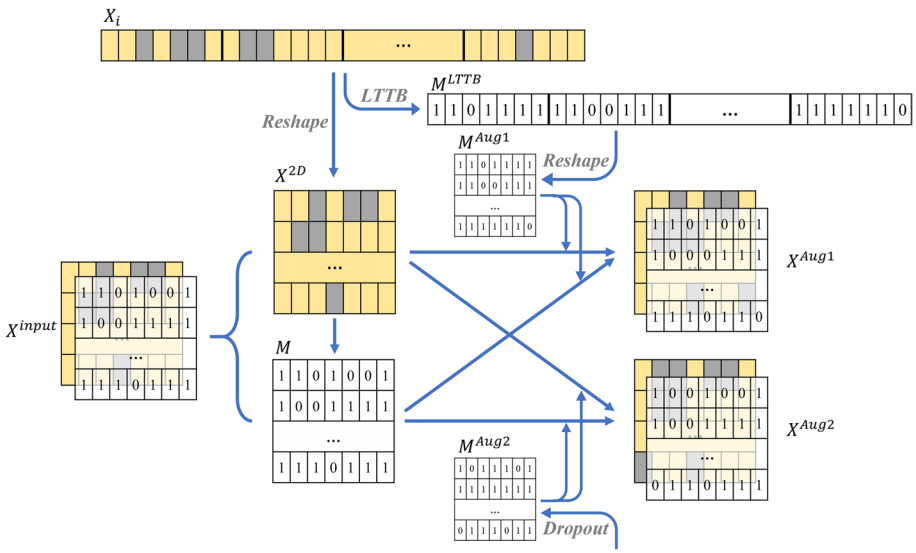

Fig. 4 Data preparation: gray parts indicate missing or discarded data

Algorithm 1 LTTB for time series data augmentation

Input: Time series data $X \in \mathbb{R}^T$; The number of retention is n;
Output: Mask vector $M^{LTTB} \in \mathbb{R}^T$ with retention information;
1: Initialize all values of M^{LTTB} to 0;
2: Initialize n buckets, put the first and last data of X into the first and last buckets, and divide the remaining data into other buckets equally;
3: Retain head and tail data $M^{LTTB}[0] = 1, M^{LTTB}[T-1] = 1$;
4: **for** i : 1 to n-2 **do**
5: Traverse all the points of the i-th bucket, get the point index of the largest EA is j ;
6: Retain data $M^{LTTB}[j] = 1$;
7: **end for**

In order to reduce the mutual information between augmented views [44], we use dropout to get the second augmented view X^{Aug2} according to formula (7). Specifically, M^{Aug2} is obtained by randomly setting part of the position in an all-1 mask matrix to 0 (representing the discarding of data at that position). Then X^{Aug2} is obtained by adjusting X^{2D} and M according to M^{Aug2}. Figure 4 shows all the details of time series data augmentation.

$$X^{Aug2} = [X^{2D} \odot M^{Aug2}, M \odot M^{Aug2}] \tag{7}$$

4.2 Contrastive representation learning

Supervised CL consists of two steps for ETD: electricity consumption representation learning and user behavior classification. Both steps require the Encoder for pattern learning and control the direction of optimization through supervised contrastive loss and classification loss. In the previous subsections, X^{Aug1} and X^{Aug2} were obtained for representation learning, and X_{input} was used for classification.

Encoder Since the input of the Encoder is a dual-channel 2-D time series, we use a Mixed Dilated Convolution(MDConv), including two dilated CNN, to learn the periodicity and temporal dependence of the electricity data with different receptive fields. In order to facilitate the deepening of the network, we add padding operations to prevent data shape changes. At the same time, we normalize and nonlinearly activate the output of MDConv. The formula is (8), where $i \in \{1, .., l\}$ represents the index of MDConv, l represents the number of MDConv layers, F_0 represents the input of the Encoder, $F_i \in \mathbb{R}^{C_i \times H \times M}$ represents the i-th layer output with C_i channels, and f represents the normalization and nonlinear activation function. Here, d_1 and d_2 represent the dilated rate of the corresponding dilated CNN. In order to preserve the characteristics of the 2-D space of the data, we replace the traditional fully connected layer with fully connected convolution(FCConv) [21] in the last layer of Encoder. Specifically, convolution kernels D of size $H \times M$ is used to perform a convolution on F_l to obtain a D-dimensional representation vector $\boldsymbol{r}$ as formula (9).

