Privacy Preserving and Efficient Data Collection Scheme for AMI Networks Using Deep Learning

Mohamed I. Ibrahem, Mohamed Mahmoud, *Senior Member, IEEE*, Mostafa M. Fouda, *Senior Member, IEEE*, Fawaz Alsolami, *Member, IEEE*, Waleed Alasmary, *Senior Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—In advanced metering infrastructure, smart meters (SMs) send fine-grained power consumption readings periodically to the utility for load monitoring and energy management. Change and transmit (CAT) is an efficient approach to collect these readings, where the readings are not transmitted when there is no enough change in consumption. However, this approach causes a privacy problem, that is, by analyzing the transmission pattern of an SM, sensitive information on the house dwellers can be inferred. For instance, since the transmission pattern is distinguishable when dwellers are on travel, attackers may analyze the pattern to launch a presence-privacy attack (PPA) to infer whether the dwellers are absent from home. In this article, we propose a scheme, called "STDL," for efficient collection of power consumption readings in advanced metering infrastructure (AMI) networks while preserving the consumers' privacy by sending spoofing transmissions using a deep-learning approach. We first use a clustering technique and real power consumption readings to create a data set for transmission patterns using the CAT approach. Then, we train a deep-learning-based attacker model, and our evaluations indicate that the attacker's success rate is about 91%. Finally, we train a deep-learning-based defense model to send spoofing transmissions efficiently to thwart the PPA. Extensive evaluations are conducted, and the results indicate that our scheme can reduce the attacker's success rate to 3.15%, while still achieving high efficiency in terms of the number of readings that should be transmitted. Our measurements indicate that the proposed scheme can increase efficiency by about 41% compared to continuously transmitting readings.**

***Index Terms*—Advanced metering infrastructure (AMI) networks, privacy preservation, smart grid, traffic analysis attack.**

I. Introduction

SMART grid is a revolutionary upgrade to the traditional power grid that aims to increase the reliability of electricity delivery, optimize grid operation, and reduce the emissions of greenhouse gases by using more renewable resources for power generation [1]. Advanced metering infrastructure (AMI) is one of the most important components of the smart grid. AMI enables bidirectional communication between the smart meters (SMs), which are deployed at consumer premises, and the electricity utility (EU) to collect fine-grained power consumption readings periodically (every few minutes) to obtain detailed information necessary for monitoring the grid load and managing the energy supply efficiently [2], [3].

However, revealing the consumers' fine-grained power consumption readings creates a serious privacy problem. This is because the readings can be analyzed to infer sensitive information about the consumers' life habits, such as the appliances that are being used, when consumers sleep, when they return home, etc., [4]. This is because each appliance has a unique signature on the power consumption. To preserve consumers' privacy, encryption schemes have been used to hide the fine-grained readings while enabling the EU to use the encrypted readings for different purposes such as load monitoring [5]–[7].

On the other hand, reporting the fine-grained readings periodically results in transmitting a massive amount of data by each SM. What makes the problem worse is that the AMI networks are scalable, i.e., they have millions of SMs. This leads to inefficient use of the available communication bandwidth, and in case of using cellular networks to transmit the SMs' readings, it is costly to transmit this large amount of data, particularly after considering the overhead caused by the encryption schemes. In a more efficient power consumption readings collection approach, called change and transmit (CAT), SMs do not need to report their consumption when there is no enough change compared to the last reported reading [8]. More specifically, in the CAT approach, a reading is transmitted when the absolute value of the percentage of change in the consumption exceeds a predefined threshold.

Manuscript received November 9, 2020; revised April 4, 2021; accepted May 3, 2021. Date of publication May 6, 2021; date of current version November 19, 2021. This work was supported by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under Grant DF-745-611-1441. *(Corresponding author: Mohamed I. Ibrahem.)*

Mohamed I. Ibrahem is with the Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN 38505 USA, and also with the Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11672, Egypt (e-mail: miibrahem42@tntech.edu).

