A Hybrid ConvLSTM-Based Anomaly Detection Approach for Combating Energy Theft

Hong-Xin Gao, Student Member, IEEE, Stefanie Kuenzel, Senior Member, IEEE, and Xiao-Yu Zhang, Member, IEEE

Abstract—In a conventional power grid, energy theft is difficult to detect due to limited communication and data transition. The smart meter along with big data mining technology leads to significant technological innovation in the field of energy theft detection (ETD). This article proposes a convolutional long short-term memory (ConvLSTM)-based ETD model to identify electricity theft users. In this work, electricity consumption data are reshaped quarterly into a 2-D matrix and used as the sequential input to the ConvLSTM. The convolutional neural network (CNN) embedded into the long short-term memory (LSTM) can better learn the features of the data on different quarters, months, weeks, and days. Besides, the proposed model incorporates batch normalization. This technique allows the proposed ETD model to support raw format electricity consumption data input, reducing training time and increasing the efficiency of model deployment. The result of the case study shows that the proposed ConvLSTM model exhibits good robustness. It outperforms the multilayer perceptron (MLP) and CNN-LSTM in terms of performance metrics and model generalization capability. Moreover, the result also demonstrates that K-fold cross validation can improve the ETD prediction accuracy.

Index Terms—Binary classification, convolutional long short-term memory (ConvLSTM), deep learning, energy theft, smart grid.

NOMENCLATURE

| Acronym | Description |
|---------|-------------|
| ETD     | Energy theft detection. |
| ConvLSTM| Convolutional long short-term memory. |
| CNN     | Convolutional neural network. |
| LSTM    | Long short-term memory. |
| MLP     | Multilayer perceptron. |
| AMR     | Automated meter reading. |
| AMI     | Advanced metering infrastructure. |
| NTL     | Nontechnical losses. |
| SVM     | Support vector machine. |
| RNN     | Recurrent neural network. |
| ReLU    | Rectified linear unit. |
| SMOTE   | Synthetic minority oversampling technique. |
| KNN     | K-nearest neighbors. |
| IQR     | Interquartile range. |
| ROC     | Receiver operating characteristic. |
| AUC     | Area under the curve. |
| PR      | Precision–recall. |
| PCC     | Pearson correlation coefficient. |

I. INTRODUCTION

The smart grid has made great progress as a mainstream trend in the current development of electricity networks. It effectively combines the electricity consumption of grid service users with intelligent communication and monitoring, enabling an evolution from AMR to AMI [1]. AMI is a critical part of the smart grid layout, integrating intelligent measurement, collection, storage, and energy data analysis [2]. It also marks a shift toward intelligent and digital communication between utility companies and electricity consumers. As the core equipment of the smart grid, smart meters not only provide precise and synchronized measurement and collection from end users and provide efficient data guarantee for AMI intelligent analysis [3]. As the information and communication modules in smart meters continue to be integrated and iterated, energy theft through a physical approach is becoming more advanced and covert. For instance, attacks go beyond traditional meter tampering by exploiting system vulnerabilities to manipulate meter readings and execute cyberattacks [4]. Energy theft is a serious social hazard, which can be illustrated as illegal electricity customers using utility’s energy in breach of contract or manipulating their meter reading to avoid paying the bill [4]. This problem causes huge financial losses to utility companies, seriously infringes on the legal rights of normal electricity users, and disrupts the fair market environment for electricity consumption. Electricity theft has become one of the main causes of NTLs in smart grids. According to the recent report, the global electricity supply sector loses approximately U.S. $25 billion annually to NTLs, including theft and fraud. For example, in India, annual losses due to electricity theft are approximately U.S. $4.5 billion [5]. The 2020s will be a critical period for the development of smart grids globally, with North America, Western Europe, and other countries...
already on the path to building smart grids. However, this is still an area of massive investment for emerging markets. The report states that the 50 emerging markets will invest over U.S. $40.7 billion in the coming years [6]. This is certainly a signal to drive the global AMI layout’s development and implies a national commitment in terms of smart grid NTL governance.

NTL-based approaches to ETD can be divided into three categories [7]: data-oriented, network-oriented, and hybrid models. Network-oriented approaches are usually based on localized AMR, and sensor ideas, e.g., [8], [9], propose smart substations and network voltage sensitivity concepts to identify theft states. Performance is based on high-cost equipment and staff training, among other aspects, and there is certain inflexibility in future deployment and modality transformation. The data-oriented approach is based on machine learning techniques and proceeds through supervised and unsupervised learning [7]. Jamaica is one of the first countries to use machine learning to combat energy theft. Its public service company identified electricity theft customers around 2017 by deploying an AMI-based machine learning model [10]. Machine learning allows for feature engineering and modeling electricity consumption data, which also implies optimization of manual detection methods. Currently, classical machine learning and deep learning are two data-oriented approaches to electricity theft detection. In classical machine learning for binary classification issues, for instance, SVM and extreme gradient boosting (XGBoost) classifiers for ETD are supervised learning methods [11], [12]. They are based on labeling electricity consumption samples to identify theft customers using electricity consumption data features. Classical decision trees and random forests also show good identification accuracy in ETD [13].