$$\begin{aligned} F_i &= f(\mathrm{MDConv}_i(F_{i-1})) \\ &= f\left([\mathrm{DilaConv}_i^{d_1}(F_{i-1}), \mathrm{DilaConv}_i^{d_2}(F_{i-1})]\right) \end{aligned} \tag{8}$$

$$\boldsymbol{r} = \mathrm{FCConv}(F_l) \in \mathbb{R}^D \tag{9}$$

Supervised contrastive loss After obtaining the representation vector $\boldsymbol{r}^1, \boldsymbol{r}^2$ of each respective augmented view, the representation similarity between positive and negative pairs can be calculated. In order to reduce the computational complexity, it is necessary to map the high-dimensional representation vector to the low-dimensional space through operator *Projector*, which consists of a layer of fully connection. At the same time, normalization is done for the output again, which makes it possible to use the inner product to calculate the vector distance [36]. The formula is (10).

$$\boldsymbol{z}^{1,2} = \mathrm{Projector}\left(\boldsymbol{r}^{1,2}\right) \in \mathbb{R}^d \tag{10}$$

Assuming a batch of data size is B, each sample can generate two augmented views, resulting in $2B$ representation vectors. The goal of the encoder is to make the representation vectors of the positive pairs as similar as possible, while the negative pairs

as dissimilar as possible. The quality of representation learning can be measured using supervised contrast loss $\mathcal{L}^{Supcon}$, and its formula is (11). Among them, $I = \{1, ..., 2B\}$ represents a index set of 2B vectors, $i \in I$ represents the index of the anchor, and $A(i) = I \backslash \{i\}$ represents other vectors except i-th. In addition, $P(i) = \{p \in A(i) \mid y_p = y_i\}$ represents the set of positive vectors. The symbol $\cdot$ represents vector inner product operator, and the operator to calculate the similarity of the vector. In addition, the hyperparameter τ is used to scale the similarity.

$$\mathcal{L}^{Supcon} = \sum_{i \in I} \mathcal{L}_i^{Supcon} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau\right)} \tag{11}$$

The $\mathcal{L}^{Supcon}$ is composed of $\mathcal{L}_i^{Supcon}$ of each representation vectors. For $\mathcal{L}_i^{Supcon}$, it is necessary to compare all $2B$ data. Specifically, the loss value is an average ratio, where the denominator is the sum of similarities of all sample pairs, and the numerator is the similarity of positive pairs. The better the encoder learns, the larger the proportion of positive similarity and the smaller the loss would be.

Classification loss Similarly, the Encoder performs pattern learning on X^{input} to obtain the electrical representation vector $\boldsymbol{r}^{elec}$ by formula (12). The detection results $\hat{y}$ can be obtained by a classifier composed of a two-layer fully connected network in formula (13). Finally, we use weighted cross-entropy loss $\mathcal{L}^{WCE}$ to measure the detection performance of the classifier, where the weight parameter θ can adjust the model's sensitivity to abnormal users. The $\mathcal{L}^{WCE}$ is calculated as formula (14).

$$\boldsymbol{r}^{elec} = \text{Projector}\left(X^{input}\right) \tag{12}$$

$$\hat{y} = \text{Classifier}\left(\boldsymbol{r}^{elec}\right) \tag{13}$$

$$\mathcal{L}^{WCE} = -\frac{1}{B} \sum_{i=1}^{B} [\theta y_i \log \hat{y}_i + (1-\theta)\left(1 - y_i\right) \log\left(1 - \hat{y}_i\right)] \tag{14}$$

Joint training Traditional supervised CL proposed in [36] is a two-step training method for downstream applications. This method emphasizes representation learning first, followed by the addition of business modules for fine-tuning. Due to the interval of the two-step training, the representation learned by Encoder is not necessarily beneficial to the classification, and the training process is not easy to control. Therefore, we propose a joint training approach that trains representation learning and action classification together rather than separately. The principle of our solution is to combine $\mathcal{L}^{Supcon}$ with $\mathcal{L}^{WCE}$. As formula is (15), the hyperparameter λ can control the importance of the two losses. The framework of the proposed method is depicted in Fig. 5.