Mohamed Mahmoud is with the Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN 38505 USA (e-mail: mmahmoud@tntech.edu).

Mostafa M. Fouda is with the Department of Electrical and Computer Engineering, College of Science and Engineering, Idaho State University, Pocatello, ID 83209 USA, and also with the Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11672, Egypt (e-mail: mfouda@ieee.org).

Fawaz Alsolami is with the Department of Computer Science, King Abdulaziz University, Jeddah 21341, Saudi Arabia (e-mail: falsolami1@kau.edu.sa).

Waleed Alasmary is with the Department of Computer Engineering, Umm Al-Qura University, Mecca 21421, Saudi Arabia (e-mail: wsasmary@uqu.edu.sa).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2021.3077897

Nevertheless, using the CAT approach causes a privacy problem, that is, by analyzing the transmission patterns of the SMs, sensitive information on the consumers can be revealed even if the readings are encrypted [8], [9]. For instance, these information can include whether the house dwellers are on travel (absent) or present at home, sleeping cycles, meal times, the number of dwellers, etc., [10], [11], because these events have distinguishable impact on the transmission pattern. These private information can be used for criminal purposes such as finding the right time to commit crimes, e.g., kidnapping, pillage, burglary, arson, and vandalism [12].

In this article, we focus on thwarting presence-privacy attack (PPA) that aims at inferring whether the house dwellers are present at home or absent by analyzing the transmission pattern of the SM. The seriousness of PPA is due to the following facts. First, PPA can be launched in an undetectable way since attackers usually work completely in the receiving mode to overhear the transmissions without disrupting the communications. Second, the use of encryption schemes cannot thwart the attack because the attackers analyze the transmission patterns and do not need to decrypt the readings. Third, AMIs mostly use wireless communication [8], [13], and the broadcast nature of this communication can facilitate collecting transmission patterns of victim consumers. In summary, the research problem we address in this article is how to thwart PPA, while achieving a high efficiency in terms of the number of readings that should be transmitted.

In the literature, a countermeasure to PPA has been proposed in [8] by sending spoofing transmissions (packets for redundant readings) to make it difficult for the attacker to analyze the transmission pattern to infer whether dwellers are absent or present. Artificial spoofing packets (ASP) are generated either by using the distribution of the power consumption transmission pattern of the consumers or using their old transmission patterns. However, the scheme suffers from the following limitations. First, the success rate of the attackers is too high, i.e., above 60%. Second, the attacker model is very simple, i.e., the attacker collects transmission patterns and compares their distribution with the distribution of the old transmission patterns of the consumers. Third, the scheme in [8] considers limited data that is for one household over only 33 days. Fourth, the utility is assumed to be trusted to learn the readings and identify the spoofing transmissions. Finally, the proposed defense in [8] is customized for one consumer because it depends on the distribution of the transmission pattern of the consumer.

Therefore, in this article, we address these limitations by proposing a new scheme, called "STDL", for efficient collection of power consumption readings while thwarting PPA by transmitting spoofing transmissions (redundant real readings) using deep learning. First, a clustering technique and real power consumption readings are used to create a data set for transmission patterns using the CAT approach. Then, an attacker model is trained using deep learning, and our evaluations indicate that the success rate of the attacker is about 91%. Hence, to thwart PPA, we train a deep-learning-based defense model to send spoofing transmissions (redundant real readings) efficiently. This defense model is trained on the transmission pattern of the consumers when they are present at home. Next, the defense model is used to generate spoofing transmissions so that the attacker cannot distinguish the pattern generated by the model from the patterns transmitted when the consumers are at home to preserve consumers' privacy. Furthermore, most of the existing privacy-preserving electricity consumption readings collection schemes in the literature use the homomorphic encryption scheme to aggregate the readings that are sent periodically by SMs, while in our work, we use homomorphic encryption in case of the CAT approach, where some readings may not be sent if there is no enough change, and thus, the ciphertext of the last reported reading should be used.

