Energy Theft Detection With Energy Privacy Preservation in the Smart Grid

Donghuan Yao, Mi Wen, *Member, IEEE*, Xiaohui Liang, *Member, IEEE*,
Zipeng Fu, Kai Zhang, and Baojia Yang

***Abstract*—As a prominent early instance of the Internet of Things in the smart grid, the advanced metering infrastructure (AMI) provides real-time information from smart meters to both grid operators and customers, exploiting the full potential of demand response. However, the newly collected information without security protection can be maliciously altered and result in huge loss. In this paper, we propose an energy theft detection scheme with energy privacy preservation in the smart grid. Especially, we use combined convolutional neural networks (CNNs) to detect abnormal behavior of the metering data from a long-period pattern observation. In addition, we employ Paillier algorithm to protect the energy privacy. In other words, the users' energy data are securely protected in the transmission and the data disclosure is minimized. Our security analysis demonstrates that in our scheme data privacy and authentication are both achieved. Experimental results illustrate that our modified CNN model can effectively detect abnormal behaviors at an accuracy up to 92.67%.**

***Index Terms*—Convolutional neural network (CNN), energy theft, privacy preserving, smart grid.**

I. Introduction

THE INTERNET of Things (IoT) and artificial intelligence (AI) are two cornerstone technologies enabling smart cities, and have been interacting with each other into an organic ecosystem. In the smart grid, smart meters and various sensors are widely used to increase the two-way communication capability. Combined with the advanced metering infrastructure (AMI), they enable energy companies to obtain real-time voltage, current, active power, reactive power, energy usage, and other measurements from the smart meters deployed at user homes [1], [2]. Recently, smart meters are shown to be vulnerable to cyber physical attacks in the smart grid due to their insecure and distributed network and physical environment [3]–[5]. One serious threat is energy theft attacks, which cost more than $25 billion every year to the energy companies [6]. Such an attack aims to pay less by attacking user meters to tamper with the energy usage sent to energy company. Another severe threat is privacy violation. As smart meters collect real-time energy usage that may reveal user's habits and behavior at home, the user privacy concern will be raised if the collected data is not well protected [7]. For example, if the user's daily energy consumption is low, it may imply that the user is not at home [8]. Thus, such privacy-sensitive information must be protected from unauthorized access. To disclose the usage for theft detection and to hide the usage for privacy preservation are conflicting goals. We aim to address both theft detection and privacy preservation in this paper.

A number of works have been conducted for energy theft detection in the smart grid. Some used the classification-based support vector machine (SVM) technique to classify the normal and attack samples from the energy usage database [9]–[11]. In addition, matrix decomposition [12], linear regression [13], and state estimation [14] can be used to analyze the data for energy theft detection. However, these approaches cannot be applied to cases with massive amounts of data. Zheng *et al.* [15] proposed a wide and deep convolutional neural network model to analyze energy theft behavior of individual users. In this paper, we additionally study the energy theft behavior from a user group perspective, i.e., a group of users may exhibit similar energy consumption patterns due to local activities for a certain period of time. We plan to exploit this behavior characteristic to more accurately detect the sophisticated attacker.

Most theft detection schemes require the access of the original smart meter data that are highly user privacy-sensitive. Although privacy-preserving techniques have been introduced in the smart grid communication [16]–[18], they are rarely proposed in the context of theft detection. One work is developed under an assumption that the normal energy output of a photovoltaic device is similar to that from a geographical region [19]. With the homomorphic encryption technique, the calculation of the distance of two vectors is conducted while the vectors (energy data) are not disclosed to unauthorized entities. However, the proposed work detects energy theft from the perspective of generators, it cannot solve the diversity of theft. For example, if a user's meter is tampered

Manuscript received September 15, 2018; revised January 13, 2019; accepted February 18, 2019. Date of publication March 6, 2019; date of current version October 8, 2019. This work was supported by the National Natural Science Foundation of China under Grant 61872230, Grant 61572311, Grant 61802248, and Grant 61702321, by the National Science Foundation of the USA under Grant 1618893, by the Dawn Program of Shanghai Education Commission under Grant 16SG47, by the Chen Guang project supported by the Shanghai Municipal Education Commission under Grant 18CG62. *(Corresponding author: Mi Wen.)*

D. Yao, M. Wen, and K. Zhang are with the College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 201101, China (e-mail: dhyao1@mail.shiep.edu.cn; miwen@shiep.edu.cn; kzhang@shiep.edu.cn).

X. Liang is with the Computer Science Department, University of Massachusetts Boston, Boston, MA 02125 USA (e-mail: Xiaohui.Liang@umb.edu).

Z. Fu is with the Department of Computer Science, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: fu-zipeng@engineering.ucla.edu).

B. Yang is with the Department of Microsoft Suzhou Technology Center, Microsoft, Suzhou 215123, China (e-mail: ybaojia91@gmail.com).

Digital Object Identifier 10.1109/JIOT.2019.2903312

2327-4662 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

with usage by external illegal attack, it cannot be detect. In addition, Salinas and Li [20] proposed a privacy-preserving state estimation scheme based on two loosely coupled filters to detect energy theft attacks and achieve privacy preservation. But it is not conformed the actual grid operation, because it protects privacy by sending residual rather than user usage, so the smart grid cannot be dispatched and paid for bills.

In this paper, we propose an energy theft detection scheme with energy privacy preservation in the smart grid to ensure user privacy and realize the detection of theft. Specifically, we employ the combined convolutional neural networks (CNNs) for analyzing the reported usage data and detecting the fake data. To our best of knowledge, only [15] and ours use CNN to detect theft. Utilizing the homomorphic encryption technique, we can protect the energy usage in the transmission and further enable the gateway (GW) to aggregate the authentic user energy usage without accessing any original usage data. In addition, the control center (CC) can only access the sum of the authentic usage data and the number of users who honestly report their usage data. The CC is unable to access the original energy usage data of individual users, which are highly privacy-sensitive. The main contributions of this paper are threefold.

1) We build a CNNs model for detecting the abnormal theft behavior based on the similarity of the users energy consuming behavior in a local user group. The use of the user group data helps overcome the data incompleteness problem, address more sophisticated theft detection problem, and eventually increases the detection accuracy.
2) We realize the dispatching of smart grid under the premise of protecting users' privacy, where we utilize the homomorphic encryption to achieve privacy-preserving data aggregation and efficient smart grid communications.
3) We provide a comprehensive security analysis to show that the proposed scheme achieves the desired security property. In addition, we conduct extensive experiments on massive realistic energy usage dataset. The experimental results show that our proposed combined CNN model outperforms other existing approaches in terms of accuracy.

The remainder of this paper is organized as follows. After related work in Section II, we introduce system model, system design goal, and system security requirements in Section III. In Section IV, we review the relevant knowledge. Section V presents our proposed scheme. We give security analysis in Section VI, while Section VII gives the experimental results. Finally, we conclude this paper.

II. Related Work

This section discusses related work in two categories: 1) energy theft detection and 2) privacy preserving with data aggregation.

A. Energy Theft Detection

Some works have been conducted to investigate the energy theft problem in the smart grid, where existing technique can be generally classified into three categories: 1) state estimation; 2) game theory; and 3) machine learning. The classic state estimation-based solutions [14], [21]–[23] usually introduced some integrated distribution state estimation tricks to realize; while the game theory-based method is considered to be a new way to detect energy theft in energy-theft issues [24], [25]. Previous works have investigated the prevention and detection of attacks by using classification-based detection technique, such as SVM. Pereira *et al.* [26] introduced an algorithm called social-spider optimization for feature selection purposes. Feature selection, tuning parameters and feature selection+tuning parameters, are chosen as model selection. However, most of these studies are less accurate in energy theft detection and require artificial feature extraction according to domain knowledge.