$$\mathcal{L}^{Joint} = \mathcal{L}^{WCE} + \lambda \mathcal{L}^{Supcon} \tag{15}$$

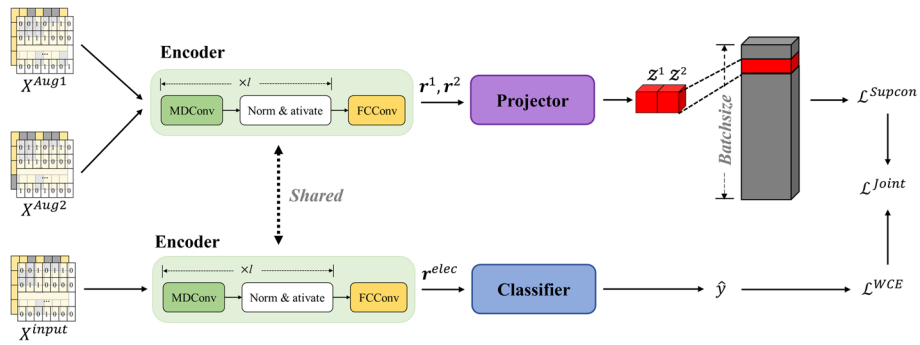

Fig. 5 Architecture of the proposed method

Table 1 The details of SGCC dataset

Item	Value
Total number of users	42,372
Normal/abnormal	38,757/3625
Time span	01/01/2014–10/31/2016
Time step	1 day
Missing rate	25.64%

5 Experiments

5.1 Dataset

The dataset we used comes from the real-world and open-sourced labeled data collected by the State Grid Corporation of China (SGCC).[1] The details of the dataset are shown in Table 1. This dataset contains the electricity consumption data of 42,372 users for a total of 1035 days from 2014 to 2016. Only 8.53% (i.e., 3615 users) are abnormal electricity theft users, and such minority implies an unbalanced dataset. Moreover, the missing rate of the data is about 25.64%, and such poor quality makes original data hardly directly used for model training. Our data preprocessing is necessary accordingly.

5.2 Setting

The samples with single value or empty value are deleted, because these samples do not contribute to model training. Then, the data is sorted in time order for subsequent processing. We have done the preprocessing like [4] to deal with outliers and missing data. In terms of normalization, Z-score standardization is adopted to make the data in standard normal distribution. In order to maintain the class distribution of the original data, we apply stratified sampling to divide the training set and test set. The training ratio (i.e., the percentage of the training set in the total) of the experiment is set to 50%, 60%, 70%, and 80%, respectively. Different experimental data can be regarded as different environments for model training.

On the unbalanced SGCC dataset, the model would appear bias toward normal users. Therefore, oversampling is adopted to sample a batch of balanced data each time, which

[1] https://github.com/henryRDlab/ElectricityTheftDetection/.

Table 2 The optimal values of key parameters

Parameters	Training ratios			
	50%	60%	70%	80%
Learning rate	0.001	0.001	0.001	0.001
Batch size	32	32	64	64
n	700	800	800	800
Discard rate	10%	10%	10%	10%
Kernel size	(3,1)	(3,3)	(3,1)	(3,1)
l	2	2	2	2
d_1	1	1	1	1
d_2	2	2	2	2
D	64	64	64	64
d	16	32	16	16
θ	0.95	0.9	0.95	0.9
λ	0.15	0.1	0.1	0.1
τ	0.07	0.07	0.07	0.07

would makes the number of normal users the same as that of abnormal ones. In details, we sample abnormal user samples or normal user samples according to the odd or even of the data index obtained by dataloader.

Our experimental environment is python 3.8.13, torch 1.12.1, and run on a machine equipped with Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz, Tesla T4 16GB. The parameters of our work include LTTB sampling number n, MDConv layer number l, convolution kernel size, discard rate of data augmentation dropout, etc. Their optimal values, shown in Table 2, are obtained in advance by control variable method.