The evaluations of our scheme indicate that our scheme can significantly reduce the attacker's success rate to less than 10% compared to more than 60% using the proposed scheme in [8]. Our scheme is also robust even if the attacker knows the defense model. In addition, it can reduce the number of transmitted readings by about 41% compared to sending consumption readings periodically.

Our contributions in this article can be summarized as follows.

1) To the best of our knowledge, this work is the first to investigate PPA using a deep-learning approach. Compared to the literature, we consider a more sophisticated and strong attack model that is trained on the consumers' transmission patterns to detect whether the dwellers are at home or not.
2) We propose a defense model against PPA, which offers a fairly low attacker's success rate compared to the literature since it is based on a hybrid deep-learning approach, which results in generating patterns that look like the patterns transmitted when the consumer is at home.
3) Our defense model is general (one model for all consumers) and it can be applied to new consumers who have no old transmission patterns.
4) We consider the network nodes, including the SMs, EU, and aggregator, are honest but curious, while the literature considered them trusted and the defense is limited only to the eavesdroppers.

The remainder of this article is organized as follows. Section II discusses the related works. Then, our system models and design objectives are discussed in Section III. Section IV illustrates the preliminaries used in this article. The data sets created for training our models are presented in Section V. Our envisioned defense scheme is presented in Section VI. Next, the performance evaluation of our scheme is discussed in Section VII. Finally, this article is concluded in Section VIII.

II. Related Work

Most of the existing privacy-preserving electricity consumption readings collection schemes in the literature [5]–[7] assume that the SMs report the power consumption readings periodically even if there is no enough change in the consumption compared to the last reported reading. These schemes use privacy-preserving data aggregation schemes to aggregate the encrypted readings and allow the EU to obtain the aggregated

reading for load monitoring and energy management without being able to access the individual readings of the SMs to preserve the consumers' privacy.

Mohammed *et al.* [5] proposed a privacy-preserving data collection scheme that uses one-time secrets to mask the readings so that the aggregated reading is obtained by adding the masked readings of the AMI network. This scheme cannot be applied in case of the CAT approach since it requires each SM to use a new mask in every data collection cycle, so if an SM does not send a reading, the last reported masked reading cannot be used in the aggregation. Moreover, a functional encryption-based scheme is proposed in [6] and [7] for aggregating the consumers' readings while preserving the consumers' privacy. This scheme cannot also be used in case of the CAT approach since it cannot use the last power consumption reading of an SM when the consumption does not change because the scheme requires using a fresh timestamp in each reading. Hence, the schemes proposed in [5]–[7] focused on preserving consumers' privacy by hiding the power consumption readings, while our focus in this article is on preserving the consumers' privacy by hiding the information that can be inferred from the consumers' transmission patterns in case of using the CAT approach.

Few works in the literature have investigated the CAT approach in smart grid [8], [14]–[19]. Some of these works studied using the CAT approach in different applications, such as demand/response [14] and load disaggregation [15], which tries to extract the power patterns of individual appliances from the aggregated power pattern measured by the SM to analyze the power consumption of each appliance. While other works [16]–[19] investigate finding a good value for the change in the consumption that necessitates transmitting a reading. The schemes in [14]–[19] that use the CAT approach do not consider consumers' privacy, so they are vulnerable to PPA.

Li *et al.* [8] have investigated PPA using the CAT approach. A defense scheme, called "ASP," is proposed to thwart PPA by generating artificial spoofing transmissions. To generate the spoofing transmissions, two heuristic-based methods have been proposed, namely, the Poisson packet generation and history template-based generation. In the Poisson packet generation method, the SM's memory stores the average number of power consumption changes in different time intervals when the dwellers are present at home. Then, when the dwellers are not at home, spoofing transmissions are generated so that the total number of changes within a certain time interval equals to that stored in the memory. On the other hand, in the history template method, old transmission patterns are used to generate the spoofing transmissions by either repeating the old transmission patterns or randomly generating the spoofing transmissions according to the distribution of the time intervals between the transmissions. However, ASP suffers from the following limitations.