B. Privacy Preserving With Data Aggregation

Recently, a number of works focused on data aggregation to preserve the privacy of users information in the smart grid [27], [28]. It is assumed that the aggregate usage data provides enough information to the entity without exposing the individual user's information privacy, that is, the entity can only know the whole data rather than the personal data. Bao and Lu [29] proposed a differentially private data aggregation scheme for aggregating smart meter measurements. In specific, every smart meter reports an encrypted data onto the GW, then, the GW aggregates all the reported data and sends the aggregated value to the CC. The CC decrypts the aggregated value to get the summation of all smart meter readings.

Different from previously privacy-preserving theft detection schemes, the set aggregation in the smart grid communications of our proposed scheme enables the CC to obtain not only the whole aggregated energy usage but also the number of users who honestly report their usage data. With this kind of set aggregation, the control server can carry out more accurate data analysis for monitoring and controlling the smart grid.

III. System Model, System Design Goal, and System Security Requirements

In this section, we formalize the system model, system design goal, and system security requirements.

A. System Model

In this section, we discuss how users send the energy usage information to the CC/GW and a perpetrator of theft detection. As shown in Fig. 1, it includes users, local area network (LAN), CC, and trusted third party (TTP).

1) *Users:* Each user is equipped with a smart meter that connects the smart devices at home to aggregate their energy consumption [16]. Then the smart meter sends the usage to the energy utility via the GW for data analysis, charging, and reasonable energy dispatching.
2) *LAN:* It is a collection of users in a certain area. The LAN is a server with memory cells, processing units, and GWs [30]. Its GW serves as a relay and aggregator role in the system. The server GW (SG) is a detection server with processing units, which used to energy theft

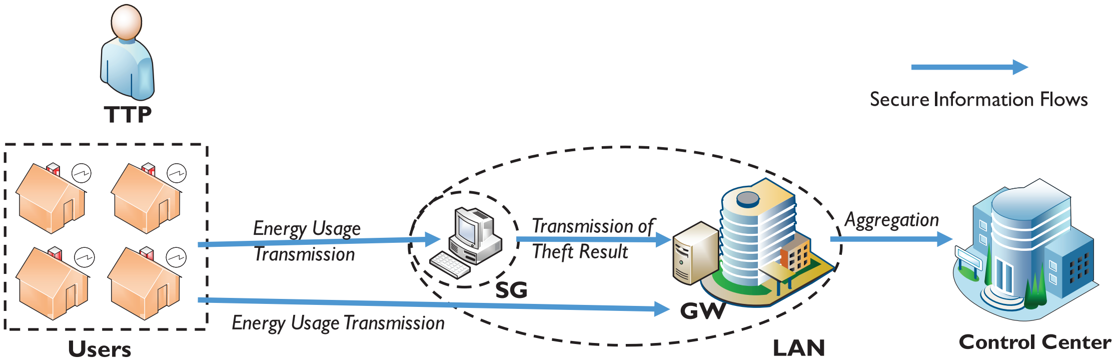

TTP
Secure Information Flows
Users
Energy Usage Transmission
SG
Transmission of Theft Result
GW
LAN
Aggregation
Control Center
Energy Usage Transmission

Fig. 1. System model.

detection (we think it is trusted). The LAN connects involved users and the CC in the smart grid.

3) *CC:* The CC is the core entity in the energy company, who is responsible for processing and analyzing the information from users. It considers the LAN as a unit and does not know the details of each user under the LAN.

4) *TTP:* The TTP is a key generation party, which issues keys to other entities; it also issues a unique ID for each user, GW and SG and these IDs are stored in a secure place. Assume that TTP is trusted by all entities and would not be compromised.

We believe that smart grid contains local area networks LAN$s = \{\text{LAN}_1, \text{LAN}_2, \ldots, \text{LAN}_m\}$. These LANs engage in two-way communications with the smart meter network, perform aggregation, and authentication operations to ensure data authenticity and integrity [31]. Simultaneously, each LAN contains users $U_s = \{U_1, U_2, \ldots, U_i, \ldots, U_w\}$, where assuming w is most 100 to alleviate the load on the LAN server. This paper leverages the measurements and communication capabilities of smart meters to detect energy thieves in a privacy-preserving manner.

B. System Design Goal

Energy theft is a criminal behavior in the smart grid that manipulates the output of a smart meter. If an illegal user is able to operate a meter, he can attack the meter and tamper with the amount of energy sent to the LAN. The purpose of the dishonest user is to reduce his own energy bills by reducing the energy usage. In this case, the illegal user may tamper all or some of the functionalities of the home-level meter, which is easy to launch and difficult to detect. Hence, the system we proposed aims to *detect energy theft* through users' energy usage pattern at user sides, that is, the behavior of stealing energy should be successfully and effectively detected, while still be expected to be realized in a privacy-preserving manner.

C. System Security Requirements

In our system under consideration, the CC and GW are honest but curious, that is, they do not change users' energy usage during communication, but they are curious about the specific electrical information of each user. However, the adversary in the region is malicious, namely, actively eavesdrop on communication between different departments, modify communication information, or launch replay attacks. Therefore, our security requirements are as follows.

1) *Data Privacy:* Users' private information is not revealed to the adversary; CC should knows nothing about the details of individual user's usage.

2) *Data Information Confidentiality:* The user's energy usage and bills should be protected against any adversary. Even if an adversary eavesdrops on data transmission links, no useful information can be extracted from them. Additionally, if the adversary steals the data from LANs' and/or CC's databases, it cannot identify each users data, either.

3) *Data Integrity and Authentication:* If an adversary tries to resend or modify data, these malicious behaviors should be detected to ensure the integrity of data. In addition, the data should ensure that any unauthorized access or modification is detected, which means that adversaries cannot invade or falsify data within the LAN. Meanwhile, only the correct reports can be received.

IV. Background Knowledge

This section reviews the convolution neural network technology that used to detect abnormal behavior of the metering data and the Paillier homomorphic algorithm that used to protect data privacy.

A. Convolution Neural Network

In the research of image recognition, including the competition of authoritative ImageNet, the top algorithms in the list are all from CNN, such as VGG, ResNet, etc. In particular, CNN algorithm plays an important role in data processing in matrix form.

1) Structure of CNN: A CNN architecture is established by a stack of distinct layers that transform the input volume into an output volume (e.g., holding the class scores) through a differentiable function. Fig. 2 shows that a CNN consists of an input, an output layer, and multiple hidden layers [32], where the hidden layers are composed up of convolutional layers, pooling layers, fully connected layers, and so on.

The *convolutional layer* consists of a set of learnable filters or kernels, which have a small receptive field but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries

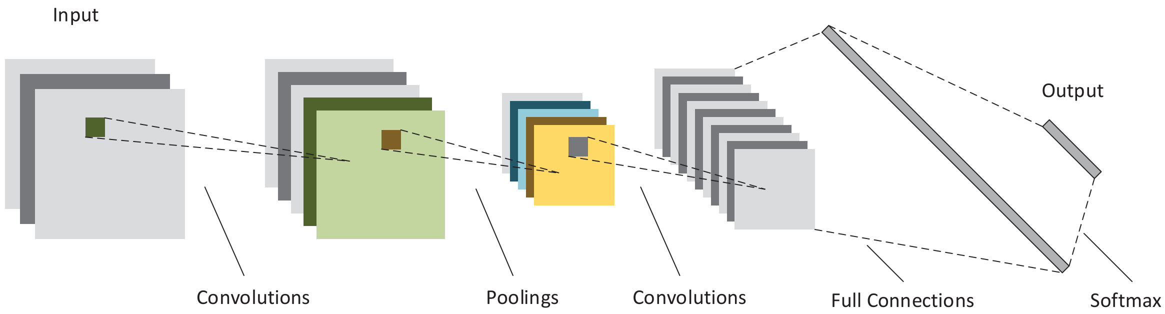


Fig. 2. CNN architecture.

of the filter and the input and producing a 2-D activation map of that filter. *The pooling layer* is a form of nonlinear downsampling and serves to progressively reduce the spatial size of the representation, to reduce the number of parameters and the amount of computation in the network, and hence to control overfitting. After several convolutional and max pooling layers, the high-level reasoning in the neural network is done via *fully connection layers*. Neurons in a fully connection layer have connections to all activations in the previous layer. The fully connection layer is used to generate the final output.