With the development of AMI and advances in neural networks, deep learning-based ETD methods have also been introduced to combat energy theft. The advantage of AMI is that it provides a large amount of data support and a comprehensive approach to energy monitoring, which caters to the characteristics of deep learning that requires substantial data support to prevent overfitting models with optimal generalization [14]. Neural networks as a concept in deep learning have been widely used in computer vision, speech recognition, and anomaly detection [15]. Neural networks are robust to noise in the input data and mapping functions and can even support learning and prediction in the presence of missing values. Also, neural networks do not make strong assumptions about the mapping function and can easily learn linear and nonlinear relationships [16]. In addition to this, deep learning allows for automatic extraction in terms of feature engineering compared to classical machine learning. It is oriented toward data and demand pattern concepts, and cross-domain techniques can be well applied to ETD problems [14]. For example, CNN is currently the mainstream neural network for processing image classification and computer vision [17], which can be good for automatic feature extraction and global optimization. LSTM, as a variant of RNN, controls the transmission state by gating the state and can capture the relationship between time series effectively [18]. More importantly, LSTM solves the problem of gradient disappearance and gradient explosion problems during training long sequences [19]. In [13], CNN and LSTM are analyzed separately for comparison and outperformed classical machine learning techniques. In [20], [21], and [22], hybrid deep learning models showed better feature extraction and model structure expansion in ETD. LSTM has successfully solved a number of power system issues, such as load forecasting [23] and energy disaggregation [24].

Madhure et al. [22] built an ETD model in a classical CNN-LSTM stack. However, the CNN used for feature extraction in the front segment is restricted to 1-D data as input. Works [20], [21] are based on the CNN-LSTM with improvements such as expanding and deepening the convolution layers and data augmentation to optimize feature extraction and support 2-D electricity consumption data input. However, this method is limited by the architecture of the underlying model, and the CNN layer feature extraction is not fully embedded in the whole and can only be fed into the LSTM after feature extraction. Front-end input limits the scaling of multidimensional data and can affect the model to extract deeper and more subtle anomalous features. In terms of model optimization, Hasan et al. [20], Zheng et al. [21], and Madhure et al. [22] incorporated a dropout layer, which can effectively prevent the overfitting of the model. ReLU is used as an activation function, which only activates positive values. In the backpropagation process, each unit calculates its weight based on the loss values emanating from the upper layer [25]. The optimizer Adam is also applied in [20], which adaptively adjusts the learning rate.

Regarding imbalanced data, the classical SMOTE is used in [20]. The traditional SMOTE runs the risk of overlapping samples from a few classes, thus overfitting the model. In the development of deep learning, the novel ConvLSTM is proposed to be applied to predict spatiotemporal sequences for regression issues [26]. Compared to CNN-LSTM, it can optimize the excessive redundancy in temporal data. Importantly, ConvLSTM uses LSTM instead of pooling layers in CNN to reduce the loss of detailed local information and capture long-term dependencies in sequences. In human behavior recognition, ConvLSTM has better recognition accuracy and false alarm rates when dealing with multiple classification issues [27].

In this article, our main contribution is adopting the ConvLSTM architecture on the proposed ETD model. It supports fully connected layers for convolutional computation, replacing the matrix multiplication of traditional CNN-LSTM stacked attributes. This allows CNN feature extraction to be embedded throughout the model, allowing better extraction of local electricity usage features. On the other hand, it also facilitates the LSTM further to capture the deeper periodicity of the electricity consumption data. In addition, a batch-normalization technique is added to the proposed model. It supports raw electricity consumption data input, eliminating the need for tedious and time-consuming data pretransformation. We use an improved dataset balancing method, borderline-SMOTE, which generates more realistic data on electricity theft users for imbalanced datasets. Furthermore, it can reduce the overlapping of data and prevent overfitting of the model.
A. Overall System

The approach of this article is to build a novel ETD model based on ConvLSTM and verify its accuracy and robustness in identifying electricity theft users by simultaneous comparison and full validation with MLP and CNN-LSTM. The dataset contains two categories of normal users and electricity theft users and essentially deals with the binary classification issue by employing supervised learning. The key processes are shown in Fig. 1. The whole process is divided into three main steps, which are data preprocessing, model training, and model evaluation. The preprocessing step contains four sub-steps: 1) data cleaning, including removing and filling missing values with KNN imputation and filtering outliers with IQR; 2) visualize the dataset by power curves and PCCs to initially check for potential trends and correlations between the data; 3) train–test splitting: the dataset is split into 64% training, 16% validation, and 20% test data; and 4) solving imbalanced classification issues in training and validation datasets. The borderline-SMOTE sampling technique generates more realistic theft user data (for training and validation datasets only). The second step contains three sub-steps: 1) data transformation and reshaping—the batch-normalization technique is embedded in ConvLSTM and CNN-LSTM models without additional transformation steps and the raw electricity consumption data can be used directly in training and testing; 2) the optimal model is trained and selected using tenfold cross validation, fivefold cross validation, and no cross validation; and 3) tuning the hyperparameters to find the optimal model parameters and save the model, which performs the best with training data. In the final step evaluation, performance metrics are employed to evaluate the proposed ConvLSTM ETD model.