1) The attacker's success rate is significantly high. In one experiment, it is assumed that the attacker knows a set of old transmission patterns, and the attacker's success rate in this case is above 60%. The attacker uses a K-S test to compare the distribution of the transmission pattern that is received from an SM with the distribution of the transmission patterns that he/she knows. The K-S test is used to calculate the error between two distributions to determine how they are close to each other. Another experiment assumes that the attacker does not have old transmission patterns, and in this case, the attacker collects transmission patterns sent in different time intervals and compares their distributions using the K-S test. If the attacker finds that the error between the distributions is small enough, he/she can conclude that the transmissions are spoofing, and thus, the consumer is on travel. The experimental results demonstrate that the attacker's success rate is above 40%. Overall, these results indicate that the attacker can detect, with high confidence, whether the consumer is present or on travel.
2) The proposed attack mechanism is very simple, i.e., the attacker collects transmission patterns received from SMs and analyzes them by comparing their distribution with the distribution of the old transmission patterns by using the K-S test. Therefore, a more sophisticated attack mechanism may result in higher success rate.
3) The study is very limited since only one household is considered over only 33 days. Also, the proposed scheme is customized, i.e., can be used for only one consumer, because it generates the spoofing transmissions based on the distribution of the transmission patterns of each consumer.
4) It is assumed that the utility is trusted. In practice, it is impossible to guarantee that a party (which is assumed trusted) does not misuse the captured data of the consumers.

In wireless sensor networks (WSNs), spoofing transmissions have been used to counter traffic analysis attacks that aim to determine the location of the sender node and the sink [20], [21]. However, these schemes cannot be applied to counter transmission pattern analysis in the AMI networks because WSNs have different characteristics and the privacy problem is different. We can summarize the main differences as follows: 1) the proposed schemes aim to preserve the location privacy of the source/destination sensor nodes, but we aim to preserve the privacy of the consumers' activities in AMI networks and 2) in WSNs, there is a restriction on energy consumption because most of the nodes are battery powered, but this restriction does not exist in AMI networks.

III. System Models and Design Objectives

This section discusses the considered network and threat models.

A. Network Model

As shown in Fig. 1, our considered network model includes SMs, EU, aggregator, and an offline key distribution center (KDC). The role of each entity is described as follows.

1) *KDC:* It is responsible for generating and distributing the public and private keys of the cryptosystem used in our scheme. After key distribution, the KDC is not involved in the SMs' readings collection process.

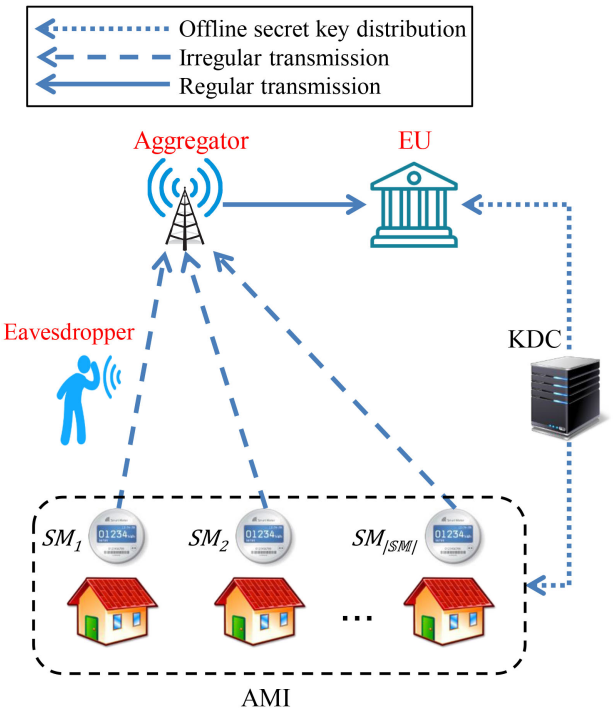

Fig. 1. Network model.

