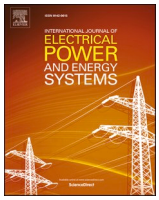
Electrical Power and Energy Systems 165 (2025) 110461

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes



Double-layer stacking optimization for electricity theft detection considering data incompleteness and intra-class imbalance

Leijiao Ge [a,*], Jingjing Li [a], Tianshuo Du [b], Luyang Hou [c]

[a] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072 China
[b] Chengxi Power Supply Branch of State Grid Tianjin Electric Power Company, Tianjin 300190 China
[c] School of Intelligent Manufacturing Engineering and Future Technologies, Fuyao University of Science and Technologies, Fuzhou 350003, China

ARTICLE INFO

Keywords:
Electricity theft detection
Data enhancement
Data interpolation
Stacked integration
Heuristic optimization

ABSTRACT

Electricity theft detection is crucial for reducing non-technical losses in electric power enterprises and ensuring fairness in electric power transactions. However, challenges such as incomplete data, a scarcity of electricity theft samples, and limited performance of detection systems hinder accurate identification of theft. To address these issues, this paper presents a method utilizing a two-stage time-series generative adversarial network (TimeGAN) with an integrated two-layer stacking optimization configuration. The first phase tackles data incompleteness through an enhanced version of TimeGAN, employing embedding and recovery layers to reconstruct incomplete power user data. It introduces an analog denoising training method and supervised information assistance to improve the interpolation accuracy. In the second stage, the method addresses inter- and intra-class imbalances in electricity theft detection by employing the K-shape clustering algorithm to identify unique patterns within the theft data. This enables balanced synthesis of theft samples using these patterns as conditional supervisory terms for TimeGAN. To concurrently optimize the combination of the electricity theft detector (ETD) model and its hyperparameters, an integrated two-layer Stacking optimization framework is developed. This framework incorporates a time-varying binary transfer function and an external repository to enhance the performance of the optimization algorithm. Simulation tests performed on 42,372 actual power consumption records demonstrated that the proposed method achieved, on average, 8.23% higher DR scores, 3.45% higher AUC scores, and 5.60% higher Macro-F1 scores compared to the baseline method.

1. Introduction

Electricity theft by power users is a significant contributor to non-technical losses in the power grid, leading to substantial economic losses for power company and compromising the stable operation of the power system [1,2]. Electricity theft is defined as the use of an electric utility's energy or manipulation of meter readings by an illegal customer to underpay or not pay for electricity without a contract [3].Traditional detecting methods, which rely on labor-intensive inspections, have struggled to meet the demands for cost-effectiveness and efficiency [4]. Consequently, data-driven approaches for detecting electricity theft have gained increasing attention from researchers worldwide [5]. The widespread deployment of advanced metering infrastructure (AMI) has further facilitated this shift, as the vast amounts of electricity consumption data generated enable the implementation of machine learning techniques for identifying fraudulent users [6,7].

The fundamental premise of machine learning methods is to construct various ETD, including support vector machines (SVM), long and short-term memory networks (LSTM) [8,9], FCM clustering [10], decision trees [11], to analyze the electricity data of the customers [12] and try to find anomalous patterns that are highly correlated with electricity theft [13]. However, as electricity consumption data continuous to grow in volume and the complexity and diversity of consumption patterns increase, a single model's recognition capability has struggled to meet the accuracy demands for electricity theft detection [14]. To this end, Ref. [15] has proposed Bagging integrated learning strategy for electricity theft detection. Ref. [16] and [17] applied classifiers based on Boosting integrated learning strategies to electricity theft detection, including XGBoost and LightGBM, and validated their effectiveness with synthetic data. Although the above integrated models have achieved some results in time series classification and prediction problems, they do not have the ability of temporal memory, which makes it difficult to better capture the periodicity or temporal

* Corresponding author.
E-mail address: legendglj99@tju.edu.cn (L. Ge).

https://doi.org/10.1016/j.ijepes.2025.110461
Received 28 September 2024; Received in revised form 28 November 2024; Accepted 3 January 2025
Available online 21 January 2025

Nomenclature

Acronyms

AE	Auto-encoder
AdaBoost	Adaptive Boosting
AMI	Advanced metering infrastructure
AUC	Area under the curve
BPNNs	Back-Propagation Neural Networks
CNNs	Convolutional Neural Networks
CTimeGAN	Conditional time-series generative adversarial network
DR	Detection rate
ERT	Extreme Random Trees
ETD	Electricity theft detector
FCM	Fuzzy-c means
GANs	Generative adversarial networks
IR	Imbalance rate
KNN	K Nearest Neighbors
LI	Linear interpolation
LightGBM	Light Gradient Boosting Machine
LR	Logistic Regression
LSTMs	Long and Short-Term Memory Networks
Macro-F1	Macro-mean F1 scores
NB	Naive Bayes
RF	Random Forest
ROS	Random Over-Sampling
RR	ridge regression
SGCC	State Grid Corporation of China
SMOTE	Synthetic Minority Over-Sampling
SVM	Support Vector Machine
TCNs	Temporal Convolutional Networks
TimeGAN	Time-series generative adversarial network
TV-WOA	Time-varying binary whale optimization algorithm
VAE	Variable Auto-Encoder
WOA	Whale optimization algorithm
XGBoost	Extreme Gradient Boosting Machine

Main variables

$\mathscr{L}_{er}$	Joint loss function
$\mathscr{L}_{gd}$	Unsupervised Adversarial Loss Function
$\mathscr{L}_{eg}$	Additional supervised loss function
$\boldsymbol{M}_t$	The mask matrix at moment t
$\boldsymbol{M}_t^N$	Stochastic multiplicative noise matrix
$\mathscr{L}_D$	Cross-entropy loss
$\mathscr{L}_C$	Modified loss
$\boldsymbol{E}_{1D}/\boldsymbol{E}_{2D}$	Customer 1D/2D electricity consumption data
$Y^{\text{Train}}/Y^{\text{Test}}$	Data training set/test set
$y_{\text{New}}^{\text{Train}}/y_{\text{New}}^{\text{Test}}$	Classification results of base classifiers on training/test datasets
K_V	The number of folds in a K-fold cross validation
$\boldsymbol{S}$	Heterogeneous model selection results
$\boldsymbol{N}^{\text{Aug}}$	Number of data enhancement samples
XM_m^d	The value of the *d-th* decision variable for the *m-th* search agent

correlation of user consumption patterns [18]. Compared with other integration strategies, the Stacking integration strategy offers a novel approach for heterogeneous multi-model fusion [3,19]. Although some studies have highlighted the advantages of Stacking integrated classifiers for identifying electricity theft customers, they often select base and *meta*-classifiers based on model characteristics or a posteriori knowledge without fully considering how different model combinations can affect the optimization of the Stacking integrated learner [20]. This oversight hampers the development of optimal solutions tailored for specific tasks [21]. Furthermore, each heterogeneous model possesses unique hyperparameter variables and ranges, and the reasonable setting of these hyperparameters significantly impacts overall model performance [22,23]. Thus, establishing a classifier for electricity theft detection that optimally configures Stacking integration presents an urgent and essential challenge that must be addressed.

The performance of ETDs is significantly influenced by data quality, necessitating a specific pre-processing procedure to prepare the data for the electricity theft detection task before it is input into the ETD [24]. This pre-processing involves two key steps: 1) Data Padding: Missing data is a common issue due to anomalies in data collection, equipment failures, and data transmission errors. Simple methods like zero-padding or linear interpolation (LI) can introduce inaccuracies that negatively impact the accuracy of electricity theft detection [25]. To address this, Ref. [26] have leveraged the reconstruction capabilities of self-encoder structures to recover samples with missing values. Ref. [27] have employed generative adversarial networks (GANs) to fill in these gaps. However, these approaches are often considered in isolation, and there is potential to enhance data filling by combining the reconstruction abilities of self-encoders with the generative and discriminative strengths of GANs. 2) Data Balancing: The number of abnormal users in the grid is significantly lower than that of normal users, leading to class imbalance that biases ETD training. This bias often results in models that classify most samples as normal to minimize training error. To mitigate this issue, Ref. [28] have utilized random under sampling to remove samples from the majority class, while Ref. [29] have applied random oversampling to enrich the sample set of electricity theft cases. Although these traditional methods can somewhat alleviate class imbalance, they often lead to serious information loss and overfitting, ultimately degrading overall classification performance. Furthermore, these conventional techniques lack robust feature learning and representation capabilities, making it challenging to extract complex and abstract feature representations from the data[30,31]. In contrast, deep learning has emerged as a more effective approach for addressing these challenges[32].

In addition, most scholars primarily focus on the imbalance between the number of electricity theft users and normal users (interclass imbalance), often overlooking the intraclass imbalance that arises from the varying frequency of different electricity theft patterns [33–35]. Traditional data enhancement techniques do not adequately address the classification bias inherent in these electricity theft patterns. As a result, ETDs tend to prioritize the more prevalent electricity theft patterns, which leads to the underrepresentation and potential misclassification of less frequently occurring patterns. This oversight further complicates the effectiveness of ETDs in accurately detecting all types of electricity theft.

To address the aforementioned challenges, this paper presents an electricity theft detection method that employs an integrated and optimized approach of two-stage TimeGAN and two-layer Stacking, as shown in Fig. 1. First, our paper develop a two-stage data interpolation and data enhancement method based on TimeGAN, carefully analyzing the characteristics of electricity theft data. In the initial stage, this paper leverage the temporal reconstruction capability of TimeGAN to recover missing data. A training approach utilizing a binary mask matrix and class denoising self-encoder is introduced to enhance learning from real data. In the second stage, this paper mitigate inter- and intra-class imbalance at the data level, utilizing K-shape clustering results as auxiliary labels to synthesize electricity theft data in a balanced manner. Next, our paper construct a two-layer optimization configuration model using a Stacking integration strategy informed by a heuristic optimization algorithm. The upper layer identifies the optimal configurations for

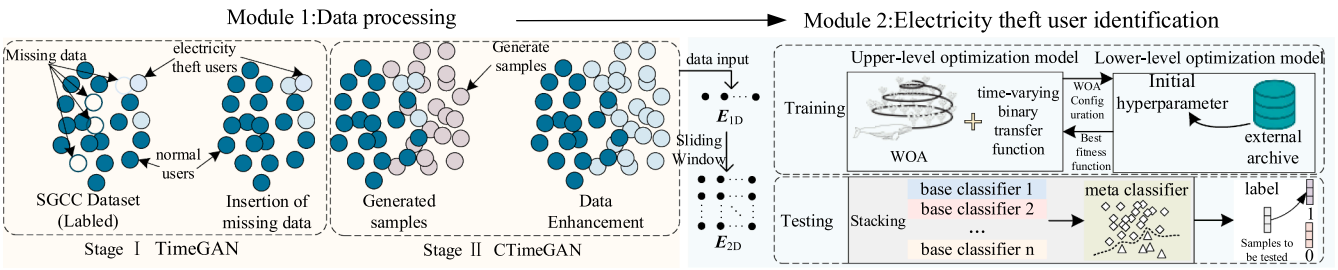


Fig. 1. Double-layer Stacking optimization.

base classifiers, *meta*-classifiers, and the number of data-enhanced samples through the whale optimization algorithm (WOA) that incorporates a time-varying binary transfer function. Meanwhile, the lower layer optimizes the hyperparameters of the heterogeneous model in real time based on the model configuration.