5.3 Evaluation metrics

To evaluate ETD effects, AUC (the area under curve) and Recall are chosen as metrics. AUC as a widely used metric in ETD tasks [4, 20], represents the area under the receiver operating characteristic(ROC) curve, whose coordinates are false positive rate(FPR) and true positive rate(TPR). The value range is [0.5, 1.0], and the closer the value is to 1, the better the performance of the classification model would be. The AUC calculation formula is (16), where Rank_i represents the rank value of i-th sample, N^{pos} and N^{neg} are the number of positive and negative samples, respectively. AUC measures general detection effect, but for the ETD task, more attention should be paid to that of abnormal users. Accordingly, Recall metric is adopted in addition. Recall indicates how many among abnormal users have been detected correctly. Its value range is [0, 1.0]. The larger the recall value is, the better the anomaly detection effect of the model would be. The calculation formula of Recall is (17), where TP represents the number of abnormal user samples correctly detected, FN represents the number of abnormal users classified as normal, and $TP + FN$ represents the actual total number of abnormal users.

$$\text{AUC} = \frac{\sum_{i \in \text{ positiveClass}} \text{Rank}_i - \frac{N^{pos}(1+N^{pos})}{2}}{N^{pos} \times N^{neg}} \tag{16}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

5.4 Comparative experiment

On the SGCC dataset, four state-of-the-art methods are chosen as baselines for comparison.

- **Wide and Deep CNN(WDCNN)** [4]: It uses the wide module to extract global features from 1-D time series data, and adopts the deep module to extract period information from 2-D time series data.
- **Hybrid Attention(HybridAttn)** [20]: It combines self-attention mechanism with CNN to detect electricity theft, and adds a mask matrix of missing information to solve incomplete data.[2]
- **Graph Convolutional Neural and CNN(GCN-CNN)** [9]: In this hybrid method, CNN is used to learn latent features, and GCN is to obtain time-series dependence and periodic features by building time-series data into a graph structure.
- **Hybrid Order Representation Learning Network(HORLN)** [21]: From the perspective of high-order representation information, it uses the first-order and second-order features on electricity consumption data for ETD.[3]

Note that, WDCNN and GCN-CNN did not provide official source codes. For WDCNN, its reproduction codes are given in the source codes of HORLN, and we restored them accordingly. For GCN-CNN, we directly use the results from the original paper.

At the same time, in order to avoid occasionality, we conducted three independent experiments on each method and then took the average as final experimental results.

5.5 Ablation experiment

To further examine the contribution of LTTB and supervised CL in our method, two variants are designed for comparison.

- **only-LTTB**: It is the variant that supervised CL is removed from our method. Specifically, the Projector is removed, the Encoder learn the electricity consumption features from X^{input}, and the Classifier outputs detection results. The model update can only rely on $\mathcal{L}^{WCE}$, which implies $\lambda = 0$ is set.
- **only-Supcon**: It is the variant that LTTB data augmentation component is removed from the proposed method. We use dropout to generate X_{Aug1} instead of LTTB, and set dropout rate to 30% as the same as LTTB module of the original method to eliminate the interference of other irrelevant factors.

Moreover, the same parameters in Table 2 are used to independently train the variant model three times under four training ratios, and then the average is taken as the result.

[2] https://github.com/neuralmind-ai/electricity-theft-detection-with-self-attention.

[3] https://github.com/GillianZhu/HORLN.

Table 3 The comparison results with the base models

Models	Training ratio 50%		Training ratio 60%		Training ratio 70%		Training ratio 80%	
	AUC	Recall	AUC	Recall	AUC	Recall	AUC	Recall
WDCNN [4]	0.6690	0.5889	0.6797	0.5905	0.6721	0.6359	0.6832	0.6339
HybridAttn [20]	0.7295	0.6096	0.7455	0.6286	0.7440	0.6055	0.7561	0.6284
GCN-CNN [9]	0.7810	–	0.7760	–	0.7870	–	0.787	–
HORLN [21]	0.7844	0.7236	**0.8021**	0.7379	0.8080	0.7583	0.8155	0.7515
Proposed method	**0.7854**	**0.7590**	0.7982	**0.7922**	**0.8101**	**0.7991**	**0.8159**	**0.8136**

Bold data represents the best results

6 Results and discussion

6.1 Comparative experiment analysis

The comparative experimental results are presented in Table 3. Our method achieved the best performance in all experimental settings, except for a slightly lower AUC compared to HORLN when the training ratio of 60%. Furthermore, we have the following findings.