2) *SM:* A set of SMs, $\mathbb{SM} = \{\text{SM}_i, 1 \leq i \leq |\mathbb{SM}|\}$, forms an AMI network, where each SM_i reports the real-time fine-grained power consumption reading if and only if there is an enough change in the power consumption comparing to the last reported reading, i.e., if the absolute value of the percentage of the change between the current consumption and the last reported reading exceeds a certain threshold. For instance, if the threshold is 5%, a reading is sent when the change in the consumption is above +5% or below −5%. Also, each SM_i encrypts its power consumption reading and sends the ciphertext to a gateway, called the aggregator. In this case, the SMs may communicate directly with the gateway, or multihop data transmission is used to connect the SMs to the gateway, where some SMs may act as routers to relay other SMs' readings.
3) *Aggregator:* It acts as a local collector that aggregates all the received/stored encrypted power consumption readings sent by the SMs. Due to the use of the CAT approach, the aggregator stores the SMs' power consumption readings to reuse the last reported readings for future reporting cycles in case that an SM does not report a power consumption reading. After collecting the encrypted SMs' readings, the aggregator computes the ciphertext of the aggregated reading and forwards it to the EU without being able to reveal the individual readings or even the aggregated reading to preserve privacy.
4) *EU:* The EU uses the fine-grained power consumption readings sent by SMs for load monitoring and energy management. It can use its decryption key to decrypt the ciphertext sent by the aggregator to obtain the total power consumption of the consumers of an AMI network while hiding the fine-gained power consumption reading of each consumer to preserve privacy.

B. Threat Model

Threats may arise from either internal or external attackers. Internal attackers, which could be the SMs, EU, or aggregator, are assumed to be honest-but-curious. They follow the scheme honestly and do not disrupt the communication, but they are curious to infer whether the house dwellers are present at home or not by analyzing their transmission pattern. On the other hand, as shown in Fig. 1, an external adversary $\mathcal{A}$, e.g., eavesdropper, can intercept the wireless transmissions between the SMs and aggregator over a period of time to create the transmission patterns of the consumers to launch PPA. This is because AMIs mostly use wireless communications [8], [13], and the broadcast nature of these communications can facilitate intercepting signals. Also, the attackers are assumed to have old absent/present records of the consumers' transmission patterns that can be used to train a deep-learning model to launch PPA.

Basically, the objective of this article is to preserve the privacy of the consumers' activities by hiding the information that can be inferred from the consumers' transmission patterns, i.e., no one including the EU and aggregator shall be able to learn the absence of the house dwellers. This complements the works in the literature that focus on preserving privacy by hiding the power consumption readings.

IV. Preliminaries

In this section, we present a brief description of the cryptosystems and deep-learning approaches, as well as the widely used activation functions that will be used in our scheme.

A. Bilinear Pairing

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of the same prime order q and P be a generator of group $\mathbb{G}$. Suppose $\mathbb{G}$ and $\mathbb{G}_T$ are equipped with a pairing, i.e., a nondegenerated and efficiently computable bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that:

1) $\hat{e}(P, P) \neq 1_{\mathbb{G}_T}$;
2) $\hat{e}(aP_1, bQ_1) = \hat{e}(P_1, Q_1)^{ab} \in \mathbb{G}_T$, for all $a, b \in \mathbb{Z}_q^*$ and any $P_1, Q_1 \in \mathbb{G}$.

We will use bilinear pairing [22] to verify the SMs' signatures efficiently.