Our contributions are outlined as follows:

i. **Application of the TimeGAN network for Electricity Theft Data Generation**: To the best of our knowledge, our research group is the first to apply the TimeGAN network specifically for generating data related to electricity theft. This innovative approach enables the model to learn both the temporal features and the generative representation of electricity theft data by jointly training the embedding and recovery functions.

ii. **Introduction of the Macro-Average F1 Score as an Objective Function in K-Fold Cross-Validation**: We present the use of the macro-average F1 score as the objective function for K-fold cross-validation. Traditional accuracy and recall metrics often fall short in accurately evaluating model performance, especially with unbalanced sample class distributions. Our approach utilizes the mean of the macro-average F1 score across K-folds, effectively mitigating the impact of unbalanced samples on model evaluation and ensuring a more accurate and reliable assessment process.

iii. **Two-Layer Optimization Configuration Model Based on the WOA**: This paper proposes a two-layer optimization configuration model inspired by the WOA. This model incorporates time-varying binary transfer functions to dynamically adjust the weights of the exploration and exploitation phases during the optimization process. As a result, it enhances the algorithm's search performance and convergence speed, demonstrating superior effectiveness compared to traditional optimization algorithms in solving optimization problems.

The rest of this paper is organized as follows: Section II introduce the characteristics of the power user data, Section III presents the proposed two-stage TimeGAN approach for data incompleteness and class imbalance, Section IV introduced the ETD based on an integrated dual-layer stacking approach, Section V conducted an experimental study and analyze the results, and Section VI conclude the paper.

2. Characterization of power user data

This paper examines the challenges of missing data and class imbalance in electricity theft detection using a real dataset released by the State Grid Corporation of China (SGCC) [36]. This dataset includes electricity consumption records from 42,372 users, of which 3,615 cases involve electricity theft. Figs. 2 and 4 are created based on this data. Fig. 2 illustrates the distribution of the dataset, highlighting that missing data is a significant issue within the actual electricity consumption records, with some users experiencing prolonged periods of continuous data loss.

In the absence of sample labels, cluster analysis serves as an effective method for uncovering potential patterns within the data. However, traditional clustering algorithms that rely on Euclidean distance struggle to accurately measure the shape similarity of time series data. To address this limitation, John Paparrizos et al. introduced the K-shape clustering algorithm, which clusters based on the number of interrelationships rather than solely on distance metrics. This approach enables the identification of sequences that exhibit similar shapes or trends, even when they vary in magnitude or phase. This paper employs the K-shape algorithm for cluster analysis and classifies the electricity consumption curves of electricity theft users into K distinct patterns. The specific process is shown in Fig. 3 and below:

$$SBD(\vec{x},\vec{y}) = 1 - \max_{\omega}\left(\frac{R_{\omega-T}(\vec{x},\vec{y})}{\sqrt{R_0(\vec{x},\vec{x})\cdot R_0(\vec{y},\vec{y})}}\right) \tag{1}$$

$$R_k(\vec{x},\vec{y}) = \begin{cases} \sum_{l=1}^{T-k} x_{l+k}\cdot y_l, k<0 \\ R_{-k}(\vec{y},\vec{x}), k\geqslant 0 \end{cases} \tag{2}$$

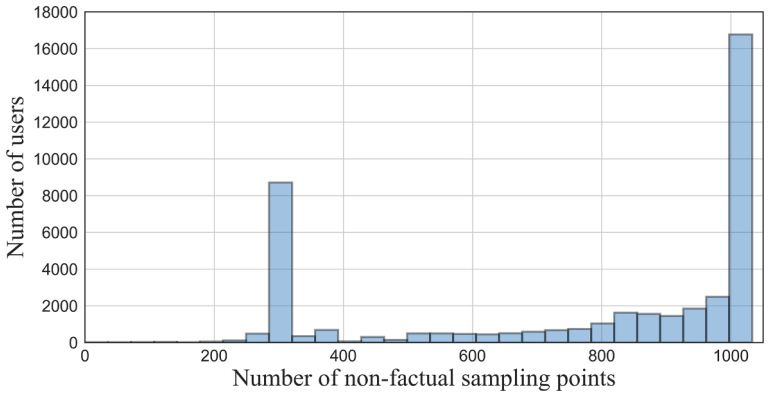


Fig. 2. Distribution of SGCC datasets.

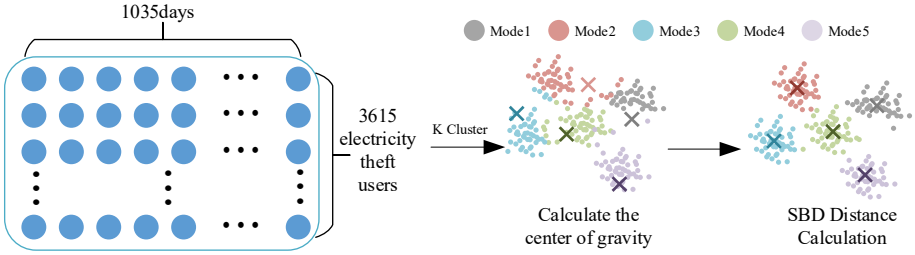


Fig. 3. K-Shape and SBD Process.

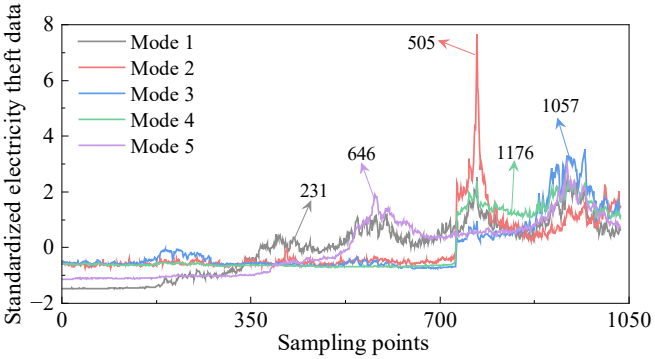


Fig. 4. Centroids and sample sizes for different modes of electricity theft.

where SBD is the shape-based distance; $\vec{x}$ and $\vec{y}$ denote the electricity consumption sequences of two users for similarity measurement; $R_{\omega-T}$ is the inter-correlation sequence; T is the length of time series; ω denotes the length of inter-correlation sequence CC_{ω}, $\omega \in \{1, 2, \cdots, 2T-1\}$; $k = \omega\text{-}T$ is the relative sliding distance of two sequences.

Ref. [37] identifies six common patterns of electricity theft exist; however, one of these patterns involves reversing the order of consumption reports while keeping the electricity consumption remains unchanged. As this specific pattern does not align with the definition of electricity theft used in this paper, we focus solely on the remaining five patterns Consequently, we categorize the 3,615 electricity theft users into five distinct modes based on these identified patterns. The centroids of multiple electricity theft patterns and the corresponding number of users following clustering with the K-shape algorithm are depicted in Fig. 4. The imbalance rate (IR), calculated using equation (3), indicates a significant disparity between normal users and those engaged in electricity theft. Specifically, the IR between the 38,757 normal users and the 3,615 electricity theft users is 10.72. Additionally, the IR among the various electricity theft patterns reaches 5.09, with mode 4 having 1,176 instances compared to mode 1, which has only 231 instances. This pronounced discrepancy can be attributed to the diverse methods and forms of electricity theft, which include illegal grid connections, meter tampering, and data manipulation. Factors such as the variety of electricity theft methods, differences in the difficulty of perpetrating these thefts, biases in data collection, and socio-economic influences contribute to significantly varying sample sizes across subcategories within the electricity theft dataset. Consequently, it is essential to address both inter- and intra-class imbalances within this dataset and to propose effective data enhancement techniques. These measures will ensure that the electricity theft detection model is well-equipped to recognize different types of electricity theft with high accuracy.

$$IR_i = \frac{N_{\max}}{N_i} \tag{3}$$

where $N_{\max}$ represents the number of samples in the majority category, and N_i is the number of categories to be calculated.

3. Two-stage TimeGAN for data incompleteness and class imbalance

TimaGAN combines data reconstruction capability, time series analysis capability, and generative discriminative capability of GAN structure, making it an effective base classifier for time series data filling and generation. This paper proposes a two-stage TimeGAN method to enhance data filling and data augmentation tasks through modifications to the original TimeGAN framework. The first stage focuses on reconstructing corrupted data using embedding and recovery layers, while incorporating a class denoising self-coding training method and a binary mask matrix to facilitate learning. In the second stage, our paper addresses class imbalance for data enhancement by designing a conditional temporal generative adversarial network. This network synthesizes electricity theft data by utilizing the different electricity theft patterns identified through K-shape clustering as conditional supervisory terms.

3.1. TimeGAN principle

The TimeGAN model consists of four distinct components: an embedding function, a recovery function, a sequence generator and a sequence discriminator, as shown in Fig. 5. The effectiveness of TimeGAN is largely attributed to the joint training of its self-coding components (the first two) and adversarial components (the last two). This integrated approach enables TimeGAN to simultaneously learn the temporal features of electricity theft data, generate meaningful representations, and iterate across time.

The embedding function maps the unprocessed electricity theft data into a latent space, allowing the adversarial network to capture the critical temporal dynamics of the data through a reduced-dimensional representation. Meanwhile, the recovery function reconstructs the original raw electricity theft data, ensuring that the temporal properties learned by the embedding function are meaningful and significant. Specifically, we have

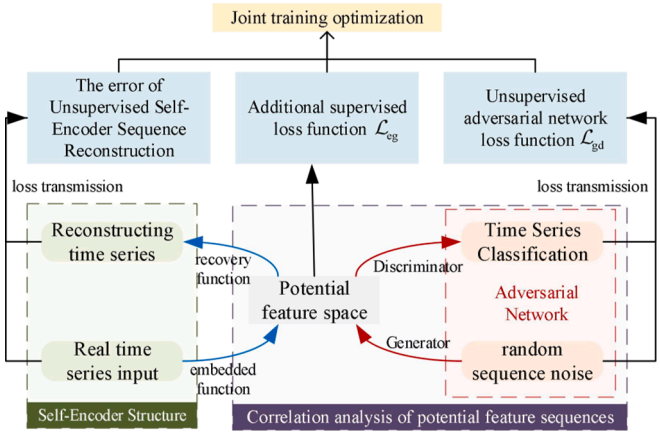


Fig. 5. Schematic diagram of TimeGAN.

$$h_t = e(h_s, h_{t-1}, x_t) \tag{4}$$

$$\tilde{x}_t = r(h_t) \tag{5}$$

where x_t and $\tilde{x}_t$ denote the dynamic feature vector and recovery value at time t, respectively; h_t denotes the dynamic feature vector after dimensionality reduction; h_s denotes the static feature vector after dimensionality reduction. In this paper, all variables are represented as dynamic feature vectors, resulting in static vectors being assigned a value of zero. The functions e and r are implemented using networks of gated recurrent units (GRUs).

In order to enable the embedding and recovery functions to accurately construct the low-dimensional feature space with reconstructing the original feature space, the joint loss function $\mathscr{L}_{er}$ is defined to provide the basis for the update of the embedding network parameter θ_e and the recovery network parameter θ_r as follows:

$$\mathscr{L}_{\mathrm{er}} = \mathbb{E}_{s,x_{1:T}\sim P}[\sum_t \left\| x_t - \tilde{x}_t \right\|_2] \tag{6}$$

where P is the probability distribution of the real data set.

The generator's output consists of the implicit layer variables within the embedding space. The discriminator's role is to differentiate between the synthesized electricity theft data and the input data, which includes both the generator's output and the implicit layer output of the real data. In this configuration, the neural network of the discriminator is replaced by a bidirectional GRU. Both components are trained using a joint loss function $\mathscr{L}_{gd}$:

$$\mathscr{L}_{\mathrm{gd}} = \mathbb{E}_{s,x_{1:T}\sim P}[\sum_t \ln y_t] + \mathbb{E}_{s,x_{1:T}\sim P}[\ln(1-\hat{y}_t)] \tag{7}$$

where y_t represents the classification result of real electricity theft data; $\hat{y}_t$ represents the classification result of synthetic electricity theft data.