(1) HybridAttn significantly improved the detection compared to WDCNN, because the mask matrix supplemented the missing position information. However, its attention mechanism at different time points cannot analyze the differences between multiple samples, which leads no significant improvement in Recall.
(2) Except for GCN-CNN, the performance of all the others improves proportionally when the training ratio increases. We regard that the learning ability of GCN-CNN has reached its limit, thus increasing the training data does not improve the detection effect.
(3) HORLN achieved the second-best performance by combining first-order and second-order representation information. It proves that higher-order features can indeed enhance the detection performance. However, its higher-order features are learned from independent samples, leading to no further improvement in performance.
(4) Despite using only two simple dilated convolutions as encoders, our proposed method surpassed the state-of-the-art models. Not only it had the best performance, but also it found more abnormal users. The feasibility of supervised CL on ETD is proved then.

Furthermore, we compared the parameter size and the time consumption of training. In Table 4, the results were recorded in an experimental environment where one epoch was trained at a training ratio of 50%. Although the size of the training set and the testing set are equal, the training time is longer than the testing time due to gradients recording and parameter updates during the training process. WDCNN consumes the least time because of the small number of model parameters. Similarly, the method we proposed is sub-optimal. Since the contrastive loss on a batch of data requires time to calculate during the training process, the training time significantly consumes much. In fact, on tens

Table 4 The number of parameters and time consumption (training ratio is 50%)

Models	Parameters (10^6)	Training time (s)	Testing time (s)
WDCNN	**0.896**	**1.74**	**1.03**
HybridAttn	51.062	14.94	4.29
HORLN	3.427	3.81	1.11
Proposed method	1.065	8.58	1.54

Bold data represents the best results

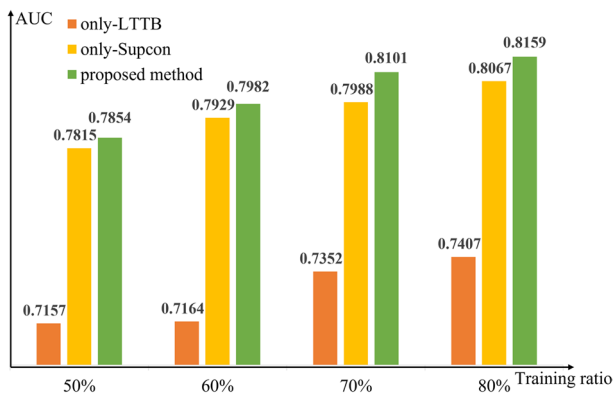

Fig. 6 Ablation experiment

of thousands data, that training time not exceeding 10 s is acceptable enough in practice. It also proves the superiority of our proposed method.

6.2 Ablation experiment analysis

The ablation experiment results are shown in Fig. 6. It can be found that without the support of supervised CL, feature learning solely through Encoder would have much worse results. After all, Encoder only has a simple structure of two layers of dilated convolution. Supervised CL can significantly improve the performance of such a small model.

As for LTTB, unlike random data augmentation methods, this method actively discards data of low-contribution to preserve the original local characteristics of the data as much as possible, thereby improves the model's detection performance. Although the improvement is not obvious, it can be observed that LTTB has advantages in time series data augmentation and the feasibility of downsampling methods as a method for time series data augmentation.

6.3 Visual effect of contrastive learning

In Sect. 2, we introduced that the goal of CL is to make the representation vectors of positive pairs as similar as possible, while negative pairs are as dissimilar as possible. In supervised CL, positive pairs represent the same category and negative pairs represent different categories. In order to achieve supervised CL visualization, we use the trained model to output 8 samples of representation vectors, then perform dot product operation and normalize to [0, 1] as similarity. At the same time, we choose 60% and 80% training ratios for comparison. The heat map is drawn according to the similarity in Fig. 7. It can be seen from the figure that the similarity of representation vectors of