B. Data Transformation and Reshaping

This article uses a novel batch-normalization technique for all three models. This technique makes the training of deep neural networks more efficient. As stated by Ioffe and Szegedy [28], the batch-normalization method can accelerate deep network training by reducing internal covariate shift. It can be embedded directly behind each input layer in the normalization model without needing separate data transformation. In practice, this saves the time and number of training sessions consumed by changes in the fed data. More importantly, batch normalization has little effect on the initial weights and model hyperparameter changes while reducing generalization errors [28]. The main equations are given as follows:

\[
E_x = \frac{1}{m} \sum_{i=1}^{j} \mu_x(i) / B
\]  \hspace{1cm} (1)

\[
Var_x = \left( \frac{m}{m-1} \right) \frac{1}{m} \sum_{i=1}^{j} \sigma_x^2(i) / B
\]  \hspace{1cm} (2)

\[
y = \frac{y}{\sqrt{Var_x + \epsilon}} + \left( \beta + \frac{y E_x}{\sqrt{Var_x + \epsilon}} \right).
\]  \hspace{1cm} (3)

where \( y \) and \( \beta \) are the initial training parameters and \( m \) and \( j \) are the sample and batch values, respectively.

Prior to feeding the 1-D electricity consumption data into the model, the data require reshaping to match the model input shape. In the CNN-LSTM model, a 2-D matrix dividing the electricity consumption data into 3 × 3 × 1 by quarter is used, which is essentially a stacked structure of CNN and LSTM supporting 3-D tensor inputs, in this article, i.e., user quarterly electricity consumption data, number of quarters, and number of samples.

To ensure the consistency in model evaluation, the proposed model reshares the electricity consumption data into three dimensions, i.e., samples, time steps, and features, in the same way as the CNN-LSTM model and according to quarters. However, the novel ConvLSTM 2-D layer [29] supports 5-D tensor inputs, i.e., samples, time, rows, columns, and channels, which can also be interpreted as the time step being decomposed into rows × columns of image data points. In this article, time is the number of quarters, column is the quarterly electricity consumption data, and time step is divided

Fig. 1. Flowchart of the proposed ConvLSTM-based energy theft model with detailed data processing, model training, and evaluation process.
into quarterly quantities × quarterly data, i.e., each subseries contains a sequence of three months of electricity consumption data. Rows and channels are one because each user’s raw electricity consumption data is 1-D, and there are no additional raw features.

C. Stacking of Classical CNN and LSTM

The CNN consists of three main parts: the convolutional layer, the pooling layer, and the fully connected layer [15], [17]. In this article, it can be elaborated as the convolutional layer is responsible for extracting local features in 2-D electricity data, i.e., features are extracted in sliding window mode within each subsequence (2-D matrix segmentation according to quarter time). The pooling layer allows for more efficient dimension reduction than the convolution layer, which reduces the number of operations and effectively avoids overfitting. In the classical CNN-LSTM structure, CNN layers can be encapsulated in time-distributed layers [30], and the extracted features are flattened for use in LSTM.

LSTM network is an RNN trained by time backpropagation [18]. It has a unique formulation that avoids the problem that other RNNs cannot be trained and scaled, while it also overcomes the problem of vanishing and exploding gradients. Truncated backpropagation through time (TBPTT) [31] is a key concept in the training LSTM model. Unlike neurons, the memory blocks of LSTM networks contain states and outputs and are connected in layers [19]. Each block has three gates: a forget gate, an input gate, and an output gate [31]. In this article, the CNN-LSTM model can be used to control whether they are triggered or not by a sigmoid [25] activation function that produces an output between 0 and 1 in the binary classification. The classical LSTM equations [18] can be derived as follows: the input gate is (4), the forget gate is (5) and (6), and the output gate and mainline generation output are (7) and (8), respectively. Here, the hidden state and cell state are \( h_t \) and \( c_t \), respectively. \( f_t, i_t, \) and \( o_t \) are the activation vectors of forget gate, input gate, and output gate, respectively. \( \sigma \) denotes the sigmoid function, \( \odot \) denotes the Hadamard product, and \( b \) is the bias vector parameter

\[
i_t = \sigma(W_i x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i) \quad (4)
\]
\[
f_t = \sigma(W_f x_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f) \quad (5)
\]
\[
c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + W_{hc} h_{t-1} + b_c) \quad (6)
\]
\[
o_t = \sigma(W_o x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o) \quad (7)
\]
\[
h_t = o_t \odot \tanh(c_t). \quad (8)
\]

A sliding window is a method to transform time series into supervised learning [17]. A maximum pooling layer is added after two consecutive CNN layers, and a 40% dropout layer is added after each CNN and LSTM. The dropout layer [32] uses adaptive regularization to prevent the model’s overfitting and ensure that the model is in an optimal state. A comparison of the architecture of CNN-LSTM and the proposed model is shown in Fig. 2.

D. Proposed ConvLSTM Model

Unlike the structure of the CNN-LSTM stack, the LSTM replaces the pooling layer in ConvLSTM and discovers deeper relationships between time-series data through the embedding structure. ConvLSTM uses convolution computation in the fully connected layer, which means that ConvLSTM replaces the matrix multiplication of each gate in the LSTM cell with convolution operations. In other words, the parameters learned are convolution kernel weights and can be used to capture the underlying spatial features by performing convolution operations in multidimensional data. Electricity consumption data with time-series properties can be fit with sequential input in ConvLSTM. ConvLSTM has been proposed to solve regression issues using its temporal memory properties [26], and it is mainly used for forecasting with multidimensional time-series properties and spatial expansion. As shown in Fig. 3, the network structure of the ConvLSTM is a variant of the LSTM with feedforward features of input transformations and recursive transformations implemented by convolution [29].