The $\mathscr{L}_{gd}$ alone is insufficient to encourage the generator to accurately capture the gradual dynamic conditional distribution in electricity theft data [25]. To address this issue, an additional monitored loss $\mathscr{L}_{eg}$ is introduced allowing the generator to be trained using both the original loss function $\mathscr{L}_{gd}$ and the new monitored loss $\mathscr{L}_{eg}$ in an alternating manner:

$$\mathscr{L}_{\mathrm{eg}} = \mathbb{E}_{s,x_{1:T}\sim P}[\sum_t \|h_t - G(h_s, h_t, z_t)\|_2] \tag{8}$$

where G represents the generator function; z_t is a dynamic random vector obeying a Wiener process.

3.2. Stage 1 of TimeGAN: Data filling

The reconstruction functions of TimeGAN's embedding and recovery layers facilitate the recovery of corrupted data. Unlike traditional methods that rely solely on LI, TimeGAN emphasizes the analysis of temporal properties within the data. Building on this foundation, this paper applies TimeGAN to time-series data interpolation. As highlighted in [25], there may be inherent correlations between different features at the same time step, and utilizing these correlated features as simultaneous inputs can enhance interpolation accuracy. This paper adopts this concept by grouping users based on similarities in their electricity usage habits using K-shape clustering [38]. These clusters serve as reconstructed inputs to TimeGAN, providing auxiliary information on electricity usage patterns and ensuring that the interpolated data aligns more closely with actual values. The overall process, including data input, reconstruction, generation and discrimination, loss calculation and optimization, is illustrated in Fig. 6. The input time series data matrix is displayed in the upper left corner, where the orange '×' indicates the missing data points. Initially, the input data is processed through TimeGAN for reconstruction, generating the reconstructed data $\tilde{x}_t$ using a random noise matrix M^V. The error between the reconstructed data $\tilde{x}_t$ and the original non-missing value data is calculated. Subsequently, the random noise matrix M^V is combined with the reconstructed data to compute the reconstruction error of the randomly corrupted data. The sum of these two errors constitutes the joint loss, guiding the tuning of network parameters in the embedding and recovery layers.

On the right side of the figure, the reconstructed data is further processed through the TimeGAN generator to produce corrected data, which is then evaluated by the TimeGAN discriminator to compute the mask estimate $\hat{M}_t$. The difference between the corrected data and the original data is assessed using cross-entropy losses and correction losses. These losses jointly control the parameter tuning of both the generator

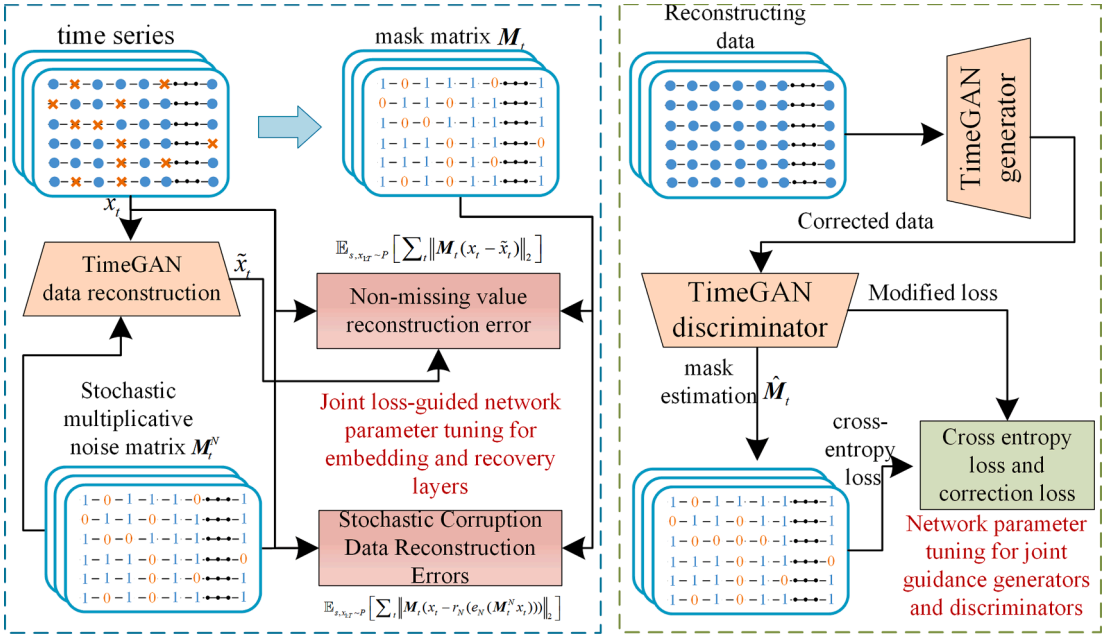


Fig. 6. Schematic diagram of TimeGAN data interpolation.

and the discriminator, ensuring effective training and improved data recovery.

Specifically, our paper characterize the data state by introducing a mask matrix $\boldsymbol{M}$ at moment t, which is defined as:

$$\boldsymbol{M}_t = [m_t^1, m_t^2, \cdots, m_t^u], u \in (1, N_U) \tag{9}$$

where N_U represents the number of input users at the time of data interpolation, and $m_t^u = 0$ indicates that the data point of the u-th user at moment t is missing.

To effectively capture temporal features that align with the true distribution, this paper designs a joint training function for the embedding and recovery layers aimed at reconstructing errors associated with non-missing values:

$$\mathscr{L}_{er}^{I} = \mathbb{E}_{s,x_{1:T}\sim P}\left[\sum_t \left\|\boldsymbol{M}_t(x_t - \widetilde{x}_t)\right\|_2\right] \tag{10}$$

Filling in missing data without supervisory information can lead to model overfitting on the existing real data. To address this issue, this paper introduces a similar denoising self-encoder training approach. In this method, multiplicative noise M_t^N is randomly added to the original signal, effectively zeroing out portions of the data. The original data is subsequently restored through feature mapping of the randomly corrupted data, enabling the encoder to learn a more robust and generalized data recovery process even in the absence of complete data. The joint loss function, which incorporates the supervised loss, is defined as follows:

$$\mathscr{L}_{er}^{N} = \mathscr{L}_{er}^{I} + \mathscr{L}_{er}^{S} \tag{11}$$

$$\mathscr{L}_{er}^{S} = \mathbb{E}_{s,x_{1:T}\sim P}\left[\sum_t \left\|\boldsymbol{M}_t(x_t - r_N(e_N(\boldsymbol{M}_t^N x_t)))\right\|_2\right] \tag{12}$$

where r_N and e_N represent the recovery function and embedding function during the TimeGAN data interpolation process, respectively.

The generator takes the reconstructed data $\widetilde{x}_t$ and the real electricity consumption data x_t^S as input for the same time period, focusing on similar user groups corresponding to the missing values. This process allows the generator to refine and correct the reconstructed data based on real consumption patterns.

$$\widehat{x}_t^G = \widetilde{x}_t + G(\widetilde{x}_t, h_t, x_t^S, \boldsymbol{M}_t^N) \tag{13}$$

The power usage data after performing data interpolation is obtained by:

$$\widehat{x}_t = M_t x_t + (1 - M_t)\widehat{x}^G \tag{14}$$

The discriminator's role is to identify which values in the estimated complete time series are synthetic. In other words, it attempts to estimate the original mask matrix $\boldsymbol{M}$. To evaluate the performance of the discriminator, the loss between its estimates $\widehat{\boldsymbol{M}}$ and $\boldsymbol{M}$ is measured using binary cross-entropy:

$$\mathscr{L}_D = BCE(\widehat{\boldsymbol{M}}, \boldsymbol{M}) = \sum_{t=1}^{N_T}\sum_{u=1}^{N_U}[m_t^u \log \widehat{m}_t^u + (1 - m_t^u)\log(1 - \widehat{m}_t^u)] \tag{15}$$

where N_T represents the length of the time series.

To tailor the generator for the data interpolation task, this paper designs a new loss function comprising two main components: the discriminator loss $\mathscr{L}_{\mathrm{D}}$ and the set correction loss $\mathscr{L}_C$:

$$\mathscr{L}_G = -\mathscr{L}_D + \lambda\mathscr{L}_C = -\mathscr{L}_D + \lambda\sum_t \left\|(1 - M_t^N)(\widehat{x}_t^G - x_t)\right\|_2 \tag{16}$$

3.3. Stage II of TimeGAN: Data enhancement

In this stage, the K-shape clustering algorithm is employed to identify potential electricity theft patterns among users in the training set. To enable TimeGAN to generate electricity theft sequences that adhere to these specified patterns, this paper designs a conditional time-series generative adversarial network (CTimeGAN) model. The clustering results y are input into TimeGAN as a conditional supervised term, alongside a dynamic random vector z. This setup allows TimeGAN to generate electricity theft sequences that reflect the specified patterns, thereby enhancing the representation of each electricity theft pattern with balanced data. The joint loss function of CTimeGAN generator and discriminator is defined as follows:

$$\mathscr{L}_{gd}^{C} = \mathbb{E}_{s,x_{1:T}\sim P}[\sum_t \ln y_t] + \mathbb{E}_{s,x_{1:T}\sim P}[\sum_t \ln(1 - D(G(z_t, y^{\mathrm{in}}), y^{\mathrm{in}}))] \tag{17}$$

The supervised loss function is defined as:

$$\mathscr{L}_{eg}^{C} = \mathbb{E}_{s,x_{1:T}\sim P}[\sum_t \left\|h_t - G(h_t, z_t, y^{\mathrm{in}})\right\|_2] \tag{18}$$

The global optimization loss function is utilized to guide the network parameter tuning of the embedding layer, recovery layer, generator, and discriminator during the training process. The global optimization loss function is defined as:

$$\begin{aligned}
&\widehat{\theta}_e, \widehat{\theta}_r = \underset{\theta_e,\theta_r}{\operatorname{argmin}}\left(\lambda\mathscr{L}_{eg}^{C}\left(y^{in}\right) + \mathscr{L}_{er}\left(x_t, \widehat{x}_t\right)\right), \\
&\widehat{\theta}_g, \widehat{\theta_d} = \underset{\theta_g}{\operatorname{argmin}}\left(\eta\mathscr{L}_{eg}^{C}\left(y^{in}\right) + \operatorname{argmax}\underset{\theta_d}{\mathscr{L}_{\mathrm{gd}}}\left(\mathrm{y_t}, \mathrm{y^{in}}\right)\right)
\end{aligned} \tag{19}$$

where θ_e, θ_r, θ_g, and θ_d represent the network parameters of the embedding layer, the recovery layer, the generator, and the discriminator, respectively; y and k are the weight coefficients, which are set to 1 and 10, respectively, based on prior research in [39].

Through the aforementioned process, CTimeGAN generates synthetic data that mirrors the statistical features of actual electricity theft behavior, making it valuable for training electricity theft detection algorithms and enhancing the security of the electricity system. However, Ref. [35] highlights that in the context of inter-class imbalance in electricity theft detection, synthesizing a small number of samples to align with the majority class is not optimal. Moreover, generating too many synthetic samples can adversely affect ETD performance. Consequently, determining the optimal number of synthesized samples for each sub-pattern presents a significant challenge for addressing intra-class imbalance, which directly influences the recognition capability of ETD. Conducting an exhaustive search for the optimal number of synthesized samples is often impractical, as it involves evaluating all possible solutions. Instead, heuristic algorithms leverage heuristic information to streamline the search process, resulting in a smaller search space and faster convergence. Therefore, this paper will optimize the number of sub-patterns synthesized using a heuristic optimization algorithm. The specific objective function and constraints will be introduced in the subsequent two-layer optimization configuration model.