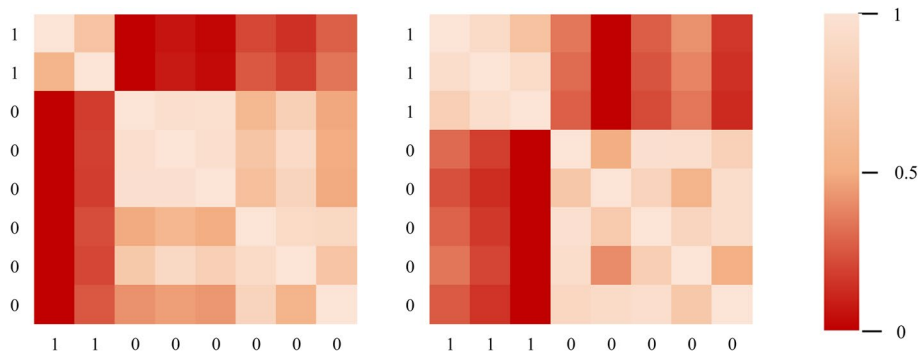

Fig. 7 The visualization of contrastive representation learning: the left is a training ratio of 60%, and on the right is a training ratio of 80%. In the heat map, the numbers on the left and bottom represent categories: 0 indicates normal users, and 1 indicates abnormal users. The lighter the color is, the higher the similarity would be. A similarity of 1 means this is the same vector

different categories is low, while the similarity of the same category is relatively higher. Increasing training samples allows the model to learn more features, thus producing better representation vectors.

7 Conclusion and future work

To actively compare the electricity consumption characteristics between different samples to improve electricity theft detection, we propose a supervised CL-based ETD method. Our method can effectively utilize the prior information of labels so that the similarity of positive pairs is far greater than that of negative pairs, thus obtaining representation vectors with information about the electricity consumption pattern. Furthermore, we innovatively apply LTTB to time series data augmentation and show the feasibility of applying the time series downsampling method to data augmentation. Experiments on real datasets show that our proposed method outperforms state-of-the-art models.

In future, we will further investigate the new potential of CL in ETD and explore other downsampling methods for time series data augmentation.

Abbreviations

ETD	Electricity theft detection
SCL	Supervised contrastive learning
LTTB	Largest triangle three bucket
SVM	Support vector machines
KNN	K-nearest neighbors
AE	Autoencode
CNN	Convolutional neural network
TCN	Temporal convolutional network
1-D	One-dimensional
2-D	Two-dimensional
WDCNN	Wide and deep convolutional neural networks
SDM	Self-dependency modeling
ROS	Oversampling
SMOTE	The synthetic minority oversampling technique
NLP	Natural language processing
EA	The effective are
MDConv	Mixed dilated convolution
DilaConv	Dilated convolution
FCConv	Fully connected convolution
SGCC	State Grid Corporation of China

AUC	Area under curve
ROC	Receiver operating characteristic
FPR	False positive rate
TPR	True positive rate
HybridAtten	Hybrid attention
GCN-CNN	Graph convolutional neural and CNN
HORLN	Hybrid Order Representation Learning Network

Author contributions
ZL designed the method framework and carried out experimental design and implementation. WD participated in and directed the study and helped draft the manuscript. MS et al. helped integrate the manuscript. All authors read and approved the final manuscript.

Funding
This work was supported by the Key-Area Research and Development Program of Guangzhou City (202206030009).

Availability of data and materials
The dataset used in our experiment comes from the real labeled data collected by the State Grid Corporation of China (SGCC) (https://github.com/henryRDlab/ElectricityTheftDetection/). The details of the dataset are shown in Table 1. This dataset is an open source dataset and is widely used in electric theft detection research.

Declarations

Competing interests
Not applicable.