In the proposed model, each user’s electricity consumption data can be divided into 2-D image \( M \times N \) in terms of quarters, i.e., quarterly electricity consumption data \( \times \) the number of quarters. Each image \( X \in R^{p \times M \times N} \), which has one
pixel $P$, and then, the observed feature value can be given as follows, with $R$ being the domain of the observed data [26].

The 2-D data can be seen as the spatial dimension of the image, and features are captured mainly by the convolutional layer and 3-D has an extra-temporal dimension, and the temporal features are captured by the LSTM, as shown in Fig. 4. The internal structure of the ConvLSTM, $C$, and $X$ represents the unit output, $H$ is the hidden state, and the example uses the structure convolution operation present and new values, as shown in Fig. 5.

In contrast to CNN-LSTM, $H$ and $X$ in ConvLSTM use convolutional operations instead of parametric matrix multiplication, and the learned parameters are the weights of the convolution kernel. * denotes the convolutional operation, which can have multiple convolutional filters. The main equations can be derived as follows [26]:

$$i_t = \sigma(W_{xi} \ast X_t + W_{hi} \ast H_{t-1} + W_{ci} \circ C_{t-1} + b_i)$$ (9)

$$f_t = \sigma(W_{xf} \ast X_t + W_{hf} \ast H_{t-1} + W_{cf} \circ C_{t-1} + b_f)$$ (10)

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} \ast X_t + W_{hc} \ast H_{t-1} + b_c)$$ (11)

$$o_t = \sigma(W_{xo} \ast X_t + W_{ho} \ast H_{t-1} + W_{co} \circ C_t + b_o)$$ (12)

$$H_t = o_t \circ \tanh(C_t).$$ (13)

Even without the multichannel features, the proposed model can still dig deeper than the baseline model in the convolutional operation into the temporal features between the number of electricity consumption data. As shown in Fig. 2(b), the structure of the proposed model becomes clearer and more concise after embedding batch normalization. A point worth noting is that the output still needs to be flattened to a long vector before the dense layer can be interpreted.

## III. EXPERIMENTAL SETUP

In this article, all experiments are based on Python (Version 3.7.6) programming, in which the deep learning framework is based on TensorFlow (Version 2.4.0). The hardware platform is a laptop computer, the processor is 2.6-GHz six-core Intel Core I7, and the graphics are AMD Radeon Pro 5300M 4 GB, meanwhile, with the support of free cloud GPU, 30-GB RAM, and eight CPUs.

### A. Data Description

1) Preview of Raw Dataset: The dataset selected for this article is obtained from real electricity consumption data published by the State Grid Corporation of China (SGCC) [21]. The dataset contains the daily electricity consumption in kilowatt-hour (kWh) of 42372 customers between January 1, 2014 and October 31, 2016 (1034 days); 38757 of these customers are normal electricity users (labeled 0) and 3615 are customers who have been identified as electricity thieves (labeled 1). An example of the dataset is presented in Table I.

2) Electricity Consumption Data Visualization: The anomalous manifestations of electricity theft are not only present on the surface of the data, but the underlying patterns and trends are also equally characteristic. While machine learning can replace manual detection of potential electricity theft, we also need to communicate with the data, which is what data analysis is all about. As shown in Fig. 6, electricity usage data for normal customers tend to be more stable and less volatile in months other than summer, with July, August, and September showing significantly stronger usage and fluctuations than other months. However, the data for electricity theft customers appear unusually chaotic, with a very large drop in December and an overall trend that does not conform to natural patterns.

In [21], data from four weeks can be extracted for further analysis, for example, plotting the data between the two types of users again and plotting PCC. These methods show correlations and potential regularities between the data in each of the two categories of users. The two categories of users can first be plotted again weekly, as shown in Fig. 7. Normally, normal users show good cyclical and seasonal patterns, but electricity theft users continue to have mixed chaotic electricity usage characteristics. Thus, annual, quarterly, monthly, weekly, and daily electricity usage characteristics can be used as a benchmark for extracting features. Fig. 8 clearly shows that the data correlation of normal users is much stronger than electricity theft users. The correlation coefficient for electricity theft customers does not exceed a maximum of 0.3 and has a negative correlation. However, the correlation coefficient for normal customers is generally higher than 0.8 and shows a strong positive correlation. In the PCC, values above 0.5 or below −0.5 represent a relatively significant correlation. Positive values closer to 1 indicate a stronger direct correlation.
Fig. 6. Average monthly electricity consumption in 2015. (a) Normal energy users. (b) Energy theft users.

Fig. 7. Average daily electricity consumption every four weeks. (a) Normal energy users. (b) Energy theft users.

A negative value closer to $-1$ represents a strong indirect correlation [33].