4. ETD based on an integrated dual-layer Stacking approach

While the aforementioned techniques enhance the quality and diversity of input samples for the classifier, the accuracy of the classification ultimately hinges on the classifier's performance. To address the challenge of insufficient classification capability of a single model in the context of big data, this paper employs a heterogeneous model integration classifier based on a Stacking strategy for the classification task of power users. This approach also considers the optimal configuration of the base classifiers, meta classifier, and hyper-parameters to improve classification effectiveness.

4.1. Feature data construction

Influenced by lifestyle and work habits, most users exhibit a weekly cyclical pattern in their power consumption. However, electricity theft disrupts this cyclical behavior, often resulting in a gradual decrease in overall electricity consumption [40]. Consequently, power theft samples exhibit complex distributional characteristics, manifesting as an irregular mixed distribution of normal and theft patterns. To aid the model in effectively capturing the transition from normal electricity consumption to electricity theft behavior, this paper transforms the one-dimensional electricity consumption data into two-dimensional data using a sliding window approach. Assuming that the one-dimensional electricity consumption data for a user, containing one N_E sampling point, is represented as follows:

$$\boldsymbol{E}_{1D} = [e_1, e_2, \cdots, e_{N_E}] \tag{20}$$

Then, our paper constructs a two-dimensional dataset based on the sliding window interval L_g as follows:

$$\boldsymbol{E}_{2D} = [\boldsymbol{E}_1, \cdots, \boldsymbol{E}_t \cdots, \boldsymbol{E}_{N_W}] \tag{21}$$

$$\boldsymbol{E}_t = [e_{t+L_g}, e_{t+L_g+1}, \cdots, e_{t+L_g+L}] \tag{22}$$

where E_t represents the electricity usage data of the t-th sliding window; L denotes the length of the sliding window; L_g is the sliding window interval; and N_W represents the number of sliding windows. Another advantage of this approach is that each sliding window can be treated as a distinct feature, facilitating data augmentation and enabling classification models to capture key information at different time steps [29].

4.2. Stacking integration principles

Different base classifier models exhibit distinct error characteristics, and the purpose of Stacking model fusion is to mitigate the impact of a single base classifier's errors on the overall classification performance of the ensemble model. This approach aims to enhance the effectiveness of the multi-model fusion classification system. The core idea of Stacking is to construct a two-layers classification framework: the first layer consists of base classifiers, while the second layer features *meta*-classifiers. In this setup, the base classifiers analyze the electricity consumption data and generate classification results, which serve as inputs for the *meta*-classifiers. The *meta*-classifier learns the relationship between the outputs of the base classifiers and the actual classification results for electricity users. The specific principles of this process are illustrated in Fig. 7.

1) Base classifier training process

First, the training data Y^{Train} is homogenized into K^S folds $Y_\kappa^{\text{Train}} = \{(x_\kappa^{\text{Train}}, y_\kappa^{\text{Train}}) | \kappa \in (1, 2, \cdots, K^S)\}$, where x_κ^{Train} and y_κ^{Train} denote the input features and category labels of the κth fold sample, respectively. The data used to train the base classifier is referred to as the base training set, with each fold serving as the base test set. Assuming that the selected N^B base classifier models are chosen through a heuristic approach as $[BC_1, BC_2, \cdots, BC_{N^B}]$. Each base classifier is trained on $K^S - 1$ folds of data to predict the remaining training data, with each fold corresponding to a separate predictor model. For instance, if BC_1 repeats the above process for K^S times, it derives K^S sub-models of base classifier 1 $[BC_{1,1}, BC_{1,2}, \cdots, BC_{1,K^S}]$. The classification results from these models are concatenated to create the training set classification result $y_{BC_1}^{\text{Train}}$. This process is similar applied to the other base classifiers. The classification results from each base classifier on the training data set are combined as follows:

$$y_{\text{New}}^{\text{Train}} = [y_{BC_1}^{\text{Train}}, y_{BC_2}^{\text{Train}}, \cdots, y_{BC_{N^B}}^{\text{Train}}]^\top \tag{23}$$

During testing, each of the K^S submodels of the base classifier classifies the test set Y^{Train}, and the K^S classifications are averaged to produce the final classification result of that base classifier $Y_{BC_1}^{\text{Test}}$. This process is repeated for each of the base classifiers to generate the test inputs for the *meta*-classifier as follows:

$$y_{\text{New}}^{\text{Test}} = [y_{BC_1}^{\text{Test}}, y_{BC_2}^{\text{Test}}, \cdots, y_{BC_{N^B}}^{\text{Test}}]^\top \tag{24}$$

2) Meta classifier training process

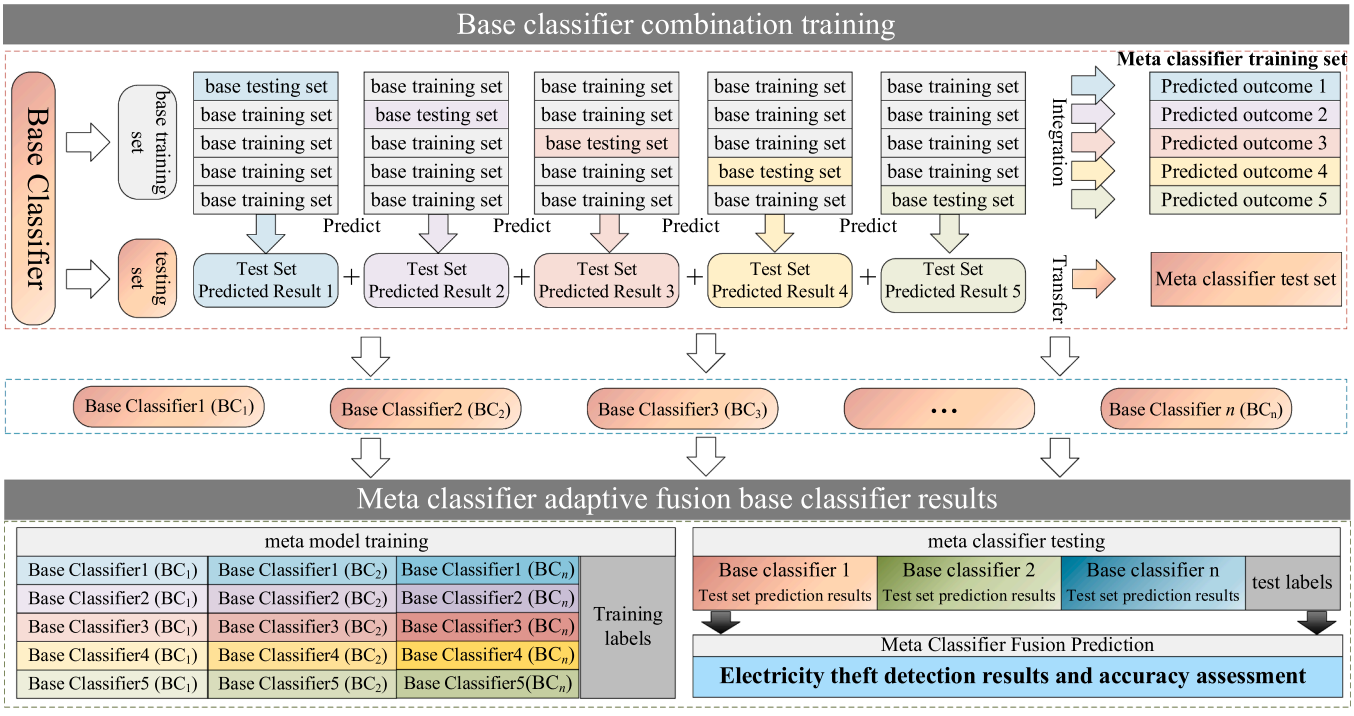


Fig. 7. Schematic diagram of Stacking ensemble classifier.

The second layer of Stacking consists of a *meta*-classifier selected through a heuristic algorithm to integrate the knowledge provided by the base classifiers. During the supervised training phase, the *meta*-classifier receives the classification result $y_{\text{New}}^{\text{Train}}$ from each base classifier on the training dataset, with the output being the category label of the electricity user. In the testing phase, a new test dataset $y_{\text{New}}^{\text{Train}}$ obtained during the training of the base predictors is used to determine the categories of power users, and the model's performance is evaluated using relevant evaluation metrics.

4.3. Two-tier stacking optimization model

To address the simultaneous dynamic optimization challenges associated with heterogeneous model combinations, the number of sample synthesis, and model hyperparameters, our paper constructs a two-layer Stacking optimization configuration model in this section. The specific settings for the objective function and constraints are outlined as follows:

1) Problem Description

Since Stacking is a heterogeneous model integration strategy, various combinations of classifiers may yield better performance on different datasets or problems. Therefore, this paper emphasizes the need for a preference-based approach rather than relying on a conventional configuration. This paper transforms the selection of a heterogeneous model into an optimization problem with decision variable $\boldsymbol{S} = [s_1, s_2, \cdots, s_{N_S}, s_{N_S+1}]$, where $s_1 \sim s_{N_S}$ is a binary variable, 1 represents that the model is selected as the base classifier, and s_{N_S+1} is an integer from 1 to N_s, representing the selection of *meta*-classifiers.

Another concern is that selecting hyperparameters is often not guided by a priori knowledge, making it challenging to make scientifically informed choices based solely on manual experience. Most researchers typically utilize heuristic algorithms to optimize hyperparameters. However, during the Stacking configuration optimization process, the number and types of hyperparameters across different heterogeneous model combinations can vary, complicating the simultaneous optimization of both heterogeneous model combinations and hyperparameters. To address this issue, this paper proposes a two-layer heuristic optimization configuration method in this paper, which considers the optimal amount of intra-class imbalance data enhancement discussed in Section 2.3. The upper layer of this optimization method defines the configurations for the base classifier, meta classifier, and the number of samples synthesized for each electricity theft mode. In contrast, the lower layer provides feedback on the corresponding hyperparameter optimization results based on the model configurations specified by the upper layer. The objective function and constraints of the optimization model are defined as follows:

i. Objective function

The heuristic requires a metric that comprehensively represents the classification ability of the model to serve as an objective function guiding the optimization process. K-fold cross-validation is employed, allowing each fold to be utilized for both training and validation. This approach effectively mitigates issues related to model overfitting and inaccurate evaluations stemming from insufficient validation data. However, traditional metrics such as accuracy and recall can be misleading in the presence of class imbalance, failing to accurately reflect the model's true classification performance. For instance, if the model labels all 42,372 users as normal users, it could still attain an accuracy of 91.47 %. To address this limitation, this paper adopts the average macro-mean F1 scores (Macro-F1) from K-fold cross-validation as the objective function to minimize the influence of sample size on model evaluation:

$$\min f = -\frac{1}{2K_V}\sum_{i=1}^{K_V}\sum_{c=0}^{1}\frac{precision_i^c recall_i^c}{precision_i^c + recall_i^c} \tag{25}$$

where K_v is the number of folds for K-fold cross-validation, which is set to 5 in this paper; $precision_i^c$ and $recall_i^c$ represent the accuracy and recall of category c when the i-th fold is used as the validation set, respectively.

ii. Constraints

The decision variables of the upper-level optimization model include the selection result of the heterogeneous model $\boldsymbol{S} = [s_1, \cdots, s_{N_S}, s_{N_S+1}]$ and the number of data enhancement samples $\boldsymbol{N}^{\text{Aug}} = [N_1^{\text{Aug}}, \cdots, N_K^{\text{Aug}}]$ with the constraints:

$$S_{N_S+1} \leqslant N_S \tag{26}$$

$$N_k - N_k^{\text{Ori}} \geqslant 0 \tag{27}$$

$$N_k^{\max} \leqslant 2N_k^{\min} \tag{28}$$

where N_k represents the sample number of the k-th electricity theft pattern after data synthesis;N_k^{Aug} is the synthesized number of the k-th electricity theft pattern; N_k^{Ori} represents the original sample number of the k-th electricity theft pattern; $N_k^{\max}$ and $N_k^{\min}$ represent the electricity theft patterns that have the highest and the lowest number of samples after data augmentation, respectively. The objective is to prevent significant disparities in the number of synthesized samples and to ensure that samples from each pattern are synthesized in a balanced manner.