2) Activation Function: In artificial neural networks, the activation function of a node defines the output of that node given an input or set of inputs, the nonlinear activation functions allow networks to compute nontrivial problems using only a small number of nodes. Rectified linear unit (Relu) is a common activation function [33], we use it in our neural network framework. The equation of Relu performs as follows [34]:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0. \end{cases}$$

For the classified problem, the softmax function is a common one added to output layer to get category, which squashes a K-dimensional vector of arbitrary real values to a K-dimensional vector of real values where each entry is in the range $(0, 1]$, and all the entries add up to 1

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \qquad \text{for } j = 1, \ldots, K.$$

3) Loss Function and Optimizer: To train the neural network, we define loss function and optimizer to adjust the weights. We use categorical cross-entropy as loss function and stochastic gradient descent (SGD) as optimizer in our neural network framework.

The cross entropy for the distributions u and v over a given discrete set is defined as

$$H(u, v) = -\sum_{x} u(x) \log v(x).$$

SGD is an iterative method for optimizing a differentiable objective function, a stochastic approximation of gradient descent optimization [35]. The basic idea is to get "gradient" through a randomly selected data (x_i, y_i), so as to update the weight W via $W_{t+1} \leftarrow W_t + \eta\theta(-y_i W_t^T x_i)(y_i x_i)$.

B. Paillier Homomorphic Algorithm

The Paillier cryptosystem can achieve the homomorphic properties, which is widely desirable in many privacy preserving applications [8], which consists of five main parts [36].

1) Generation of Homomorphic Key: Select a security parameter κ and two large primes p and q, where $|p| = |q| = \kappa$. Compute the parameters $n = pq$ and $\lambda = \text{lcm}(p-1, q-1)$ and select the element $g \in Z_{n^2}^*$; set public key as (n, g) and private key as λ. Define the function

$$L(\phi) = (\phi - 1)/n.$$

2) Encryption: Select a random number $r_i \in Z_{n^2}^*$, the encryption operation for plaintext $c_i = E(m_i) = g^{m_i} r_i^n$ where c_i is the ciphertext of the plaintext m_i.

3) Decryption: Decrypt the ciphertext c_i into a plaintext m_i

$$m_i = D(c_i) = \frac{L\left(c_i^\lambda \bmod n^2\right)}{L\left(g^\lambda \bmod n^2\right)} \bmod n.$$

4) Aggregation: Aggregate multiple ciphertext $c_i = E(m_i) = g^{m_i} r_i^n$, which $1 \leq i \leq w$, as follows:

$$c = \prod_{i=1}^{w} c_i \bmod n^2 == \prod_{i=1}^{w} g^{m_1+m_2+\cdots+m_n} r_i^n \bmod n^2.$$

5) Decrypt the Aggregated Ciphertext: Decrypt the aggregated ciphertext as

$$m = \frac{L\left(c^\lambda \bmod n^2\right)}{L\left(g^\lambda \bmod n^2\right)} \bmod n$$

among them, $m = m_1 + m_2 + \cdots + m_w$.

V. Proposed Scheme

In this section, we introduce our energy theft detection scheme with energy privacy preservation in the smart grid. To describe this scheme, we divide the process into two parts, namely, energy privacy preservation and energy theft detection with proposed combined CNN.

Part 0: Feasibility Analysis and Preparation

This part analyzes the feasibility of theft detection with the combined CNN model.

A. Data Attributes

Two important attributes of electricity consumption behavior of users are considered: one is periodicity, that is users usually consume energy cyclically (daily or weekly) [15]; the other one is group similarity, that is users always follow some similarly patterns with others that are in a same group. For instance, the users who come from one community share similar energy consumption environments and may also produce same behaviors that may cause big valid changes on energy usage side.

B. Modeling

According to the periodicity, the obtained sequence data can be converted into the matrix form. For example, a 28 days of energy usage data can be formalized into a matrix with the shape of 4 × 7 by weekly cycle. To use group similarity, we select some reference users who come from the same group to the auxiliary input of the model as the detected target user. Therefore, the CNN model consist of two inputs: 1) target user data and 2) reference users data.

C. Training Model

A major difficulty in detecting is to obtain abnormal meter data, as is hard to manually label the data, we use the labeled database from State Grid Corporation of China (SGCC) [37] which contains the energy usage data of 42 372 customers within 1035 days. Additionally, we insert a zero value to pad of each user's energy usage sequence to make each user's data length up to 1036, then convert each data into a matrix whose shape is (148 × 7), which aims to enables the length of input data to meet the integer multiple of the cycle in CNN.

PART 1: ENERGY PRIVACY PRESERVATION

In this part, we show how to use the Paillier homomorphic encryption to protect the privacy of the energy usage data along with state information.

A. System Initialization

The system initialization inputs a security parameter (1^{λ}) and generates the public parameter $pp = (q, P, G_1, G_2, G_T, g_1, g_2, \varphi, H_{\Lambda}(\cdot), e)$, where G_1 and G_2 are two cyclic groups of the same prime order q, $P \in G$ is a generator, G_T is a multiplicative cyclic group, g_1 and g_2 are the generators of G_1 and G_2, respectively, and $\varphi(g_2) = g_1$, φ is an isomorphic mapping, $e : G_1 \times G_2 \to G_T$ is a bilinear mapping and $H_{\Lambda}(\cdot)$ is an hash function with a key.

The TTP selects a system master key $s \in Z_p^*$ and computes the system public key $y = g_2^s$, also randomly chooses $\delta, x \in Z_q^*$ and computes $e(P, P)^{\delta}$, $Y = xP$ and selects two hash functions: $H_1(\cdot) : \{0, 1\}^* \to G_1$ and $H_2(\cdot) : \{0, 1\}^* \to G_2$.

Receiving pp and a security parameter κ chosen by TTP, the CC also initializes the Paillier encryption algorithm by selecting two large prime numbers p and q with regard to κ satisfying $|p| = |q| = \kappa$, computing two parameters $n = pq$ and $\lambda = \mathrm{lcm}(p - 1, q - 1)$. Select $g \in Z_{n^2}^*$ as the generator and set the public key is (n, g) and private key is (λ) in the Paillier encryption algorithm.

B. System Registration

When to register the system, a *GW of the LAN* first chooses a random number $x_g \in Z_q^*$ as the private key, and computes the corresponding public key $Y_g = x_gP$; a SG chooses a random number $x_s \in Z_q^*$ as the private key, and computes the corresponding public key $Y_s = x_sP$; a user $i \in U$ of the LAN chooses a random number $x_i \in Z_q^*$ as the private key, and computes the corresponding public key $Y_i = x_iP$.

C. Energy Usage Transmission

As the CC needs to proceed managements and control decision toward the grid power system, in such a case that, each user should send its real-time data to the CC.