B. Homomorphic Encryption

Homomorphic encryption is based on the Paillier cryptosystem that allows certain mathematical operations on the plaintext to be performed directly on the ciphertext [23]. Basically, the Paillier cryptosystem is comprised of three algorithms: 1) key generation; 2) encryption; and 3) decryption.

1) *Key Generation:* To construct the public key, two large prime numbers p_1 and q_1 are first chosen, where $|p_1| = |q_1|$, and then, n and λ are computed as follows: $n = p_1 q_1$ and $\lambda = 1\ \text{cm}(p_1 - 1, q_1 - 1)$. Next, μ is calculated as follows: $\mu = L(g^\lambda \bmod n^2)^{-1} \bmod n$, where $L(u) = ([u - 1]/n)$, and a generator $g \in \mathbb{Z}_{n^2}^*$. Hence, the public

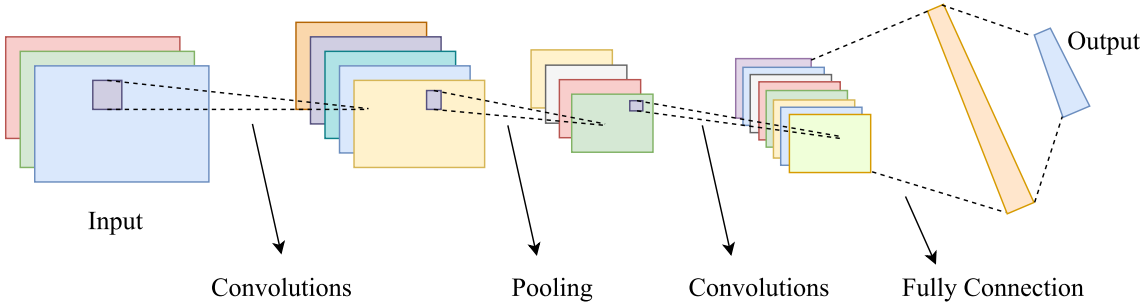

Fig. 2. Typical architecture of convolutional neural network (CNN).

key is $pk = (n, g)$, and the corresponding private key is $sk = (\lambda, \mu)$.

2) *Encryption:* Let $E(\cdot)$, $m \in \mathbb{Z}_n$, and $r \in \mathbb{Z}_n^*$ be the encryption function, a message, and a random number, respectively. The ciphertext can be calculated as follows.

$$c = E(m) = g^m \cdot r^n \bmod n^2.$$

3) *Decryption:* Given the ciphertext $c \in \mathbb{Z}_{n^2}^*$, the plaintext is

$$m = D(c) = L\left(c^{\lambda \bmod n^2}\right) \cdot \mu \bmod n.$$

Then, the additive homomorphic property is as follows:

$$\begin{aligned} E(m_1) \cdot E(m_2) &= \left(g^{m_1} \cdot r_1^n\right)\left(g^{m_2} \cdot r_2^n\right) \bmod n^2 \\ &= g^{m_1+m_2} \cdot (r_1 r_2)^n \bmod n^2 \\ &= E(m_1 + m_2). \end{aligned}$$

The homomorphic encryption is used in our scheme to aggregate the encrypted power consumption readings of the consumers in an AMI network without revealing the individual readings to preserve privacy.

C. Deep Learning

Deep learning is a special case of neural networks (NNs) in which multiple hidden layers are used. In general, the structure of NNs consists of three types of layers: 1) input; 2) output; and 3) hidden [24]. There are two types of learning, including supervised and unsupervised learning. In supervised learning, a labeled data set is used to train a model. Examples of such supervised learning approaches are the multilayer perceptron (MLP) [25], CNN [26], and recurrent neural network (RNN) [27]. Unsupervised learning, on the other hand, is used when the data set is unlabeled. Clustering is the widely used approach in unsupervised learning to cluster unlabeled data with similar features together such as K-means clustering algorithm [28].