The lower-level optimization model decision variable is the hyperparameter setting $\boldsymbol{P} = [\boldsymbol{p}_1, \cdots, \boldsymbol{p}_i, \cdots, \boldsymbol{p}_{N_S+1}]$ with constraints:

$$pmax_i^j \leqslant p_i^j \leqslant pmin_i^j \tag{29}$$

where $\boldsymbol{p}_i$ represents the hyperparameter combination of the i-th model; $\boldsymbol{p}_{N_S+1}$ represents the hyperparameter combination of the meta classifier; p_i^j represents the j-th hyperparameter of the i-th model; and $pmax_i^j$ and $pmin_i^j$ represent the upper and lower limits of the j-th hyperparameter of the i-th model, respectively.

2) Two-layer optimization model

The WOA demonstrates superior search performance compared to many traditional optimization algorithms, such as particle swarm optimization, genetic algorithm, differential evolution, ray optimization, in addressing optimization problems [41]. Therefore, this paper constructs a two-layer optimization configuration model based on WOA, as illustrated in Fig. 8. The fundamental principle is that the optimization process of the lower-layer WOA is guided by the configuration established in the upper-layer WOA. Additionally, the adaptations of the lower-layer heuristic are fed back to the upper-layer, facilitating a closed-loop iteration.

i. Upper optimization model: Combining WOA with time-varying binary transfer function

In the WOA, whale behavior is categorized into two phases: the exploitation phase and the exploration phase. During the exploitation phase, the individual position is first updated by encircling the prey and approaching the best search agent, using the following position calculation formula:

$$D = \left|C \cdot X_t^B - X_t\right| \tag{30}$$

$$X_{t+1} = X_t^B - A \cdot D \tag{31}$$

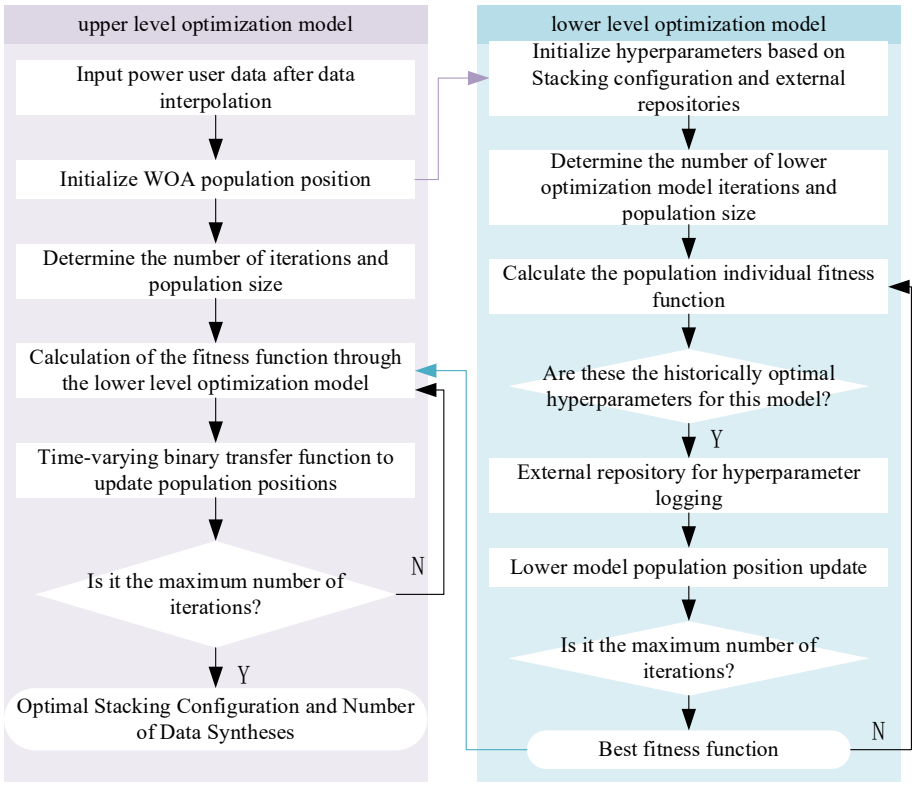


Fig. 8. Flow chart of two-layer optimization configuration model.

where t denotes the current number of iterations; D is the distance between the current whale individual and the optimal solution; X_t^B represents the optimal position under the current number of iterations; X_t denotes the current position of the whale individual; A and C are the position update coefficients, respectively.

Since the whale swims along a spiral shaped path while encircling its prey, a probability of 50 % is established to randomly select between the two behaviors, thereby facilitating the update of the whale's position.

$$X_{t+1} = \begin{cases} X_t^B - A \cdot D\rho \ < \ 0.5 \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + X_t^B \rho \geqslant 0.5 \end{cases} \tag{32}$$

It is noteworthy that the upper model optimization process involves binary variables, and the traditional continuous numerical optimization may lose their effectiveness in this context. Additionally, conventional T-type and S-type binary transfer functions, which directly map the search agent positions to the binary space, do not account for the dynamic characteristics of the optimization algorithm's search process. In the early stages of operation, optimization algorithms should prioritize exploration to avoid converging to local optima, while in the later stages, they need to emphasize exploitation to enhance the quality of the solution. Considering these factors, this paper introduces a time-varying binary transfer function within the WOA framework:

$$TV(XM, \varphi) = \frac{1}{1 + e^{-\frac{XM}{\varphi}}} \tag{33}$$

$$\varphi = \varphi_{\max} - t \cdot \left(\frac{\varphi_{\max} - \varphi_{\min}}{t_{\max}}\right) \tag{34}$$

where XM represents the position of individuals of the algorithm; t is the current number of iterations; t_{max} is the maximum number of iterations; φ is the control parameter; φ_{max} and φ_{min} are the upper and lower limit of the control parameter, respectively, which are set to 4 and 0.05 in this paper.

The location of the upper-level decision variables regarding the Stacking configuration can be transformed as follows:

$$XM_m^d(t+1) = \begin{cases} 1 rand \ < \ \mathrm{TV}(\mathrm{XM}_m^d(t), \varphi) \\ 0 rand \geqslant \mathrm{TV}(\mathrm{XM}_m^d(t), \varphi) \end{cases} \tag{35}$$

where XM_m^d represent the value of the d-th decision variable of the m-th agent, and *rand* denotes a random number between 0 and 1.

ii. Lower-level optimization model: External repository-assisted WOA

As illustrated in the optimization configuration process in Fig. 8, the upper optimization framework involves decision variables related to the selection of heterogeneous models and the number of synthesized samples. Assuming that a total of 11 models are involved in the optimization, the decision variable can be characterized as $\boldsymbol{X} = [1,0,1,0,0,1,1,0,0,0,0,5|N_k^{\mathrm{Aug}}]$, implying that the number of samples augmented with data from each electricity theft pattern is denoted as N. In this configuration, classification models 1, 3, 6, and 7 are selected as base classifiers, while classification model 5 is designated as a meta classifier. The hyperparameters of models 1, 3, 5, 6, and 7 will serve as decision variables for optimization within the lower layer framework. The optimal classification results obtained will be fed back to the upper layer as fitness function values, thereby continuing the optimization process. To facilitate the heuristic in quickly identifying optimal hyperparameters, this paper constructs an external archive that records the hyperparameters corresponding to the historically optimal objective function of each model when utilized as either a base classifier or meta classifier. During the lower-level model optimization, the initial hyperparameter values for each classification model will be sourced from this external archive rather than being generated randomly.

5. Experiment results and analysis

The simulation software system operates on Ubuntu 18.04, with all algorithms implemented using PyTorch 1.9.0 framework, with Python 3.8 and Cuda 11.1 neural network framework. Training is conducted on an RTX 3090 GPU (24 GB), paired with a 12 vCPU Intel® Xeon® Platinum 8375C CPU operating at 2.90 GHz. To validate the effectiveness and advantages of the proposed model, four sets of experiments have been designed, simulating and validating the methods discussed in Sections 3 and 4. Specifically, the authentic National Grid dataset presented in Section 1 includes data from 38,757 legitimate customers and 3,615 fraudulent customers, with each customer's electricity sequence consisting of 1,035 sampling points. The dataset is split into a training set and a test set in an 80:20 ratio, with the test set featuring user data that exhibits five distinct patterns of electricity theft. To thoroughly evaluate the model's classification performance under conditions of sample imbalance, we employ several metrics: the detection rate (DR), Macro-F1 score, and area under the curve (AUC). These metrics are defined as follows:

$$DR = \frac{TP}{TP + FN} \tag{36}$$

$$Macro - F1 = \frac{1}{2}\sum_{c=0}^{1}\frac{precision_i^c recall_i^c}{precision_i^c + recall_i^c} \tag{37}$$

$$AUC = \frac{1}{MN}\left[\sum_{i\in C} p_i - 0.5M(M+1)\right] \tag{38}$$

where TP represents the number of electricity theft samples correctly categorized by the model; FN represents the number of electricity theft samples identified as normal samples by the model; M and N represent the number of electricity theft samples and normal samples, respectively; P_i represents the number of electricity theft samples that are ranked higher than normal samples, calculated from all the normal samples for each individual electricity theft sample; and C is the set of electricity theft samples. In this study, we define an electricity theft user as a user who has at least one day of electricity theft behavior during the monitoring period. Since the data used are long time series data, the judgment criteria chosen are: if the number of days of electricity theft is greater than or equal to 1 day during the monitoring period, and the frequency of electricity theft is greater than or equal to 1 time, the user is labelled as an electricity theft user.

5.1. Optimized configuration and analysis of electricity theft detection results

First, to prevent the issue of gradient infinity caused by missing data, our paper interpolates the missing values in the dataset using the data interpolation method outlined in Section 2.1. Importantly, to preserve the potential abnormal behavior of electricity theft users, this paper only applies interpolation to normal users using TimeGAN data. Following the principle of "better but different", this paper selects 11 advanced classifiers in the field of data processing, including: Convolutional Neural Networks (CNNs), LSTMs, Temporal Convolutional Networks (TCNs), and Back-Propagation Neural Networks (BPNNs) in the category of Neural Networks; and learning with Light Gradient Boosting Machine (LightGBM), Extreme Gradient Boosting Machine (XGBoost), and Random Forest (RF); and traditional machine learning models with SVM, K Nearest Neighbors (KNN), Logistic Regression (LR), and Naive Bayes (NB).

The upper WOA population size is set to 150, with 200 iterations, while the lower WOA population size is 60, accompanied by 50 iterations. The optimization utilizes the normalized training set data as the input, resulting in the outcomes illustrated in Table 1. In order to verify the performance of WOA in the optimization task, we compare it with three other heuristic algorithms (Sparrow Search Algorithm [42], the classical Particle Swarm Optimization [43], and the newly proposed Chaos Game Optimization [44]) under the same optimization task, and the results of the are shown in Table 1, WOA in the ability to complete optimized configuration tasks faster and more efficiently.

The optimization utilizes the normalized training set data as the input, resulting in the outcomes illustrated in Table 2.The frequent selection of the integrated classifier suggests its superior applicability in the electricity theft detection task. Notably, the XGBoost classifier is chosen as both the base classifier and the meta classifier, indicating its effectiveness in fusing detection results from various classifiers and its robust generalization capability. After the optimization process, the total number of synthesized samples increases to 18,954, supporting the assertion made in [40] that simply synthesizing minority class samples to match the effectiveness of majority class samples is not always the most effective strategy in electricity theft detection. This finding emphasizes the importance of optimizing the number of synthesized samples, as discussed in this paper.