Concretely, the user i encrypts its data m_i by Paillier homomorphic encryption by choosing a random number $r_i \in Z_{n^2}^*$ and computing a ciphertext

$$c_i = E(m_i) = g^{m_i} r_i^n \mod n^2.$$

Then, the user i uses the private key x_i to generate a signature σ_i on a c_i as [38]

$$\sigma_i = x_i H(c_i \| \mathrm{LAN} \| U_i \| \mathrm{TS})$$

where TS is the current timestamp (used to resist potential replay attack). Finally, the user sends the encrypted usage data $c_i \| \mathrm{LAN} \| U_i \| \mathrm{TS} \| \sigma_i$ to both the GW and the SG.

D. Recovery of the Encrypted Energy Usage

The SG verifies the validity of $e(P, \sigma_i) = e(Y_i, H(c_i \| \mathrm{LAN} \| U_i \| \mathrm{TS}))$ and recover the corresponding usage data $m_i = D(\mathrm{c}_i)$ from the ciphertext c_i.

E. Transmission of Theft Result

The theft detection state is expressed as 1 or 0, where the abnormal data is labeled as 1, and vice the normal data is represented as 0. The SG picks $\mathrm{r}_i \in Z_{n^2}^*$ and encrypts the detection result t_i into a ciphertext

$$a_i = E(t_i) = g^{t_i} r_i^n \mod n^2$$

where t_i is 1 or 0. It uses the private key x_s under the current timestamp TS to generate a signature $\beta_i = x_s H(a_i \| \mathrm{LAN} \| \mathrm{SG} \| \mathrm{TS})$, in order to resist potential replay attack. Finally, the SG sends the encrypted detection result $a_i \| \mathrm{LAN} \| \mathrm{SG} \| \mathrm{TS} \| \beta_i$ to the GW.

F. Aggregation

Receiving the total ω encrypted energy usage data $c_i \| \mathrm{LAN} \| U_i \| \mathrm{TS} \| \sigma_i$, for $i = 1, 2, \ldots, \omega$, the local GW first checks the time stamp TS and the signature σ_i to verify its validity by $e(P, \sigma_i) = e(Y_i, H(c_i \| \mathrm{LAN} \| U_i \| \mathrm{TS}))$. In order to efficiently proceed the verification, the GW performs verification in a batch way as

$$\begin{aligned} e\left(P, \sum_{i=1}^{\omega} \sigma_i\right) &= \prod_{i=1}^{\omega} e(P, x_i H(c_i \| \mathrm{LAN} \| U_i \| \mathrm{TS})) \\ &= \prod_{i=1}^{\omega} e(Y_i, H(c_i \| \mathrm{LAN} \| U_i \| \mathrm{TS})). \end{aligned}$$

Similarly, the GW verifies the validity of the SG. Then it performs the following steps for privacy-preserving report aggregation. First, the GW aggregates the encrypted usage data $c_1, c_2, \ldots, c_\omega$ into c as

$$c = \prod_{i=1}^{\omega} c_i \mod n^2 = g^{m_1+m_2+\cdots+m_w}\left(\prod_{i=1}^{w} r_i^n\right) \mod n^2.$$

Similarly, the GW aggregates the encrypted detection results $a_1, a_2, \ldots, a_\omega$ into a as

$$a = \prod_{i=1}^{\omega} a_i \mod n^2 = g^{a_1+a_2+\cdots+a_w}\left(\prod_{i=1}^{w} r_i^n\right) \mod n^2.$$

Then, the GW uses its private key x_g to produce a signature $\sigma_g = x_g H(c\|a\|\mathrm{LAN}\|GW\|\mathrm{TS})$, where TS is the current time stamp. Finally, the GW publishes the aggregated encryption data $c\|a\|\mathrm{LAN}\|GW\|\mathrm{TS}\|\sigma_g$ to the CC.

G. Decryption the Aggregated Ciphertext

Upon $c\|a\|\mathrm{LAN}\|GW\|\mathrm{TS}\|\sigma_g$, the CC checks $e(P, \sigma_g) = e(Y_g, H(c\|a\|\mathrm{LAN}\|GW\|\mathrm{TS}))$, and decrypts the aggregated data c and a as

$$m = \frac{L\left(c^{\lambda} \mod n^2\right)}{L\left(g^{\lambda} \mod n^2\right)} \mod n$$

$$dt = \frac{L\left(a^{\lambda} \mod n^2\right)}{L\left(g^{\lambda} \mod n^2\right)} \mod n$$

where $m = m_1 + m_2 + \cdots + m_w$ and $t = t_1 + t_2 + \cdots + t_w$.

So far, the CC gets the knowledge of sum of energy usage and the number of normal meters and abnormal meters in an area without knowing each user's energy usage and the number of normal meters, in order to give an accurate decision for the grid.

Part 2: Theft Detection With Our Proposed Combined CNN

In this part, we show how to use our combined CNN model to analyze the decrypted data of smart meters and send the detection results in a ciphertext version.

A. Data Preprocessing

The energy usage data consists of missing or erroneous values. We exploit the forward interpolation method to recover the missing values as

$$f(x_i) = \begin{cases} 0, & x_i \in \mathrm{NaN}, i = 1 \\ x_{i-1}, & x_i \in \mathrm{NaN}, i > 1 \\ x_i, & x_i \notin \mathrm{NaN} \end{cases}$$

where x_i represents the value in the energy usage data over a period (e.g., a day). If x_i is a null or a non-numeric character, we set it as a member of NaN (NaN is a set). We obtain energy data from m users for n units of time. Assume that the length of time period is c units, we have the sample data set $n = k*c$ based on its k time cycles, including a reference group users' data.

We could get m single samples in total and each sample s is a vector whose length is $k * c + 1$, where the last value of the vector is y which is a single value (1 or 0). Assuming the size of each group is g, some samples can be combined to reference groups. We construct a $k * c$ matrix S from previous values of each sample s.

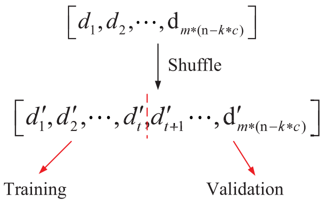

$[d_1, d_2, \cdots, d_{m*(n-k*c)}]$
Shuffle
$[d'_1, d'_2, \cdots, d'_t, d'_{t+1} \cdots, d'_{m*(n-k*c)}]$
Training
Validation

Fig. 3. Split data set.

For the common CNN input, its shape should be (height, width, channel), which means the height, width, and color channel of an image. To format our data, the height should be the number of cycle (k), the width should be the length of cycle (c), and we can use the channel dimension to present the number of different users. For a single target user, the channel should be 1. For the reference group users, the channel should be g. So we can get an image-like data structure for a single user, namely,

$$\begin{pmatrix} [S_{1,1}] & \cdots & [S_{1,c}] \\ \vdots & \ddots & \vdots \\ [S_{k,1}] & \cdots & [S_{k,c}] \end{pmatrix}$$

the shape is $(k, c, 1)$. And we can get an image-like data structure for group users, the shape is (k, c, g).

For the output data, our purpose is to detect the label value of current data, it should be a single dimensional vector as $[s_{k*c+1}]$, the shape is (1). For the models we want to train, each of them have two data set, one is the training data set, the other is the validation data set. In order to split them out, we apply the shuffle algorithm in our data set first, then slice the data set to get training and validation data. So the samples are randomly selected as training or validation, as shown in Fig. 3.

B. Our Combined CNN Model

We use 2-D convolution layers and full connection layers to build our proposed combined CNN framework, and use the merge layer to merge two input threads as shown in Fig. 4, which includes three stages and a combine.

1) *Individual Features Extracting Stage:* For the input layer, we have two input threads, one for the target user, and one for the reference users. The shape of target input data should be $(j, c, 1)$ since the target is a single user, while the shape of reference input data should be (j, c, g) since there are g users as reference. We use a dedicated convolution layer for the two input threads, they both have α convolution filters; after the double parallel convolution layers, the shape of two sets of data will all be (j, c, α).
2) *Combine:* We use multiple parallel convolution layers to extract features from the two input data independently. Then, we use merge layer to combine the features from the target data and reference data. Assume that after β parallel layers, the two sets of data are all in the form of (j, c, α), it turns into $(j, c, 2\alpha)$ after the merge layer.