Received: 12 January 2023 Accepted: 1 June 2023
Published online: 23 June 2023

References

1. M. Xing, W. Ding, H. Li, T. Zhang, A power transformer fault prediction method through temporal convolutional network on dissolved gas chromatography data. Secur. Commun. Netw. **2022**, 66 (2022)
2. S.S.S.R. Depuru, L. Wang, V. Devabhaktuni, Electricity theft: overview, issues, prevention and a smart meter based approach to control theft. Energy Policy **39**(2), 1007–1015 (2011)
3. Q. Chen, K. Zheng, C. Kang, F. Huangfu, Detection methods of abnormal electricity consumption behaviors: review and prospect. Autom. Electr. Power Syst. **42**(17), 189–199 (2018)
4. 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 Trans. Ind. Inform. **14**(4), 1606–1615 (2017)
5. A. Arif, T.A. Alghamdi, Z.A. Khan, N. Javaid, Towards efficient energy utilization using big data analytics in smart cities for electricity theft detection. Big Data Res. **27**, 100285 (2022)
6. H. Gao, W. Huang, T. Liu, Y. Yin, Y. Li, Ppo2: location privacy-oriented task offloading to edge computing using reinforcement learning for intelligent autonomous transport systems. IEEE Trans. Intell. Transp. Syst. **6**, 66 (2022)
7. W. Ding, Z. Wang, Y. Xia, K. Ma, An efficient interpolation method through trends prediction in smart power grid. Intell. Mob. Serv. Comput. **66**, 79–92 (2021)
8. M.I. Ibrahem, M. Mahmoud, F. Alsolami, W. Alasmary, A.-G. Abdullah, X. Shen, Electricity theft detection for change-and-transmit advanced metering infrastructure. IEEE Internet Things J. **9**, 25565 (2022)
9. W. Liao, Z. Yang, K. Liu, B. Zhang, X. Chen, R. Song, Electricity theft detection using Euclidean and graph convolutional neural networks. IEEE Trans. Power Syst. **6**, 66 (2022)
10. P. Jokar, N. Arianpoo, V.C. Leung, Electricity theft detection in ami using customers' consumption patterns. IEEE Trans. Smart Grid **7**(1), 216–226 (2015)
11. X. Kong, X. Zhao, C. Liu, Q. Li, D. Dong, Y. Li, Electricity theft detection in low-voltage stations based on similarity measure and dt-ksvm. Int. J. Electr. Power Energy Syst. **125**, 106544 (2021)
12. Y. Himeur, A. Alsalemi, F. Bensaali, A. Amira, Smart power consumption abnormality detection in buildings using micromoments and improved k-nearest neighbors. Int. J. Intell. Syst. **36**(6), 2865–2894 (2021)
13. L. Cui, L. Guo, L. Gao, B. Cai, Y. Qu, Y. Zhou, S. Yu, A covert electricity-theft cyber-attack against machine learning-based detection models. IEEE Trans. Ind. Inform. **6**, 66 (2021)
14. Z. Yan, H. Wen, Electricity theft detection base on extreme gradient boosting in ami. IEEE Trans. Instrum. Meas. **70**, 1–9 (2021)
15. Y. Huang, Q. Xu, Electricity theft detection based on stacked sparse denoising autoencoder. Int. J. Electr. Power Energy Syst. **125**, 106448 (2021)
16. H. Gao, B. Qiu, R.J.D. Barroso, W. Hussain, Y. Xu, X. Wang, Tsmae: a novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder. IEEE Trans. Netw. Sci. Eng. **6**, 66 (2022)
17. X. Cui, S. Liu, Z. Lin, J. Ma, F. Wen, Y. Ding, L. Yang, W. Guo, X. Feng, Two-step electricity theft detection strategy considering economic return based on convolutional autoencoder and improved regression algorithm. IEEE Trans. Power Syst. **37**(3), 2346–2359 (2021)
18. 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**, 107085 (2021)
19. S. Sharma, M. Saraswat, A.K. Dubey, Fake news detection on twitter. Int. J. Web Inf. Syst. **6**, 66 (2022)