To demonstrate the superiority of the electricity theft detection model presented in this paper, our paper compares the Stacking integrated classifier with 11 state-of-the-art classifiers, whose hyperparameters have been sourced from external repositories. The resulting DR scores and Macro-F1 scores are displayed in Fig. 7. It is important to note that the Macro-F1 scores address metric bias resulting from sample size by averaging the F1 scores of both positive and negative classes; thus, they tend to be lower. Conversely, while the weighted F1 score typically hovers around 0.9—appearing high—this is largely due to the disproportionate number of normal users in the test set. Consequently, the evaluation metrics favor the recognition accuracy of normal users, failing to adequately differentiate the comprehensive classification capabilities of the various models. This paper observes significant variations in DRs across different models, with the CNN and TCN classifiers—equipped with feature extraction capabilities—showing notably higher DRs. Overall, the classifiers developed in this paper demonstrate optimal classification performance across nearly all categories, achieving an average Macro-F1 score that is 12.93 % higher and an average DR score that is 15.78 % higher than their counterparts. This strongly supports the superiority of the proposed electricity theft detection model. In contrast, the NB classifier performs the worst in these experiments, highlighting that its fundamental assumption of feature independence fails to meet the requirements of the electricity theft detection task and underscoring the presence of coupled temporal correlations in users' electricity consumption time series.

For comparison, our paper utilized samples without data enhancement to train the models, and the accuracy of each electricity theft detection model is shown in Fig. 9. It can be seen that the data enhancement method proposed in this paper significantly improves the DR scores for each model by synthesizing samples that represent various electricity theft patterns. Additionally, the comprehensive learning of electricity theft samples enhances the overall classification performance of the models, leading to increased Macro-F1 scores across the board. Notably, both before and after data enhancement, the Stacking integrated electricity theft detection model introduced in this paper consistently demonstrates the highest effectiveness and best generalization ability.

In order to demonstrate the effectiveness of the optimized Stacking configuration proposed in this paper, this paper compares it with

Table 1
Performance comparison of four heuristic algorithms.

heuristic algorithm	Number of iterative convergence	Time used (h)
WOA(This article uses)	**148**	**5.64**
Sparrow Search Algorithm	198	10.35
Particle Swarm Optimization	187	9.78
Chaos Game Optimization	161	6.12

Table 2
Optimization configuration results.

Upper level heuristic optimization results	Base classifiers:	lightGBM, XGBoost, CNN, TCN, LSTM, KNN
	Meta-classifiers:	XGBoost
	Number of data enhancement samples:	3852, 3674, 4548, 4017, 2863
Lower level heuristic optimization results	LightGBM:	learning_rate = 0.0581, n_Estimators = 412, Num_leaves = 28, min_child_sample = 13, max_depth = 5, reg_lambda = 0.054
	XGBoost(Base):	learning_rate = 0.0603, n_estimators = 359, Num_leaves = 31, min_child_sample = 14, max_depth = 5, reg_lambda = 0.077
	CNN:	Conv2D(filters = 16, kernel_size=(3,3)), Conv2D(filters = 24, kemel_size=(3,3)),MaxPooling2D (pool_size=(2.2) , Dense(units = 31,activation='relu'), Dense(units = 1,activation='relu'), learning_rate = 0.0134
	KNN:	n_neighbors = 11, algorithm='ball_tree', leaf_size = 31, weights='distance'
	LSTM:	units_1 = 75, units_2 = 48, dropout = 0.116, activation='tanh', leaming_rate = 0.0337
	TCN:	nb_filters = 51, kernel_size = 13, nb_stacks = 2
	XGBoost(Meta):	learning_rate = 0.0582, n_estimators = 368, Num_leaves = 31, min_child_sample = 13, max_depth = 5, reg_lambda = 0.149

Stacking configuration from other recent studies, as detailed in Table 3. Notably, Ref. [18] pertains to electricity theft detection, while [48] focuses on fault diagnosis. The ROC curves and corresponding AUC scores for each configuration scheme are illustrated in Fig. 10. Configuration 1 lacks comprehensiveness regarding algorithmic diversity, resulting in lower AUC scores on the dataset utilized in this study. Configuration 2 improved upon this by selecting base classifiers and *meta*-classifiers based on model performance within a mixed dataset, thereby achieving higher accuracy than Configuration 1. Configuration 3 considered nine different classifiers as base classifiers, which enhanced diversity; however, the inclusion of more models without a clear preference led to significant information redundancy, resulting in lower accuracy compared to the proposed configuration in this paper. The results from

Table 3
Stacking configuration for different studies.

number	Stacking Integrated Classifier Configuration Scheme
Configuration 1 [45]	Base classifiers: RF, SVM, GBDT, Deep forest; Meta-classifier: XGBoost
Configuration 2 [46]	Base Classifier: KNN, LR, RF, XGBoost; Meta Classifier: XGBoost
Configuration 3 [47]	Base Classifier: Extreme Random Trees (ERT), ridge regression (RR), RF, KNN, BPNN, CNN, LSTM, LightGBM, XGBoost;and Meta-classifier: XGBoost
Configuration 4 [48]	Base Classifier: NB, BPNN, SVM, LightGBM, Adaptive Boosting (AdaBoost); Meta Classifier: LR

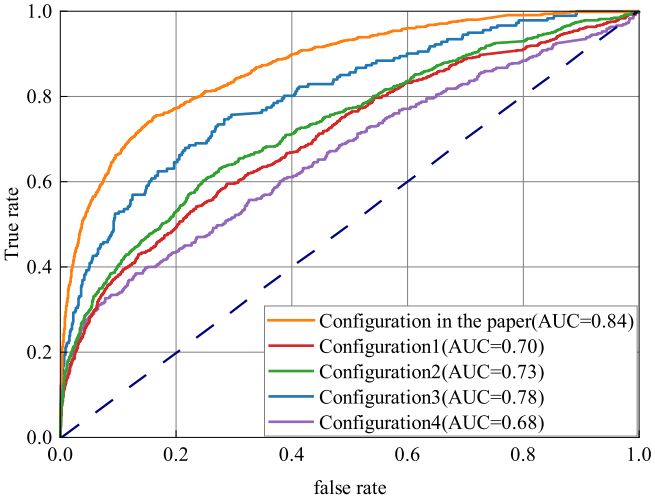


Fig. 10. ROC curves for different Stacking configurations.

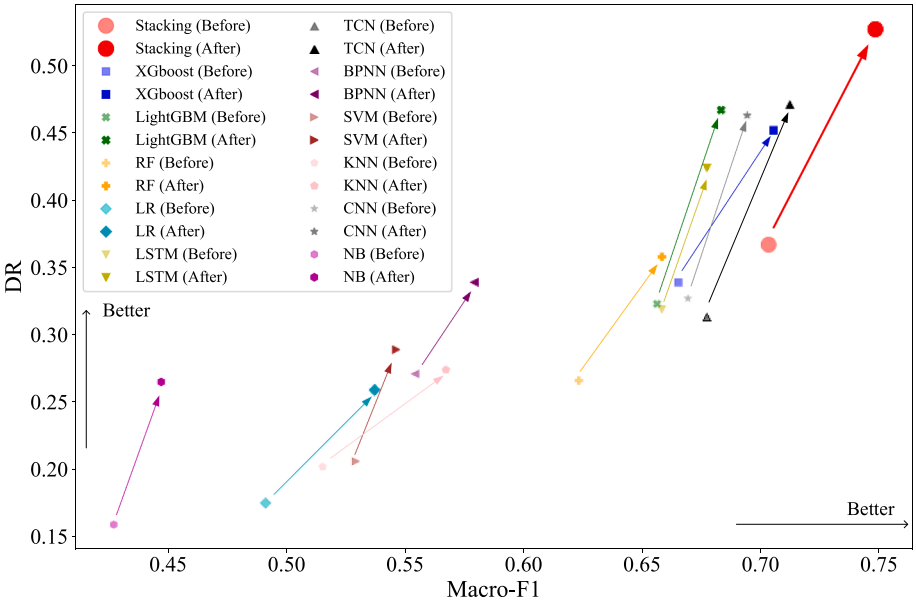


Fig. 9. The accuracy of various electricity theft detection models before and after data augmentation.

Configurations 3 and 4 indicate that Stacking configurations successful in fault diagnosis do not necessarily translate to effective electricity theft detection. This highlights the necessity of optimizing configurations for specific problems. Overall, three of the configurations utilized XGBoost as the *meta*-classifier, further validating the rationale behind the configurations presented in this paper, which yielded the best results for electricity theft detection.

5.2. Data interpolation effect analysis

To visualize the data filling effect, this paper manually simulates missing measurements in the time series of normal users by masking certain values and treating them as real data for validation. The percentage of missing data ranges from 10 % to 40 %, with the remaining data utilized for training. The baseline methods for data interpolation include advanced K-nearest neighbor (KNN), GAN, auto-encoder (AE), and LI. The experiment is repeated 10 times to calculate the average absolute error for each method at various missing rates, as illustrated in Fig. 11. The results indicate that the difficulty of data interpolation increases for all models as the missing rate rises. LI demonstrates high accuracy at lower missing rates (10 %), primarily because most missing points are single-point omissions, allowing surrounding data points to better represent the true value of the missing points. However, as the missing data rate increases, the traditional LI method gradually loses its effectiveness, suggesting that the linear relationship assumption becomes insufficient in this context. In contrast, deep learning models exhibit greater robustness in handling more missing data by effectively capturing and modeling complex patterns and relationships, enabling them to make reasonable predictions despite the lack of information. Overall, the TimeGAN data interpolation methods proposed in this paper achieve the highest accuracy across different percentages of missing data.

The classification results serve as indirect evidence of interpolation accuracy. This paper examines how different data interpolation methods impact the overall performance of the model when utilizing Stacking as the classifier. To ensure consistency, the missing values in the training set are set at 10 % and 30 %, and the experiment is repeated five times. The Macro-F1 scores before and after data interpolation using various methods are illustrated in Fig. 12 and Fig. 13. The findings highlight that data integrity significantly affects the effectiveness of electricity theft detection. As the missing rate increases, the likelihood of masking key information also rises, resulting in a substantial decrease in classifier accuracy. Notably, at a lower missing rate (10 %), LI aligns more closely with the true distribution compared to KNN, resulting in better detection accuracy for electricity theft. Conversely, at a higher missing rate (30 %), both LI and KNN show significantly reduced effectiveness in comparison to deep learning methods, which corresponds with the data interpolation error results shown in Fig. 11. Overall, the model proposed in this paper achieves an average Macro-F1 score that is 2.325 % higher at a 10 % missing rate and 11.1 % higher at a 30 % missing rate compared to other data interpolation methods. This indicates that the approach outlined in this paper yields estimates that more accurately reflect the real data distribution, thereby enabling the model to more effectively distinguish between normal and electricity theft users.

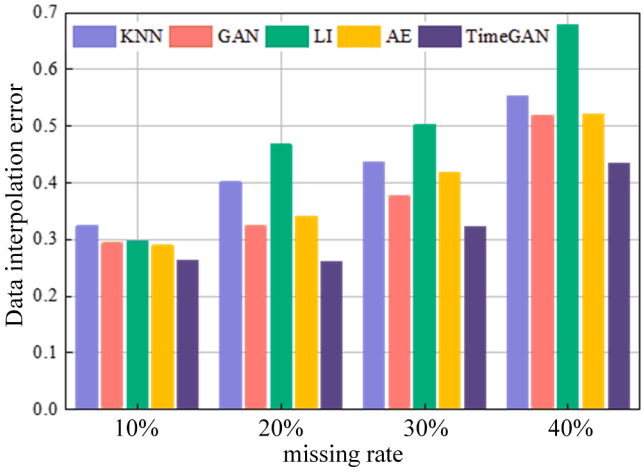


Fig. 11. The data interpolation error of each algorithm with a missing rate from 10% to 40%.