B. Data Preparation

1) Missing Data Filtering and Imputation: This article sets a rejection baseline of 3%, i.e., users with more than one month of missing data will be removed. The minimum threshold of missing data is also used for monthly data to retain certain characteristics of the raw data. The dataset used for this article involves continuous missing data. Therefore, the KNN imputation is a more efficient method. Its algorithm is based on similarity and relies on a distance metric, the default of which is the Euclidean distance metric. KNN imputation is effective for handling missing values in continuous and ordered data, and its imputation accuracy and reduction of statistical errors are typically better than 1NN (e.g., two neighboring data) [34]. The main point is that the imputed values are the actual values that occur rather than the constructed values, which also allows for better preservation of the original data structure. In this article, the number of nearest neighbors $k$ is selected as 5.

2) Outlier Processing: To maintain a true state of the electricity consumption data, we use boxplots to screen for outliers. It uses the quartiles of the data to identify outliers among them. The boxplot shows the distribution of data based on a summary of five numbers (minimum, first quartile (Q1), median, third quartile (Q3), and maximum), and the maximum value is $Q3 + 1.5 \times \text{IQR}$ and the minimum value $Q1 - 1.5 \times \text{IQR}$. However, due to the characteristic nature of electricity consumption data, the statistics can be conducted in such a way as to create “false” outliers. $Q3 + 1.5 \times \text{IQR}$/Q1 $- 1.5 \times \text{IQR}$ can be defined as a minor or moderate outlier, and $Q3 + 3 \times \text{IQR}/Q1, - 3 \times \text{IQR}$ can be defined as an extreme outlier [35]. By screening the minor and extreme outliers for all users over 1034 days, a partial sample is shown in Fig. 9.

3) Imbalanced Classification Sorting: Borderline-SMOTE is an improvement on SMOTE. For example, the overlap between the minority and majority classes in the raw dataset or
statistical observations of electricity data is possible. SMOTE may confuse the two classes of data, resulting in inaccurate classification data. However, borderline-SMOTE will classify observations in this minority class as noise points when the data adjacent to the minority class are all in the majority class and ignore them when generating the data [36]. It is equivalent to creating boundaries in the vicinity of some outliers, which is more conducive to the accuracy of the generated data. Fig. 10 compares the theft users generated by borderline-SMOTE and the original data (for the training and validation datasets only), with the trend matching the real theft user’s status.

C. Hyperparameter

To ensure the reliability and authenticity of the experimental evaluation, the activation functions and main hyperparameters of the three models are maximally kept consistent. More specifically, to further validate the superiority of the ConvLSTM ontology in mining time-series depth features, we intentionally align the training parameters of the CNN-LSTM with the ConvLSTM, as shown in Table II. In this article, the output layer activation function is sigmoid [25], which is more stable for scaled data. Sigmoid is suitable for this article’s binary classification prediction output (0 represents the label of normal users and 1 represents the label of electricity theft users) because it exists between 0 and 1.

\[ S(x) = \frac{1}{1 + e^{-x}}. \]  

The other activation function is ReLU [25], which trains the model faster, ensures near-global weight optimization, and increases the nonlinearity of the network. It is more conducive to backpropagation and avoids gradient explosion or vanishing issues

\[ f(x_i) = \max(0, x_i). \]  

The optimizer is Adam (an initial learning rate is 0.001) [37], an optimization algorithm that can replace the classical stochastic gradient descent method to update the network weights in the training data iteratively. In short, Adam can adaptively adjust each network’s learning rate. Furthermore, the logarithmic loss is also the first to deal with binary classification issues, namely, the binary_crossentropy in Keras. Binary cross entropy compares each predicted probability with the actual category output, which can be either 0 or 1. It then calculates a score that penalizes the probability based on the distance from the expected value. To ensure training and modeling efficiency, “early stopping” is applied to the model so that the network could be better generalized. The dropout layer is only applied to MLP and CNN-LSTM, as the proposed ConvLSTM model did not show overfitting performance during the experiments.

D. Validation Strategy

This article selects the optimal model during the training and validation process and is applied to a separate dataset for testing. Unlike the traditional training-to-test process, this approach increases the test results’ stability, optimally, and reliability. K-fold cross validation is applied to the validation process, where the training-validation dataset is divided into K copies, with K − 1 copies used as training data and one copy as validation data on a rotating basis [38].

To further ensure the impartiality of the test results, three validation strategies are used simultaneously for the three models, i.e., the classical tenfold and fivefold cross validation and no cross validation. A total of nine test results are used for the final comparison and analysis.

IV. RESULT

A. Metrics Description

For model evaluation, comprehensive performance metrics were used on the test dataset (20%) to validate the accuracy, reliability, and robustness of the proposed model in identifying electricity theft users.

1) Loss (Binary Cross Entropy/Log Loss): The loss function can be used to judge the predicted outcome of a classification model, i.e., the difference between the predicted value and the actual value. The loss function in binary classification is the binary cross entropy, where \( y_i \) represents 0 or 1 in the label. The larger the prediction deviation, the higher the log-loss value. The equation is shown as follows:

\[ \text{Loss} = \frac{1}{N} \sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)). \]  

2) Confusion Matrix: The confusion matrix is a technique used to summarize the performance of a classification algorithm by providing a more intuitive indication of the correctness and error types of the model [39]. It also reduces false positives (FPs) and increases true positives (TPs) to ensure that the model efficiently fits a real-world usage scenario.
TP refers to the number of theft users correctly classified in this article. True negative (TN) is the number of normal users correctly classified. FP is the number of normal users incorrectly predicted as theft users. False negative (FN) is the number of theft users incorrectly predicted as normal users. A typical confusion matrix is shown in Table III.