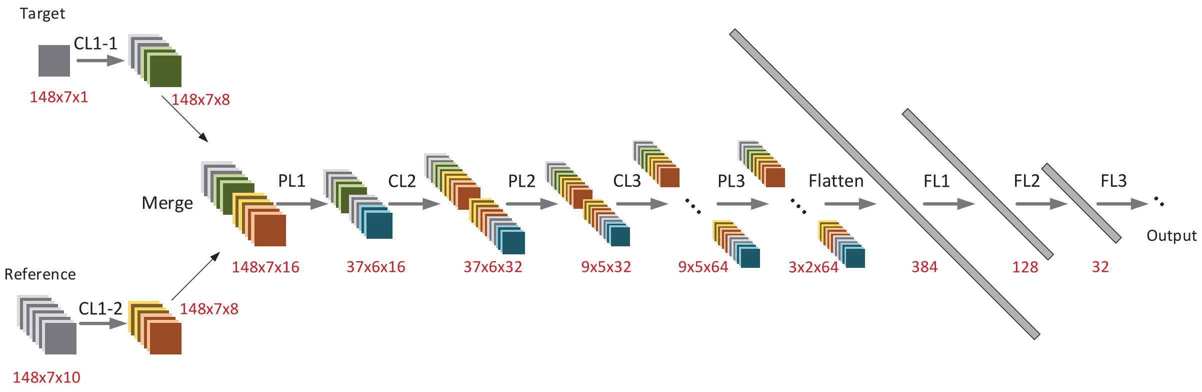


Fig. 4. Our proposed combined CNN model framework.

3) *Combined Features Extracting Stage:* We stack convolution and pooling layers just as common CNN models. To extract more features and reduce computation, we stack the convolution layer and the pooling layer alternately, one convolution layer and one pooling layer each time, while the convolution layer doubles the number of features and the pooling layer changes the shape. For example, assume the current shape is (j, c, γ), after pooling layer, it become $(j/2, c - 1, \gamma)$; after convolution layer, it become $(j/2, c - 1, 2\gamma)$.
4) *Combined Features Reducing Stage:* After several convolution and pooling layers, we flatten the shape to one dimension so we can stack some full connection layers. The shape of the layer just flattened will be very large, we add a full connection layer whose length is φ to change the shape to (φ), and stack smaller full connection layers in the following. Finally, we employ a full connecting layer with softmax as the output layer to classify the target. Since we only have two categories, theft or not, so the final output shape is (2). One is the probability of theft, the other is the probability of normal, and the sum of the two is 1. If the probability of theft is greater than normal, we think the metering data is abnormal, and vice versa.

The above variables are adjustable, which gives more space to optimize the expressiveness and efficiency for our combined CNN model. Fig. 4 shows an example parameter set, which j equals 128, c equals 7, g equals 10, α equals 8, and ρ equals 128. To simplify the example model, we add one parallel convolution layer in the individual features extracting stage, three pooling layers and two convolution layers in the combined features extracting stage, and three full connection layers in the combined features reducing stage.

VI. Security Analysis

In this section, we analyze the security of the proposed scheme. According to the security requirements proposed in Section III-C, we discuss whether the proposed scheme meets the requirements.

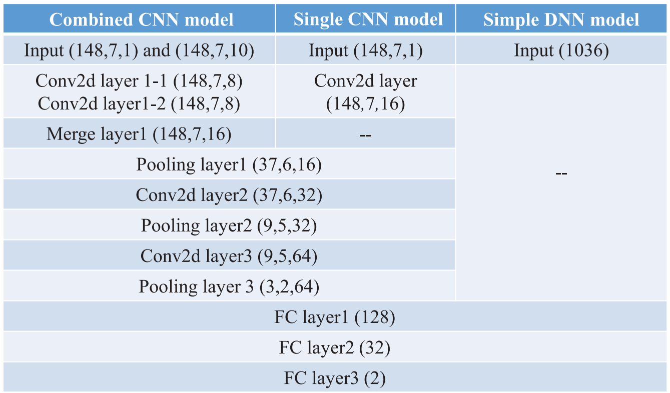
1) *Fine-Grained Data Privacy Preservation:* In the scheme, the energy usage m_i and detection result t_i are encrypted, while the LAN aggregates users' information into c and t and then sends them to the CC. For another side, the usage data sent to the CC is an aggregated set that include a number of users, in such a case that, the CC cannot reveal the energy usage of each individual user. In addition, the CC can access the number of user who honestly reports its usage data rather than the state of each individual user.
2) *Data Confidentiality:* The user sends the energy usage $c_i \| \text{LAN} \| U_i \| \text{TS} \| \sigma_i$ and the SG sends the encrypted detection result $a_i \| \text{LAN} \| \text{SG} \| \text{TS} \| \beta_i$ to the LAN. Here, other users including the LAN know nothing about the actual energy usage plaintext and detection result. The data of smart meters are sent to LAN after being encrypted by the Paillier algorithm, and then transmitted to CC by encrypted aggregated data under homomorphism. In this process, the users data have been transmitted in ciphertext formats, the attacker cannot get any information about the data.
3) *Data Authentication and Data Integrity:* In our scheme, each user's data and the aggregated data are signed by a short signature combined with a timestamp, in such a way that, the validation and the integrity of the data can be nicely guaranteed. If any adversary attempts to modify the stored data, the LAN GW or CC can detect it.

VII. Experimental Results

In order to evaluate the proposed energy theft detection scheme with energy privacy preservation, we conduct the simulations on a 64 bit computer with dual Intel Core i5-2410M 2.30-GHz CPU and 4-GB RAM, using Python, Numpy, Pandas, TensorFlow, and Keras. The energy usage data comes from SGCC [37].

A. Experimental Data

We get the data from 42 372 users during two and a half years, where each value means energy usage of each day and the data has similarity per cycle whose length is seven days [15]. Therefore, we set c to be 7 and k to be 148, which is used to detect theft based on history data. By randomly selecting 80% of samples from the total 44 218 samples, we compose the training data set while the remaining 20% of samples to compose the validation data set. Moreover, we use Keras as the implementation tool to build and train our model.



Combined CNN model	Single CNN model	Simple DNN model
Input (148,7,1) and (148,7,10)	Input (148,7,1)	Input (1036)
Conv2d layer 1-1 (148,7,8) Conv2d layer1-2 (148,7,8)	Conv2d layer (148,7,16)	--
Merge layer1 (148,7,16)	--	
Pooling layer1 (37,6,16)		
Conv2d layer2 (37,6,32)		
Pooling layer2 (9,5,32)		
Conv2d layer3 (9,5,64)		
Pooling layer 3 (3,2,64)		
FC layer1 (128)		
FC layer2 (32)		
FC layer3 (2)		

Fig. 5. Model configurations.

Accuracy Score: To train our model, we use the categorical cross-entropy as the loss function. To evaluate the performance of models, we use accuracy score as performance score. The y means the predicted value and $\hat{y}$ means the true value. y_i is the predicted value of the ith sample and $\hat{y}_i$ is the corresponding true value. The following models share this accuracy score as:

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} \delta(\hat{y}_i, y_i)$$

where

$$\delta(\hat{y}_i, y_i) = \begin{cases} 1, & \hat{y}_i = y_i \\ 0, & \text{else.} \end{cases}$$

B. Model Comparison

We derive the described CNN framework to build the proposed combined CNN model shown in Fig. 4. As comparison, we employ a single CNN model and a simple deep neural network (DNN) model. Our proposed combined CNN model includes two input layers, four convolution layers, three pooling layers, and three full connection layers. The single CNN model include three convolution layers, three pooling layers, and three full connection layers. Simple DNN model only include three full connection layers.