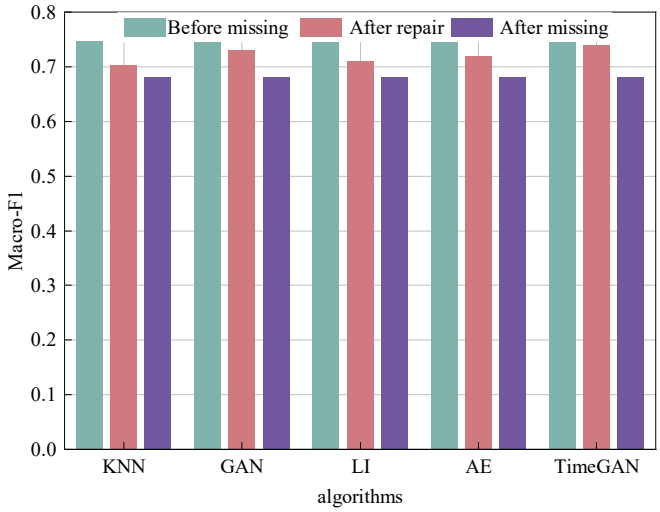


Fig. 12. Changes in classification accuracy before and after 10% missing data interpolation.

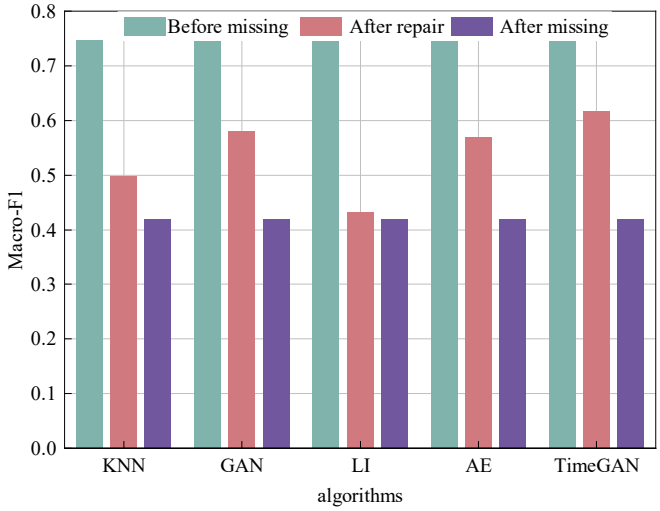


Fig. 13. Changes in classification accuracy before and after 30% missing data interpolation.

5.3. Data enhancement effectiveness analysis

To demonstrate the effect of data enhancement while addressing intra-class imbalance, our paper classified the imbalanced data and the synthesized data using the Stacking integrated classifier, resulting in the confusion matrix presented in Fig. 14. In this matrix, electricity theft users are labeled as 1 and normal users as 0, and the counts have been normalized for easier visualization. Fig. 14(a) illustrates the classification results without any data enhancement. Fig. 14(b) depicts the results when addressing only inter-class imbalance through data enhancement using traditional TimeGAN, with the number of samples aligned with the

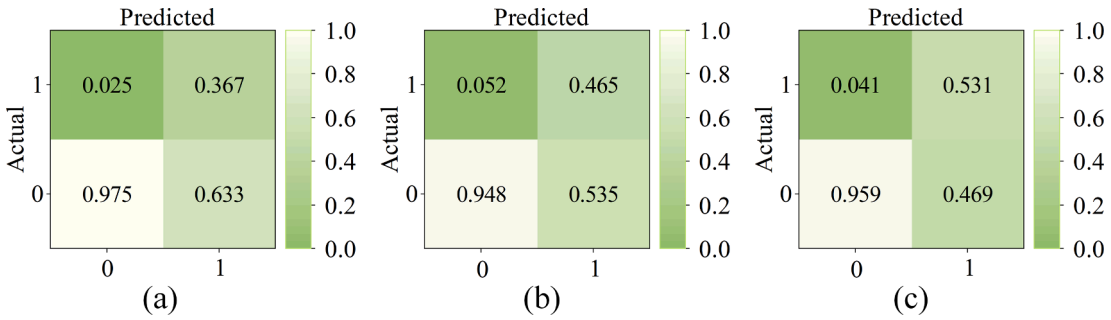


Fig. 14. Confusion matrix before and after data augmentation.

optimized configuration results. Fig. 14(c) presents the classification results after considering intra-class imbalance through conditional CTimeGAN for data enhancement. The results in Fig. 14(a) reveal a significant classification bias, with only 36.7 % of electricity theft users correctly identified. Following data enhancement, the model increases its focus on detecting electricity theft users, which can lead to a higher false DR where some normal users are misclassified as electricity theft users. However, when only inter-class imbalance is addressed, the model shows a higher false DR compared to when intra-class imbalance is also considered. The latter approach reduces noise interference by synthesizing electricity theft samples in a more balanced manner. The data enhancement method proposed in this paper, which takes intra-class imbalance into account (Fig. 14(c)), allows for a more comprehensive learning of different electricity theft patterns. This results in accurately identifying 48 more electricity theft users (a 6.6 % increase) compared to the approach that only considers inter-class imbalance (Fig. 14(b)).

Furthermore, this paper analyzes the change in the intra-class IR before and after data enhancement using an unsupervised clustering method. Three datasets are established: Dataset 1 consists of user data without any data enhancement, Dataset 2 includes user data enhanced by TimeGAN, which considers only inter-class imbalance, and Dataset 3 comprises user data enhanced by CTimeGAN, which accounts for intra-class imbalance. The enhanced samples from each dataset are re-clustered using the K-shape algorithm, allowing for a comparison of the number of sequences in each cluster before and after data enhancement, as presented in Table 4. The results indicate a significant intra-class imbalance remains after data enhancement when only inter-class imbalance is addressed. In contrast, the distribution of sub-modes within the electricity theft dataset becomes more balanced following sample synthesis using the method proposed in this paper. This demonstrates that our approach effectively alleviates the issue of intra-class imbalance, contributing to a more equitable representation of electricity theft patterns in the dataset.

To validate the superiority of the proposed CTimeGAN data augmentation method, CTimeGAN is compared with different benchmarks, including GAN, Synthetic Minority Over-Sampling (SMOTE), Variable Auto-Encoder (VAE), and Random Over-Sampling (ROS). Electricity theft samples are synthesized by these data enhancement methods, and the accuracy post-enhancement is visualized based on the Stacking integrated classifier, as illustrated in Fig. 15. The results indicate that the overall performance of ROS is the least effective, which is due to the repeated sampling of a few classes of samples leading to an increase in redundant information. This can be attributed to the

Table 4
Number of samples for each electricity theft mode before and after data augmentation.

Data set number	Number of clusters	Intra-class IR
Data set 1	231, 505, 646, 1057, 1176	5.09
Data set 2	1748, 2771, 4032, 6586, 7432	4.25
Data set 3	3262, 3679, 5041, 6217, 4370	1.91

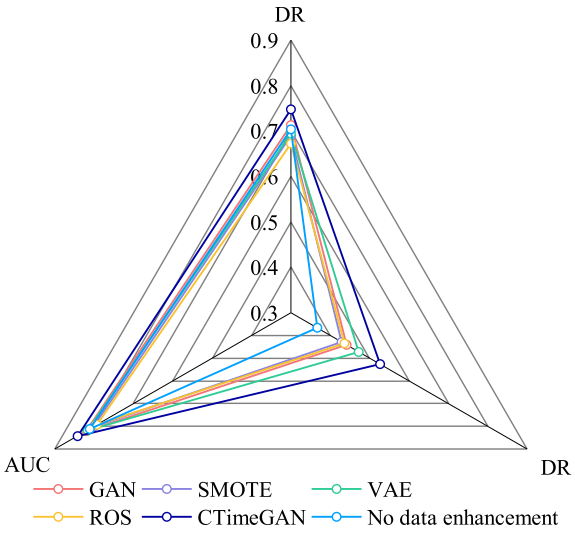


Fig. 15. Comparison of classification accuracy of data augmentation algorithms.

repeated sampling of a limited number of classes, which increases redundant information and diminishes the model's generalization ability. While ROS enhances the DR, it also leads to a higher rate of normal users being misclassified as electricity theft users. On the other hand, the GAN and VAE deep learning methods are capable of learning complex data distributions and generating higher-quality samples compared to SMOTE and ROS. However, they struggle to capture the temporal dependencies in the data, rendering them less effective than CTimeGAN. Notably, VAE tends to overfit the distribution of electricity theft data, which enhances the model's ability to identify these users and improves the DR, but ultimately results in a decrease in overall accuracy. In contrast, the CTimeGAN method proposed in this paper effectively learns the data distribution while also capturing the dynamics and transformation patterns within the time series. This approach enhances the model's ability to detect electricity theft users while introducing minimal noise. The results show that CTimeGAN achieves an average DR score that is 8.23 % higher than the baseline method, 3.45 % higher than the AUC score, and 5.60 % higher than the Macro-F1 score.

5.4. Optimization algorithm effect analysis

To verify the superiority of the time-varying binary whale optimization algorithm (TV-WOA) proposed in this paper, it was compared with the traditional T-type and S-type binary transfer functions. With the population size set to 150 and the number of iterations set to 200, the convergence results of the three optimization algorithms are illustrated in Fig. 16. The results indicate that while the T-type and S-type transfer functions exhibit a faster search speed during the initial stages of the

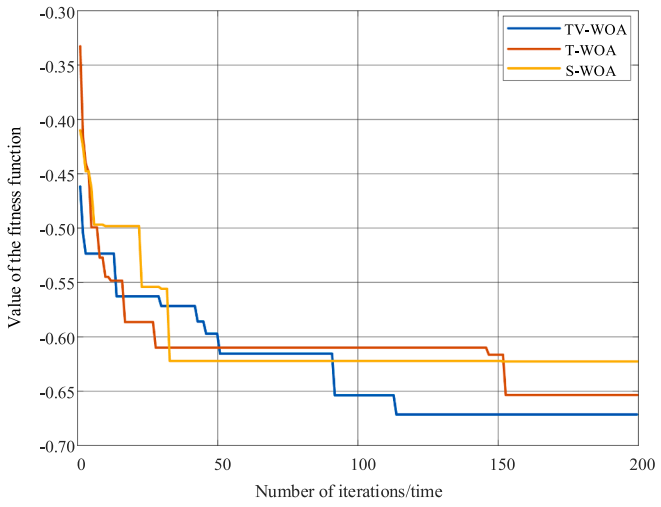


Fig. 16. The iterative process of different binary transfer functions.

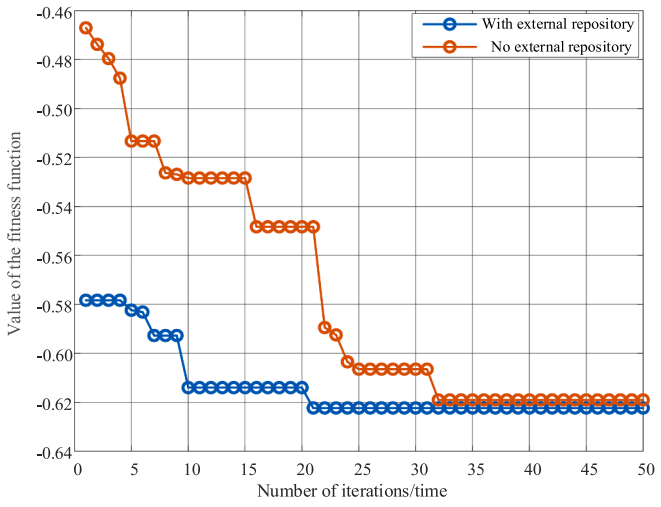


Fig. 17. Comparison of iterative processes with and without external repositories.

algorithm's iterations, they are prone to falling into local optima in the later stages. In contrast, TV-WOA effectively maintains a balance between exploration and exploitation, demonstrating superior global search capability compared to the traditional methods.