Accuracy is a straightforward and meaningful metric in a state where the number of dataset classes is balanced. It refers to the frequency of correct predictions out of all predictions made by the model

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FN} + \text{TP} + \text{FP}}.$$  

(17)

Precision indicates the ability of the model to correctly predict positives from all positive predictions, whereas recall indicates the ability of the model to correctly predict positives from actual positive samples, i.e., representing the classification accuracy of theft users and actual theft users

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$  

(18)

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$  

(19)

F1-score gives equal weight to precision and recall and captures trends in them. It is an important measure of a classification model in the presence of FPs and FNs

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

(20)

Cohen’s kappa can be used to judge the strength of the model’s classification predictions. The kappa value CK is a metric for comparing the observed accuracy with the expected accuracy

$$\text{CK} = \frac{2 \times (\text{TP} \times \text{TN} - \text{FN} \times \text{FP})}{(\text{TP} + \text{FP})(\text{FP} + \text{TN})(\text{TP} + \text{FN})(\text{FN} + \text{TN})}.$$  

(21)

The ROC curve is composed of TP and FP, and the area under the ROC curve (ROC-AUC) is the area of the ROC curve and FP. The larger the ROC-AUC, the better the classification ability of the model. The PR curve consists of precision and recall and is evaluated similar to the ROC-AUC. The ROC-AUC is independent of threshold selection and reflects the characteristics of the model, while the PR-AUC can be considered as the average of the precision calculated for each recall threshold. The focus of the PR curve on the minority class makes it an effective diagnostic for imbalanced binary classification models.

### B. Case Study

In this section, the performance of the proposed ConvLSTM-based ETD method is evaluated by comparing with related deep-learning-based ETD models (MLP and CNN-LSTM models). Three cross validation methods as introduced in Section III-D are applied for each model. The ETD performance is shown in Table IV. From the table, the proposed ConvLSTM model outperforms other models in prediction. In particular, the ConvLSTM with tenfold cross validation shows the highest values of almost all metrics, expect for loss. The result demonstrates that K-fold cross validation can optimize the deep neural network model and reduces the overfitting at the same time. It is also noticed that the benchmark model, MLP, shows the worst performance in all cases. MLP model has the simplest structure without the memory cell to store the historical information. Hence, MLP is inefficient in time-series tasks such as the ETD in this article.

The PR and ROC curves for the proposed ConvLSTM ETD method and other related ETD methods are plotted in Figs. 11 and 12. The PR curve is a plot of the precision (y-axis) and the recall (x-axis) for different probability thresholds and a model with perfect skill is depicted as a point at a coordinate of (1, 1); a no-skill classifier will be a horizontal line with value 0.5 on the plot (blue with dashes in Fig. 11). In turns of the ROC curve, it plots FP rate versus TP rate, and a point in the top left of the plot indicates a perfect prediction, while the no skill classifier is presented as a diagonal line (blue with dashes in Fig. 12). The ConvLSTM with tenfold cross validation has the best performance in terms of both

| True Class | Negative (normal user) | Positive (theft user) |
|-----------|------------------------|----------------------|
| Predicted Class | Negative (normal user) | Positive (theft user) |
| TN | FN | FP | TP |

### TABLE III

**Typical Binary Classification Confusion Matrix**

Fig. 11. PR curves for the proposed ConvLSTM ETD method and other related ETD methods.

Fig. 12. ROC curves for the proposed ConvLSTM ETD method and other related ETD methods.
TABLE IV
ETD PERFORMANCE

| Models               | Accuracy | Loss  | Precision | Recall | F1-score | CK    | ROC-AUC | PR-AUC |
|----------------------|----------|-------|-----------|--------|----------|-------|---------|--------|
| MLP with 10-Fold     | 0.915    | 0.336 | 0.95      | 0.876  | 0.911    | 0.83  | 0.959   | 0.963  |
| CNN-LSTM with 10-Fold| 0.957    | 0.204 | 0.967     | 0.947  | 0.957    | 0.915 | 0.976   | 0.982  |
| ConvLSTM with 10-Fold| 0.966    | 0.23  | 0.984     | 0.948  | 0.966    | 0.932 | 0.977   | 0.98   |
| MLP with 5-Fold      | 0.896    | 0.315 | 0.949     | 0.838  | 0.89     | 0.7921| 0.93    | 0.957  |
| CNN-LSTM with 5-Fold | 0.944    | 0.201 | 0.972     | 0.914  | 0.942    | 0.887 | 0.972   | 0.978  |
| ConvLSTM with 5-Fold | 0.958    | 0.288 | 0.975     | 0.947  | 0.957    | 0.916 | 0.974   | 0.974  |
| MLP without K-Fold   | 0.891    | 0.363 | 0.92      | 0.856  | 0.887    | 0.792 | 0.944   | 0.947  |
| CNN-LSTM without K-Fold | 0.941  | 0.202 | 0.958     | 0.922  | 0.94     | 0.969 | 0.969   | 0.974  |
| ConvLSTM without K-Fold | 0.942  | 0.221 | 0.972     | 0.911  | 0.94     | 0.884 | 0.974   | 0.978  |

ROC and PR curves. It has the largest AUC (a value close to 1), which indicates that the performance of ConvLSTM is very close to the perfect point. It is also observed that CNN-LSTM with tenfold cross validation has similar performance as ConvLSTM model on the ROC and PR curves. In TP and FP recognition, it can achieve the prediction accuracies of 47.37% and 1.63%, respectively, which is close to the performance of ConvLSTM with tenfold. This indicates that CNN-LSTM with tenfold also has a strong ability to identify and discriminate against normal users but is not as good as ConvLSTM in predicting electricity theft users.