The models configurations, evaluated in this paper, are outlined in Fig. 5, one per column. The shape of target input data of our proposed combined CNN model is $(148, 7, 1)$. Reference input data's shape is $(148, 7, 10)$. Through Conv2d layer 1-1, the shape of the target thread data become $(148, 7, 8)$. Followed by Conv2d layer 1-2, the shape of the reference thread data become $(148, 7, 8)$. Merge layer combines the target data and the reference data to one. After this layer, the shape of the data turns to be $(148, 7, 16)$. Pooling layer uses maxpooling, the shape of the data turns to be $(37, 6, 16)$. Similarly, the corresponding shapes are formed through these layers in sequence in each model.

As shown in Table I, we set random values as the origin weights, 128 as the batch size. After that, we compile the model with SGD optimizer and the loss function. To evaluate the performance of models, we consider the accuracy score as the metric function.

TABLE I
MODEL CONFIGURATIONS

	Origin weights	Optimizer
Proposed combined CNN model	random	SGD
Single CNN model	random	SGD
Simple DNN model	random	SGD

After model compiling, we train the model using input data in a batch way and evaluate the performance metric value in each epoch, where an epoch means that all samples are selected once at the training data set. The training results are shown in Fig. 6, where the horizontal axis indicates the number of epochs of training and the vertical axis indicates the average loss and accuracy score value. We observe that the average loss of our proposed combined CNN model becomes smaller and smaller as the training goes, and it achieves a higher accuracy score than the single CNN model after 100 epochs training. The training result of the simple DNN model achieves a lower accuracy score than CNN models after 100 epochs training.

C. Method Comparison

To evaluate the performance of our combined CNN model, we present the experimental results over the given dataset to have a performance comparison with other traditional machine learning methods. Concretely, linear SVC [39] is an implementation of support vector classification for the linear kernel case; random forest [40] is an averaging algorithm based on randomized decision trees; logistic regression [41] uses a logistic function to describe the possible outcomes of a single trial are modeled. Table II gives the arguments used for the baseline methods to train these models. Using the same training data set and validation data set, we see that our proposed combined CNN model gets the highest accuracy score of 0.9267 from Table II.

D. Parameter Study

There are various configurable parameters of model which cannot optimize by training but can affect the performance of model, such as batch size, learning rate of optimizer, and dropout rate. Batch size χ means how many samples will be used to evaluate loss and do optimize each times; learning rate ε defines how many loss gradient will be used to optimize the model and determines how fast the optimization is; dropout rate τ defines how much the ratio of signals between layers will be random ignored; to avoid overfitting, dropout layers are imported to the model. Hence, we give a deep analysis on the impacts of these parameters on the performance of our proposed combined CNN model. In Fig. 6, the parameter of the proposed combined CNN model is $\chi = 32, \varepsilon = 0.1$, and $\tau = 0.2$. The training results of models for parameter study as follows.

1) Effect of Batch Size χ: Fig. 7 shows the performance of our combined model (1) with setting the batch size as 128 which achieves a high max accuracy with 0.9268, while needs more epochs to optimize. We can see that a smaller batch size can speed up the optimizing within same epochs, which suggests that setting the bath size between 32 an 128 is more acceptable.

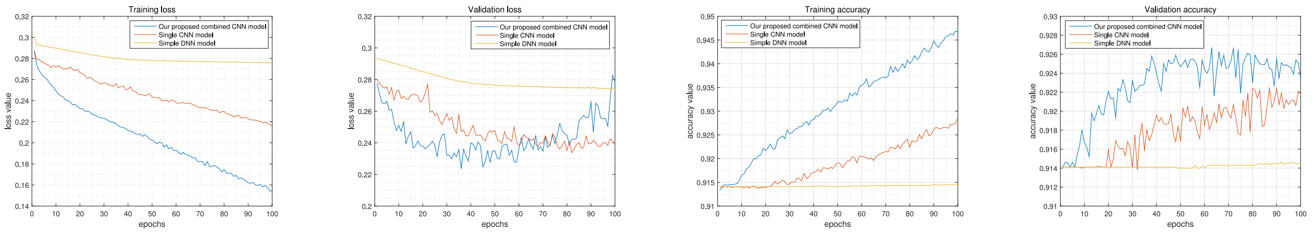


Fig. 6. Model comparison.

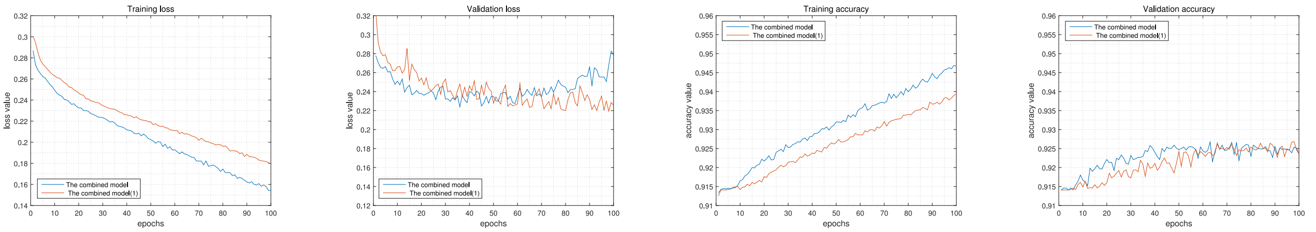


Fig. 7. Parameter study of χ.

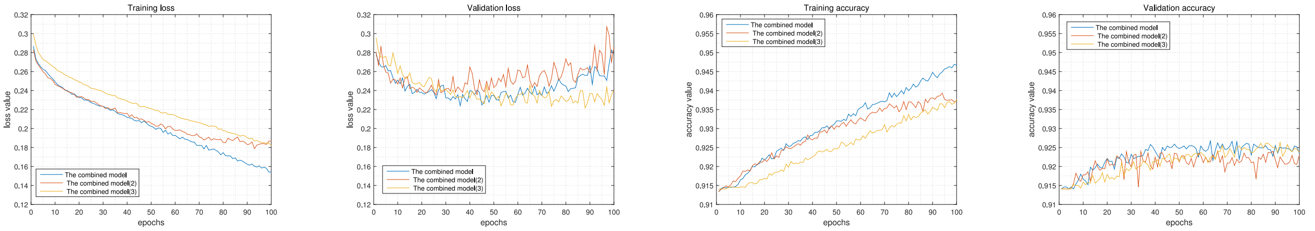


Fig. 8. Parameter study of ε.

TABLE II
ALGORITHM ACCURACY SCORES

Algorithm	Arguments	Accuracy score
Proposed combined CNN model	100 epochs	0.9267
Single CNN model	100 epochs	0.9218
Simple DNN model	100 epochs	0.9145
Linear SVC	kernel: linear function	0.9178
Random Forest	max depth: 7	0.9164
Logistic Regression	penalty: L2	0.9141

2) Effect of Learning Rate ε: Fig. 8 shows the performance of our combined model (2) with the learning rate is 0.3 which achieves a lower accuracy of 0.9241 and reaches its max accuracy faster. While the learning rate in comparing combined model (3) is 0.03 shown in Fig. 8, this model achieves a high accuracy of 0.9263 and achieves its max accuracy later. We can see that the learning rate affects the speed of optimizing, and in our model setting learning rate not bigger than 0.1 is more acceptable.