To verify the effectiveness of the proposed external repository, the lower layer iterative process corresponding to a specific round of upper layer iterations was selected. The search performance between the iterative process with and without the external repository was then compared, as depicted in Fig. 17. The results indicate that the upper-layer optimization model provides various stacking configurations, causing the lower layer to optimize different decision variables. This leads to slower convergence in the traditional scheme, which relies on randomly initialized hyperparameters. In contrast, the external repository incorporates a heuristic memory function, enabling it to offer initial hyperparameters that are closer to the optimal solutions for lower-layer heuristic optimization. Consequently, this approach facilitates the attainment of a higher-quality solution set with fewer iterations, thereby demonstrating the effectiveness of the proposed method.

6. Conclusion

This paper presents an electricity theft detection method that utilizes a two-stage TimeGAN integrated with a two-layer Stacking optimization framework. Key contributions include the introduction of intra-class imbalance in electricity theft detection and the development of a two-phase TimeGAN model to tackle data incompleteness and class imbalance. Additionally, we construct a two-layer heuristic optimization framework for the dynamic optimization of stacking model configurations and hyperparameters, providing an effective solution for industrial applications. The optimized Stacking classifier significantly outperforms its unoptimized counterpart, achieving superior classification performance. Specifically, our methodology results in an average increase of 8.23 % in DR scores, 3.45 % in AUC scores, and 5.60 % in Macro-F1 scores compared to the baseline methodology.

However, the study is limited by the dataset, which only includes power consumption data and excludes other relevant features. Future work will focus on leveraging multi-dimensional power data by incorporating voltage, current, and calendar information. Additionally, we aim to explore more diverse methods, such as few-shot learning, while utilizing a broader range of datasets to validate and enhance the proposed method.

CRediT authorship contribution statement

Leijiao Ge: Formal analysis. **Jingjing Li:** Simulation experiment. **Tianshuo Du:** Writing – original draft. **Luyang Hou:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the Science and Technology Major Project of the Department of Science and Technology of Yunnan Province in China (No.202402AF080006), in part by National Key R&D Program of China (2022YFB2403800), in part by National Key R&D Program of China (2022ZD0116900) and in part by Natural Science Foundation of Tianjin (22JCZDJC00660).

Data availability

The data that has been used is confidential.

References

[1] Chen J, Nanehkaran YA, Chen W, Liu Y, Zhang D. Data-driven intelligent method for detection of electricity theft. Int J Electr Power Energy Syst 2023;148:108948.
[2] Ahrari M, Shirini K, Gharehveran SS, Ahsaee MG, Haidari S, Anvari P. A security-constrained robust optimization for energy management of active distribution networks with presence of energy storage and demand flexibility. J Energy Storage 2024;84:111024.
[3] Yan Z, Wen H. Performance analysis of electricity theft detection for the smart grid: an overview. IEEE Trans Instrum Meas 2022;71:2502928.
[4] Xia X, Xiao Y, Liang W. ABSI: an adaptive binary splitting algorithm for malicious meter inspection in smart grid. IEEE Trans Inf Forensics Secur 2019;14(2):445–58.
[5] Xia X, Lin J, Jia Q, Wang X, Ma C, Cui J, et al. ETD-ConvLSTM: a deep learning approach for electricity theft detection in smart grids. IEEE Trans Inf Forensics Secur 2023;18:2553–68.
[6] Narayanan VL, Dhaked DK, Sitharthan R. Improved machine learning-based pitch controller for rated power generation in large-scale wind turbine. Renewable Energy Focus 2024;50:100603.
[7] Dhaked DK, Kumar P, Ganguly S. Development of Data Driven Model for Proton Exchange Membrane Fuel Cell Using Machine Learning Approaches. In: 2024 IEEE 3rd International Conference on Control, Instrumentation, Energy and Communication (CIEC). IEEE; 2024. p. 67–72.
[8] Wang J, Si Y, Zhu Y, Zhang K, Yin S, Liu B. Cyberattack detection for electricity theft in smart grids via stacking ensemble GRU optimization algorithm using federated learning framework. Int J Electr Power Energy Syst 2024;157:109848.
[9] Dhaked DK, Dadhich S, Birla D. Power output forecasting of solar photovoltaic plant using LSTM. Green Energy Intell Transp 2023;2(5):100113.

[10] Chen H, Ma R, Liu X, Liu R. Detecting energy theft with partially observed anomalies. Int J Electr Power Energy Syst 2024;162:110323.
[11] Jindal A, Dua A, Kaur K, Singh M, Kumar N, Mishra S. Decision tree and SVM-based data analytics for theft detection in smart grid. IEEE Trans Ind Inf 2016;12(3):1005–16.
[12] Cai Q, Li P, Zhao Z, Wang R. Dynamic electricity theft behavior analysis based on active learning and incremental learning in new power systems. Int J Electr Power Energy Syst 2024;162:110309.
[13] Liao W, Takiddin A, Tariq M, Chen S, Ge L, Yang Z. Sample adaptive transfer for electricity theft detection with distribution shifts. IEEE Trans Power Syst 2024;39(6):7012–24.
[14] Elgarhy I, Badr MM, Mahmoud MMEA, Mahmoud MN, Alsabaan M, Ibrahem MI. Securing smart grid false data detectors against white-box evasion attacks without sacrificing accuracy. IEEE Internet Things J 2024;11(20):33873–89.
[15] Zidi S, Mihoub A, Qaisar SM, Krichen M, Abu Al-Haija Q. Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment. Journal of King Saud University-Computer and Information Sciences 2023;35(1):13–25.
[16] Punmiya R, Choe S. Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. IEEE Trans Smart Grid 2019;10(2):2326–9.
[17] Yan Z, Wen H. Electricity theft detection base on extreme gradient boosting in AMI. IEEE Transactions on Instrumentation and Measurement 2021;70:2504909.
[18] Li S, Meng W, Liu C, He S. Feature attention distillation defense for backdoor attack in artificial neural network-based electricity theft detection. IEEE Internet Things J 2024:1.
[19] Gao Y, Foggo B, Yu N. A physically inspired data-driven model for electricity theft detection with smart meter data. IEEE Trans Ind Inf 2019;15(9):5076–88.
[20] Shirini K, Aghdasi HS, Saeedvand S. A comprehensive survey on multiple-runway aircraft landing optimization problem. Int J Aeronaut Space Sci 2024;25(4):1574–602.
[21] Zhou W, Li B, Xiao H, Xiao H, Wang W, Zheng Y, et al. Electricity theft detection of residential users with correlation of water and electricity usage. IEEE Trans Ind Inf 2024;20(4):5339–51.
[22] Gharehveran SS, Ghassemzadeh S, Rostami N. Two-stage resilience-constrained planning of coupled multi-energy microgrids in the presence of battery energy storages. Sustain Cities Soc 2022;83:103952.
[23] Gharehveran SS, Zadeh SG, Rostami N. Resilience-oriented planning and pre-positioning of vehicle-mounted energy storage facilities in community microgrids. Journal of Energy Storage 2023;72:108263.
[24] Liao W, Zhu R, Ishizaki T, Li Y, Jia Y, Yang Z. Can Gas consumption data improve the performance of electricity theft detection? IEEE Trans Ind Inf 2024;20(6):8453–65.
[25] Yang S, Dong M, Wang Y, Xu C. Adversarial recurrent time series imputation. IEEE Trans Neural Networks Learn Syst 2023;34(4):1639–50.
[26] Choudhury SJ, Pal NR. Deep and structure-preserving autoencoders for clustering data with missing information. IEEE Trans Emerging Top Comput Intell 2021;5(4):639–50.
[27] Tao Y, Qiu J, Lai S. A data-driven management strategy of electric vehicles and thermostatically controlled loads based on modified generative adversarial network. IEEE Trans Transp Electrif 2022;8(1):1430–44.
[28] Pereira J, Saraiva F. Convolutional neural network applied to detect electricity theft: A comparative study on unbalanced data handling techniques. Int J Electr Power Energy Syst 2021;131:107085.
[29] Gunturi SK, Sarkar D. Ensemble machine learning models for the detection of energy theft. Electr Power Syst Res 2021;192:106904.
[30] Shirini K, Aghdasi HS, Saeedvand S. Modified imperialist competitive algorithm for aircraft landing scheduling problem. J Supercomput 2024;80(10):13782–812.
[31] Shirini K, Aghdasi HS, Saeedvand S. Multi-objective aircraft landing problem: a multi-population solution based on non-dominated sorting genetic algorithm-II. J Supercomput 2024;80(17):25283–314.
[32] Gharehveran SS, Shirini K, Khavar SC, Mousavi SH, Abdolahi A. Deep learning-based demand response for short-term operation of renewable-based microgrids. J Supercomput 2024;80(18):26002–35.
[33] Lin G, Feng H, Feng X, Wen H, Li Y, Hong S, et al. Electricity theft detection in power consumption data based on adaptive tuning recurrent neural network. Front Energy Res 2021;9:773805.
[34] Yao R, Wang N, Ke W, Liu Z, Yan Z, Sheng X. Electricity theft detection in incremental scenario: a novel semi-supervised approach based on hybrid replay strategy. IEEE Trans Instrum Meas 2023;72:2530012.
[35] Liao W, Yang Z, Bak-Jensen B, Pillai JR, Von Krannichfeldt L, Wang Y, et al. Simple data augmentation tricks for boosting performance on electricity theft detection tasks. IEEE Trans Indus Appl 2023;59(4):4846–58.
[36] "Electricity Theft Detection," [Online]. Available: https://github.com/henryRDlab/ElectricityTheftDetection, 2018.
[37] Gu D, Gao Y, Chen K, Shi J, Li Y, Cao Y. Electricity theft detection in AMI with low false positive rate based on deep learning and evolutionary algorithm. IEEE Trans Power Syst 2022;37(6):4568–78.
[38] Yang L, Zhang Z. A Deep attention convolutional recurrent network assisted by K-shape clustering and enhanced memory for short term wind speed predictions. IEEE Trans Sustainable Energy 2022;13(2):856–67.
[39] Yoon J, Jarrett D, Van der Schaar M. Time-series generative adversarial networks. Adv Neural Inf Proces Syst 2019;32.
[40] Yao R, Wang N, Ke W, Chen P, Sheng X. Electricity theft detection in unbalanced sample distribution: a novel approach including a mechanism of sample augmentation. Appl Intell 2023;53(9):11162–81.
[41] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw 2016;95:51–67.
[42] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm. Syst Sci Control Eng 2020;8(1):22–34.
[43] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, Vol. 4. ieee; 1995. p. 1942–8.
[44] Talatahari S, Azizi M. Chaos game optimization: a novel metaheuristic algorithm. Artif Intell Rev 2021;54(2):917–1004.
[45] Cheng C, Peng X, Zeng Y, Xu F. An abnormal power user recognition method for stacking integrated structures with different models. Power System Technology 2021;45(12):4828–36.
[46] You W, Li Q, Yang N, Shen K, Li W, Wu Z. Electricity theft detection based on multiple different learner fusion by stacking ensemble learning. Automation of Electric Power Systems 2022;46(24):178–86.
[47] Ma L, Geng Y, Liang S, Cheng D. Anomaly warning of wind turbine gearbox oil pool temperature based on stacking fusion of multiple models. Proc CSEE 2023;43(S1):242–51.
[48] Tian S, Li J, Zhang J, Li C. STLRF-Stack: a fault prediction model for pure electric vehicles based on a high dimensional imbalanced dataset. IET Intel Transport Syst 2023;17(2):400–17.