Fig. 13 shows the loss and accuracy of the three models by taking tenfold validation strategy as examples. ConvLSTM performs the best in generalization ability and convergence efficiency. As shown in Fig. 13, its generalization gap is around 0.1. In addition, ConvLSTM only requires around 60 epochs to converge to the best model state, while MLP and CNN require 150 and 125 epochs, respectively. MLP and CNN-LSTM reach smooth convergence at around 100 and 80 epochs, respectively. Furthermore, ConvLSTM still outperforms the other two models in noise control without dropout. This result demonstrates that ConvLSTM has strong predictive robustness, and its model structure can effectively avoid overfitting.

V. CONCLUSION

In this article, a hybrid ConvLSTM ETD method is proposed. The proposed method combines ConvLSTM and a batch normalization to improve the flexibility of the training and testing phases. Moreover, the ETD model utilizes borderline-SMOTE to generate synthetic energy theft samples to better solve the imbalanced classification problem. From the simulation, the ConvLSTM shows excellent identification of electricity theft users under all three validation strategies and has significant advantages in model robustness, convergence efficiency, and generalization ability. This article also demonstrates that the model with tenfold cross validation outperforms the models with the fivefold cross-validation and no cross-validation methods.

The extension of multidimensional electricity usage data is an issue to consider in future work. Multiple features can be added to the dataset to better match the input of a multidimensional tensor, e.g., weather and geographical location. This is also a way for the ETD model to incorporate objective factors to detect potential electricity thieves.

ACKNOWLEDGMENT

For the purpose of open access, the author(s) has applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding authors upon reasonable request.

REFERENCES

[1] X. Fang, S. Misra, G. Xue, and D. Yang, “Smart grid—The new and improved power grid: A survey,” IEEE Commun. Surv. Tuts., vol. 14, no. 4, pp. 944–980, 4th Quart., 2012.
[2] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato, “A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid,” IEEE Trans. Instrum. Meas., vol. 64, no. 8, pp. 2072–2085, Aug. 2015.
[3] L. Peretto, “The role of measurements in the smart grid era,” IEEE Instrum. Meas. Mag., vol. 13, no. 3, pp. 22–25, Jun. 2010.
Z. Yan and H. Wen, “Performance analysis of electricity theft detection for the smart grid: An overview,” IEEE Trans. Instrum. Meas., vol. 71, pp. 1–28, 2022.

S. S. S. R. Depuru, L. Wang, and V. Devabhaktuni, “Electricity theft: Overview, issues, prevention and a smart meter based approach to control theft,” Energy Policy, vol. 39, pp. 1007–1015, Feb. 2011.

G. Chandra-Welch and L. Beshiras, “Smart grids in emerging markets-private sector perspectives,” Nat. Renew. Energy Lab (NREL), Golden, CO, USA, Tech. Rep. NREL/TP-74A0-77546, 2020.

G. M. M. Messinis and N. D. Hatziargyriou, “Review of non-technical loss detection methods,” Electric Power Syst. Res., vol. 158, pp. 250–266, May 2018.

S. Weeks, N. Gonzalez, J. Tani, T. De Rybel, and J. Driesen, “Parameter identification of unknown radial grids for theft detection,” in Proc. 3rd IEEE PES Innov. Smart Grid Technol. Eur. (ISGT Europe), Oct. 2012, pp. 1–6.

P. Kadurek, J. Blom, J. F. G. Cobben, and W. L. Kling, “Theft detection and smart metering practices and expectations in The Netherlands,” in Proc. IEEE PES Innov. Smart Grid Technol. Conf. Eur. (ISGT Eur.), Oct. 2010, pp. 1–6.

Energy Analytics for Development: Big Data for Energy Access, Energy Efficiency, and Renewable Energy, Energy Sector Management Assistance Program, World Bank, Washington, DC, USA, 2017.

B. Camarota, R. Gavaldà, S. Alciverco, and V. Martin, “Fraud detection in energy consumption: A supervised approach,” in Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA), Oct. 2016, pp. 120–129.

Z. Yan and H. Wen, “Electricity theft detection base on extreme gradient boosting in AML,” IEEE Trans. Instrum. Meas., vol. 70, pp. 1–9, 2021.

R. R. Bhate, R. D. Trevizan, R. Sengupta, X. Li, and A. Bezas, “Identifying nontechnical power loss via spatial and temporal deep learning,” in Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2016, pp. 272–279.

Y. LeCun, Y. Bengio, and G. E. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, Dec. 2015.

M. K anche and S. Shirmohammadi, “Applied AI in instrumentation and measurement: The deep learning revolution,” IEEE Instrum. Meas. Mag., vol. 23, no. 6, pp. 10–17, Sep. 2020.