20. P. Finardi, I. Campiotti, G. Plensack, R.D. de Souza, R. Nogueira, G. Pinheiro, R. Lotufo, *Electricity theft detection with self-attention*. arXiv preprint arXiv:2002.06219 (2020)
21. Y. Zhu, Y. Zhang, L. Liu, Y. Liu, G. Bin Li, M. Mao, L. Lin, Hybrid-order representation learning for electricity theft detection. IEEE Trans. Ind. Inform. **6**, 66 (2022)
22. S. Li, Y. Han, X. Yao, S. Yingchen, J. Wang, Q. Zhao, Electricity theft detection in power grids with deep learning and random forests. J. Electr. Comput. Eng. **2019**, 66 (2019)
23. M.N. Hasan, R.N. Toma, A.-A. Nahid, M.M. Islam, J.-M. Kim, Electricity theft detection in smart grid systems: a cnn-lstm based approach. Energies **12**(17), 3310 (2019)
24. P.H. Le-Khac, G. Healy, A.F. Smeaton, Contrastive representation learning: a framework and review. IEEE Access **8**, 193907–193934 (2020)
25. J. Li, P. Zhou, C. Xiong, S.C. Hoi, *Prototypical contrastive learning of unsupervised representations*. arXiv preprint arXiv:2005.04966 (2020)
26. K. Hassani, A.H. Khasahmadi, Contrastive multi-view representation learning on graphs, in *International Conference on Machine Learning* (PMLR, 2020), pp. 4116–4126
27. J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, J. Tang, Gcc: graph contrastive coding for graph neural network pre-training, in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020), pp. 1150–1160
28. P. Velickovic, W. Fedus, W.L. Hamilton, P. Liò, Y. Bengio, R.D. Hjelm, Deep graph infomax. ICLR Poster **2**(3), 4 (2019)
29. H. Kuang, Y. Zhu, Z. Zhang, X. Li, J. Tighe, S. Schwertfeger, C. Stachniss, M. Li, Video contrastive learning with global context, in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 3195–3204
30. J. Spijkervet, J.A. Burgoyne, *Contrastive learning of musical representations*. arXiv preprint arXiv:2103.09410 (2021)
31. X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, B. Cui, Contrastive learning for sequential recommendation, in *2022 IEEE 38th International Conference on Data Engineering (ICDE)* (IEEE, 2022), pp. 1259–1273
32. Z. Wu, S. Wang, J. Gu, M. Khabsa, F. Sun, H. Ma, *Clear: Contrastive learning for sentence representation*. arXiv preprint arXiv:2012.15466 (2020)
33. X. Liu, Y. Liang, C. Huang, Y. Zheng, B. Hooi, R. Zimmermann, When do contrastive learning signals help spatio-temporal graph forecasting? in *Proceedings of the 30th International Conference on Advances in Geographic Information Systems* (2022), pp. 1–12
34. Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, Y. Shen, Graph contrastive learning with augmentations. Adv. Neural Inf. Process. Syst. **33**, 5812–5823 (2020)
35. Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Graph contrastive learning with adaptive augmentation, in *Proceedings of the Web Conference 2021* (2021), pp. 2069–2080
36. P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning. Adv. Neural Inf. Process. Syst. **33**, 18661–18673 (2020)
37. K. Fei, Q. Li, C. Zhu, M. Dong, Y. Li, Electricity frauds detection in low-voltage networks with contrastive predictive coding. Int. J. Electr. Power Energy Syst. **137**, 107715 (2022)
38. T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in *International Conference on Machine Learning* (PMLR, 2020), pp. 1597–1607
39. S.-A. Rebuffi, S. Gowal, D.A. Calian, F. Stimberg, O. Wiles, T.A. Mann, Data augmentation can improve robustness. Adv. Inf. Process. Syst. **34**, 29935–29948 (2021)
40. X. Bouthillier, K. Konda, P. Vincent, R. Memisevic, *Dropout as data augmentation*. arXiv preprint arXiv:1506.08700 (2015)
41. S. Yun, D. Han, S.J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 6023–6032
42. H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, *mixup: beyond empirical risk minimization*. arXiv preprint arXiv:1710.09412 (2017)
43. Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34 (2020), pp. 13001–13008
44. Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, P. Isola, What makes for good views for contrastive learning? Adv. Neural Inf. Process. Syst. **33**, 6827–6839 (2020)
45. S. Steinarsson, *Downsampling Time Series for Visual Representation*. PhD thesis (2013)
46. M. Wen, Y. Ma, W. Zhang, Y. Tian, Y. Wang, High-resolution load profile clustering approach based on dynamic largest triangle three buckets and multiscale dynamic warping path under limited warping path length. J. Mod. Power Syst. Clean Energy **6**, 66 (2022)
47. J. Van Der Donckt, J. Van Der Donckt, E. Deprost, S. Van Hoecke, Plotly-resampler: Effective visual analytics for large time series, in *2022 IEEE Visualization and Visual Analytics (VIS)* (IEEE, 2022), pp. 21–25

Publisher's Note