3) Effect of Dropout Rate τ: The dropout rate in the combined model (4) is 0.1 shown in Fig. 9, which achieves a high accuracy of 0.9267 but not always steady. And the performance gap between the validation set and the training set is very big, while the max accuracy in the training set is 0.9607. We can see that the dropout rate reduces overfitting; and in our model importing dropout and setting its rate bigger than 0.1 is more acceptable.

E. Comparison With Existing Schemes

This section elaborates the comparison of the proposed scheme with the existing schemes. The comparison results reveal that user energy usage in Zheng *et al.*'s [15] and Jindal *et al.*'s [10] scheme may be leaked and users' privacy cannot be guaranteed. Salinas and Li's scheme [20] cannot realize the dispatch in the smart grid because the CC cannot know the total energy usage in the area by sending residual to the operator. Furthermore, Salinas's scheme and Jindal's scheme are unable to detect theft for massive data. Thus, as shown in Table III, we can see that the proposed scheme can achieve user privacy and the dispatching of the smart grid, and detect energy theft for massive data.

F. Proposed Combined CNN Model Versus [15]

As the only two schemes that used the neural network to detect energy theft, we make a comparison. We build the model which consists of the wide component and the deep CNN component from [15], and set paraments in the model as $\omega = 90$, $\psi = 60$, $\xi = 90$, and $\zeta = 3$ (ω, ψ, ξ: a parameter controlling the number of neurons, ζ: the number of convolution layers) to compare with our proposed combined CNN model shown in Fig. 4 under the same data set.

As shown in Fig. 10, using our proposed combined CNN model, the loss decrease faster and accuracy increase faster at training and validation set; the performance of our proposed

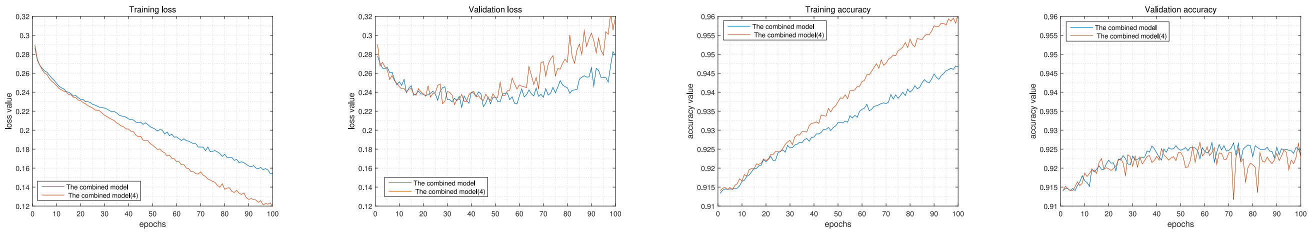


Fig. 9. Parameter study of τ.

TABLE III
PROPERTIES COMPARISON

	Proposed scheme	Zheng *et al.* [15]	Salinas *et al.* [20]	Jindal *et al.* [10]
Technique adopted	CNN	CNN	state estimation	decision tree and SVM
User privacy	Yes	No	Yes	No
Dispatching of smart grid	Yes	No	Yes	Yes
Massive data processing	Yes	Yes	No	No

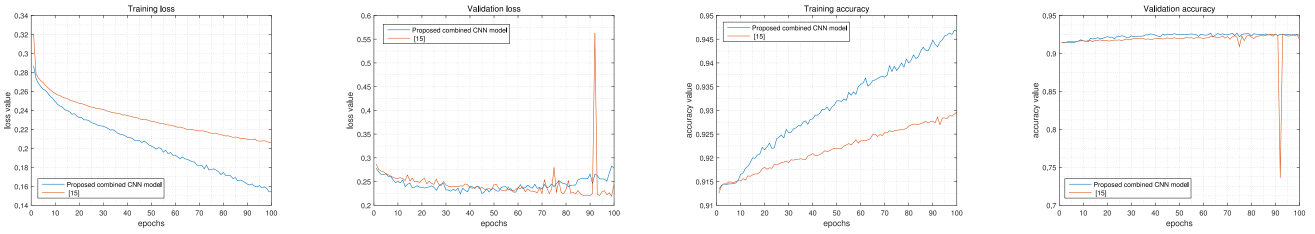


Fig. 10. Our proposed combined CNN model versus [15].

combined CNN model improves more stable along with training; finally, the max accuracy we achieved is 0.9267, is larger than 0.9254 which [15] achieved. This improvement may owe to using two input threads from a user group perspective to study the energy theft behavior.

VIII. CONCLUSION

In this paper, we have proposed an energy theft detection scheme with energy privacy preservation in the smart grid. The energy theft detection based on our proposed combined CNN model is used to detect whether the metering data has an abnormal behavior. Moreover, the usage data of users and the number of users who honestly report their usage data are protected by the Paillier homomorphic algorithm. In addition, the security analysis shows that our scheme achieves confidentiality and integrity, as well as data privacy. The experimental results show that the accuracy of anomaly detection is more better than others. For our future work, we intend to improve our scheme with less communication and computing overhead.

REFERENCES

[1] Y. Sun, L. Lampe, and V. W. S. Wong, “Smart meter privacy: Exploiting the potential of household energy storage units,” *IEEE Internet Things J.*, vol. 5, no. 1, pp. 69–78, Feb. 2018.

[2] T. Song *et al.*, “A privacy preserving communication protocol for IoT applications in smart homes,” *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1844–1852, Dec. 2017.

[3] S. Mclaughlin, D. Podkuiko, and P. Mcdaniel, “Energy theft in the advanced metering infrastructure,” in *Proc. Crit. Inf. Infrastruct. Security Int. Workshop (CRITIS)*, Bonn, Germany, Oct. 2009, pp. 176–187.

[4] R. Jiang *et al.*, “Energy-theft detection issues for advanced metering infrastructure in smart grid,” *Tsinghua Sci. Technol.*, vol. 19, no. 2, pp. 105–120, Apr. 2014.

[5] M. Wen, “PaRQ: A privacy-preserving range query scheme over encrypted metering data for smart grid,” *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 1, pp. 178–191, Jun. 2013.

[6] T. Ahmad, D. Q. U. Hasan, and S. Zada, “Non-technical loss detection, prevention and suppression issues for AMI in smart grid,” *Int. J. Sci. Eng. Res.*, vol. 6, no. 3, pp. 217–228, 2015.

[7] P. Mcdaniel and S. Mclaughlin, “Security and privacy challenges in the smart grid,” *IEEE Security Privacy*, vol. 7, no. 3, pp. 75–77, May/Jun. 2009.

[8] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, “EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[9] S. S. S. R. Depuru, L. Wang, and V. Devabhaktuni, “Support vector machine based data classification for detection of electricity theft,” in *Proc. Power Syst. Conf. Expo.*, Phoenix, AZ, USA, 2011, pp. 1–8.

[10] A. Jindal *et al.*, “Decision tree and SVM-based data analytics for theft detection in smart grid,” *IEEE Trans. Ind. Informat.*, vol. 12, no. 3, pp. 1005–1016, Jun. 2016.

[11] J. Nagi, K. S. Yap, S. K. Tiong, S. K. Ahmed, and A. M. Mohammad, “Detection of abnormalities and electricity theft using genetic support vector machines,” in *Proc. TENCON IEEE Region 10 Conf.*, 2009, pp. 1–6.

[12] S. Salinas, M. Li, and P. Li, “Privacy-preserving energy theft detection in smart grids,” in *Proc. Sensor Mesh Ad Hoc Commun. Netw.*, 2012, pp. 257–267.

[13] S.-C. Yip, C.-K. Tan, W.-N. Tan, M.-T. Gan, and A.-H. A. Bakar, “Energy theft and defective meters detection in AMI using linear regression,” in *Proc. IEEE Int. Conf. Environ. Elect. Eng. IEEE Ind. Commercial Power Syst. Europe*, 2017, pp. 1–6.