G. Dorffner, “Neural networks for time series processing,” in Neural Network World. Princeton, NJ, USA: Citeseer, 1996.

S. Alhwai, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in Proc. Int. Conf. Eng. Technol. (ICET), Aug. 2017, pp. 1–6.

S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in Proc. Int. Conf. Mach. Learn., 2013, pp. 1310–1318.

M. N. Hasan, R. N. Toma, A.-A. Nahid, M. M. M. Islam, and J.-M. Kim, “Electricity theft detection in smart grid systems: A CNN-LSTM based approach,” Energies, vol. 12, no. 17, p. 3310, 2019.

J. D. Rodriguez, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, “Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids,” IEEE Trans. Ind. Informat., vol. 12, no. 4, pp. 1606–1615, Apr. 2018.

R. U. Madhure, R. Raman, and S. K. Singh, “CNN-LSTM based electricity theft detector in advanced metering infrastructure,” in Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), Jul. 2020, pp. 1–6.

Z. Yang, S. Kuenzel, N. Colombo, and C. Watkins, “Hybrid short-term load forecasting method based on empirical wavelet transform and bidirectional long short-term memory neural networks,” J. Modern Power Syst. Clean Energy, early access, Aug. 9, 2022, doi: 10.35833/MPCIE.2021.000276.

X. Zhang, C. Watkins, and S. Kuenzel, “Multi-quantile recurrent neural network for feeder-level probabilistic energy disaggregation considering rooftop solar energy,” Eng. Appl. Artif. Intell., vol. 110, Apr. 2022, Art. no. 104707.

S. Sharma, S. Sharma, and A. Athiaya, “Activation functions in neural networks,” Towards Data Sci., vol. 6, no. 2, pp. 310–316, 2017.

X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28, 2015, pp. 1–12.

Y. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep convolutional neural networks on multichannel time series for human activity recognition,” in Proc. 24th Int. Joint Conf. Artif. Intell., 2015, pp. 3995–4001.

S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. Int. Conf. Mach. Learn., 2015, pp. 448–456.

T. T. Verlekar and A. Bernardino, “Video based fire detection using Xception and Conv-LSTM,” in Proc. Int. Symp. Vis. Comput. Cham, Switzerland: Springer, 2020, pp. 277–285.

R. Mutageki and D. S. R. Kamara, “A CNN-LSTM approach to human activity recognition,” in Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAHCI), Feb. 2020, pp. 362–366.

A. Graves and J. Schmidhuber, “Fraysimly phoneme classification with bidirectional LSTM and other neural network architectures,” Neural Netw., vol. 18, no. 5, pp. 602–610, 2005.

S. Wager, S. Wang, and P. S. Liang, “Dropout training as adaptive regularization,” in Proc. Adv. Neural Inf. Process. Syst., vol. 26, 2013, pp. 1–11.

G. Corder and D. Foreman, Nonparametric Statistics. A Step-By-Step Approach, 2nd ed. 2014.

M. Kuhn et al., Applied Predictive Modeling, vol. 26. Cham, Switzerland: Springer, 2013.

R. Dawson, “How significant is a boxplot outlier?” J. Statist. Educ., vol. 19, no. 2, pp. 1–14, Jul. 2011.

H. Han, W.-Y. Wang, and B.-H. Mao, “Borderline-smote: A new oversampling method in imbalanced data sets learning,” in Proc. Int. Conf. Intell. Comput. Cham, Switzerland: Springer, 2005, pp. 878–887.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv, 2014, arXiv:1412.6980.

J. D. Rodriguez, A. Perez, and J. A. Lozano, “Sensitivity analysis of k-fold cross validation in prediction error estimation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 3, pp. 569–575, Mar. 2010.

P. Branco, L. Torgo, and R. P. Ribeiro, “A survey of predictive modeling on imbalanced domains,” ACM Comput. Surv., vol. 49, no. 2, pp. 1–50, 2016.

Hong-Xin Gao (Student Member, IEEE) received the B.B.A. and B.Eng. degrees from Shandong Agricultural University, Tai’an, Shandong, China, in 2017 and the M.S. degree (Hons.) in information security from the Royal Holloway, University of London, London, U.K., in 2021.

He worked for a state-owned electric power enterprise in China from November 2011 to May 2019. During this time, he worked in production, bidding, and project management positions, mainly responsible for project operations for power smart terminals.

Stefanie Kuenzel (Senior Member, IEEE) received the M.Eng. and Ph.D. degrees from Imperial College, London, U.K., in 2010 and 2014, respectively.

She is currently the Head of the Power Systems Group and a Senior Lecturer with the Department of Electronic Engineering, Royal Holloway, University of London, London, U.K., in 2021.

His current research area is deep learning for anomaly detection in advanced smart grids.

Xiao-Yu Zhang (Member, IEEE) received the B.Eng. degree from North China Electric Power University, Beijing, China, in 2016, the M.S. degree (Hons.) in electrical power system from the University of Birmingham, Birmingham, U.K., in 2017, and the Ph.D. degree in electrical engineering from the Royal Holloway, University of London, London, U.K., in 2022.

He is currently a Lecturer with the School of Artificial Intelligence, Anhui University, Hefei, China. His research interests include deep learning technology and data analytics in smart grids, smart grid privacy and security, and demand-side management.