[14] S. Salinas, C. Luo, W. Liao, and P. Li, “State estimation for energy theft detection in microgrids,” in *Proc. Int. Conf. Commun. Netw. China*, 2015, pp. 96–101.

[15] Z. Zheng, Y. Yang, X. Niu, H. N. Dai, and Y. Zhou, “Wide & deep convolutional neural networks for electricity-theft detection to secure smart grids,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1606–1615, Apr. 2018.

[16] A. Abdallah and X. Shen, “Lightweight security and privacy preserving scheme for smart grid customer-side networks,” *IEEE Trans. Smart Grid*, vol. 8, no. 3, pp. 1064–1074, May 2017.

[17] M. Wen, X. Zhang, H. Li, and J. Li, "A data aggregation scheme with fine-grained access control for the smart grid," in *Proc. Veh. Technol. Conf.*, Toronto, ON, Canada, 2018, pp. 1–5.

[18] M. Wen, R. Lu, J. Lei, X. Liang, H. Li, and X. Shen, "ECQ: An efficient conjunctive query scheme over encrypted multidimensional data in smart grid," in *Proc. Glob. Commun. Conf.*, Atlanta, GA, USA, 2013, pp. 796–801.

[19] C. Richardson, N. Race, and P. Smith, "A privacy preserving approach to energy theft detection in smart grids," in *Proc. Smart Cities Conf.*, 2016, pp. 1–4.

[20] S. A. Salinas and P. Li, "Privacy-preserving energy theft detection in microgrids: A state estimation approach," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 883–894, Mar. 2016.

[21] W. Luan *et al.*, "Energy theft detection via integrated distribution state estimation based on AMI and SCADA measurements," in *Proc. Int. Conf. Elect. Utility Deregulation Restruct. Power Technol.*, 2016, pp. 751–756.

[22] C.-L. Su, W.-H. Lee, and C.-K. Wen, "Electricity theft detection in low voltage networks with smart meters using state estimation," in *Proc. IEEE Int. Conf. Ind. Technol.*, 2016, pp. 493–498.

[23] S.-C. Huang, Y.-L. Lo, and C.-N. Lu, "Non-technical loss detection using state estimation and analysis of variance," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2959–2966, Aug. 2013.

[24] S. Amin, G. A. Schwartz, A. A. Cardenas, and S. S. Sastry, "Game-theoretic models of electricity theft detection in smart utility networks: Providing new capabilities with advanced metering infrastructure," *IEEE Control Syst. Mag.*, vol. 35, no. 1, pp. 66–81, Feb. 2015.

[25] A. A. Cárdenas, S. Amin, G. Schwartz, R. Dong, and S. Sastry, "A game theory model for electricity theft detection and privacy-aware control in AMI systems," in *Proc. Commun. Control Comput.*, 2012, pp. 1830–1837.

[26] D. R. Pereira *et al.*, "Social-spider optimization-based support vector machines applied for energy theft detection," *Comput. Elect. Eng.*, vol. 49, pp. 25–38, Jan. 2016.

[27] Z. Erkin and G. Tsudik, "Private computation of spatial and temporal power consumption with smart meters," in *Proc. Int. Conf. Appl. Cryptography Netw. Security*, 2012, pp. 561–577.

[28] F. Li, B. Luo, and P. Liu, "Secure and privacy-preserving information aggregation for smart grids," *Int. J. Security Netw.*, vol. 6, no. 1, pp. 28–39, 2011.

[29] H. Bao and R. Lu, "A new differentially private data aggregation with fault tolerance for smart grid communications," *IEEE Internet Things J.*, vol. 2, no. 3, pp. 248–258, Jun. 2017.

[30] P. Duplessis and P. Lescuyer, "Access network, gateway and management server for a cellular wireless communication system," U.S. Patent Appl. 12 093 693, 2010.

[31] Y. H. Heo, Z. Cai, A. M. Earnshaw, S. Mcbeath, and M. H. Fong, "Reporting power headroom for aggregated carriers," U.S. Patent 8 351 359, Jan. 8, 2013.

[32] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 791–808.

[33] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, Canada, 2013, pp. 8609–8613.

[34] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[35] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade* (LNCS 7700), G. Montavon, G. B. Orr, and K. R. Müller, Eds. Berlin, Germany: Springer, 2012.

[36] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 1999, pp. 223–238.

[37] *DataFountain*. Accessed: Dec. 2016. [Online]. Available: https://www.datafountain.cn./competitions/241/details

[38] B. Dan, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security Adv. Cryptol.*, 2001, pp. 1–24.

[39] H. Li and B. Liu, "Loss analysis simulation of SVC/DC deicer under SVC mode," in *Proc. Power Energy Eng. Conf.*, 2016, pp. 2564–2568.

[40] V. Svetnik *et al.*, "Random forest: A classification and regression tool for compound classification and QSAR modeling," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, p. 1947, 2003.

[41] D. W. Hosmer, T. Hosmer, C. S. Le, and S. Lemeshow, "A comparison of goodness-of-fit tests for the logistic regression model," *Stat. Med.*, vol. 16, no. 9, pp. 965–980, 2015.

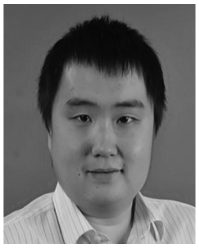
Donghuan Yao received the bachelor's degree from the School of Electronics and Electrical Engineering, Changsha University, Changsha, China, in 2015, and the master's degree from the Department of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, China, in 2019.

Her current research interests include smart grid and information security.

Mi Wen (M'10) received the M.S. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2008.

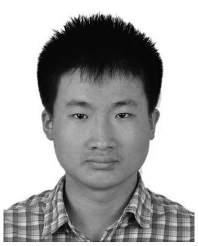
She is currently an Associate Professor with the College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai. From 2012 to 2013, she was a Visiting Scholar with the University of Waterloo, Waterloo, ON, Canada. Her current research interests include privacy preserving in wireless sensor networks and smart grid.

Dr. Wen serves an Associate Editor for *Peer-to-Peer Networking and Applications* (Springer). She is a TPC member for some flagship conferences, such as IEEE INFOCOM, IEEE ICC, and IEEE GLOEBECOM in 2012.



Xiaohui Liang (M'13) received the B.Sc. degree in computer science and engineering and the M.Sc. degree in computer software and theory from Shanghai Jiao Tong University, Shanghai, China, in 2006 and 2009, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2013.

He is currently an Assistant Professor with the Department of Computer Science, University of Massachusetts, Boston, MA, USA. His research interests include applied cryptography, security and privacy issues for e-healthcare system, cloud computing, mobile social networks, and smart grid.

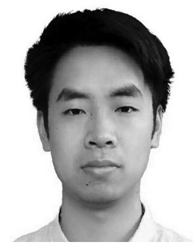


Zipeng Fu is currently pursuing the B.Sc. degree in computer science and engineering and the B.Sc. degree in applied mathematics at the University of California at Los Angeles, Los Angeles, CA, USA.

His current research interests include multiagent reinforcement learning and machine learning.

Kai Zhang received the bachelor's degree from the School of Information Science and Engineering, Shandong Normal University, Jinan, China, in 2012, and the Ph.D. degree from the Department of Computer Science and Technology, East China Normal University, Shanghai, China, in 2017.

He is currently an Assistant Professor with the Shanghai University of Electric Power, Shanghai. His current research interests include applied cryptography and information security.



Baojia Yang received the bachelor's degree from the Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2014, and the master's degree in computer technology from the University of Chinese Academy of Sciences, Beijing, China, in 2017.

He is currently a Software Engineer with Microsoft Suzhou, Suzhou, China. His current research interests include machine learning and deep learning.