The training of a deep-learning model starts with feeding the features/input data into the model layers. Then, these features are gradually mapped into higher level abstractions via the iterative update (also known as, feedforward and backpropagation) through the intermediate layers of the model for a predefined number of iterations. These mapping abstractions can be used to determine the decision in the output layer. The model training involves learning the weight and the bias parameters Θ by defining a cost function and selecting an optimizer. Using the backpropagation algorithm [29], the model weights and biases are updated in each iteration using the gradients of the cost function. The output values of the model are compared to the correct values for optimizing the cost function. Then, the error is fed back through the layers to adjust the weights of each connection in order to reduce the cost (loss) function [29]. For the cost function in classification tasks, *categorical cross-entropy* $C(y, \hat{y})$ is defined to measure the loss due to the difference between the true distribution y and the learned distribution $\hat{y}$, for M classes as follows:

$$C\left(y, \hat{y}\right) = \min_{\Theta}\left(-\sum_{c=1}^{M} y(c) \log\left(\hat{y}(c)\right)\right).$$

During training, an optimization method, e.g., *Adam* [30], is used for optimizing the cost function, and labeled data are used to train the model. In addition, hyperparameters of the model, such as the number of neurons in each layer, the number of layers, and type of the optimizer, can be adjusted using k-fold cross-validation, hyperopt [31], or any other validation methods [32].

In our scheme, we use CNN and gated recurrent unit neural network (GRU) to train the attacker and defense models, and use K-means clustering to label our data set.

D. Convolutional Neural Network

CNN is widely used in solving many challenging machine learning problems, such as computer vision applications, natural language processing, and speech processing [26]. This wide adoption of CNN is due to its capability of capturing complex patterns in the input. In the following, we discuss the architecture of a CNN model.

As shown in Fig. 2, a typical architecture of a CNN model has input, convolution, pooling, fully connected, and output layers. A convolution layer consists of a set of independent filters and max pooling layer(s). The convolution operation is done by sliding each filter, which is usually small in size, over the input to extract features from the input. After the convolution step, a nonlinear activation function, such as rectified linear unit (ReLU), sigmoid, or Tanh, is used to enable the model to make complex decisions and solve difficult tasks. On the other hand, a pooling layer is used to compress the output of the convolution layer by subsampling the feature maps while retaining the most important information. Convolution and pooling layers are usually followed by one or more fully

connected layers that process the features extracted to be used for inference.

E. Gated Recurrent Unit Neural Network

GRU is one type of RNNs that can memorize long sequences of input patterns by using hidden states, sometimes called hidden memory, and building connections between the internal units to form a directed graph as shown in Fig. 3 [33]. The key component in RNN is the transition function in each time step t, which takes the current time information X_t and the previous hidden state H_{t-1} and updates the current hidden state as follows:

$$H_t = F(X_t, H_{t-1}) \tag{1}$$

where F is a nonlinear transformation/activation function, e.g., sigmoid and tanh functions. Due to the recurrent structure, H_{t-1} in (1) can be regarded as a memory of previous inputs, i.e., RNN can remember previous inputs in the network's internal state. In GRU, two gates are used, namely, reset and update, to control the information flow and learn which information is important to keep and which information can be discarded. Hence, GRU has the ability to capture the correlations between the input, and that is why we will use it in training our models. GRU is usually used in the applications that need to predict the next word of a sentence based on the preceding words such as text generation [34]. It is also commonly used in speech recognition and synthesis [35].

F. K-Means Clustering Technique

The K-means clustering technique is an unsupervised learning algorithm that is used to classify a given unlabeled data set, which contains data objects/points, into a certain number of clusters based on the minimum distance between the cluster center and the object [28], [36]. A cluster is a collection of data objects that are homogeneous within the same cluster and heterogeneous to the objects in the other cluster(s) [36]. The main idea is to define k centroids, one for each cluster, and each point in the data set needs to be associated to the nearest centroid. Then, k new centroids need to be recalculated, and thus, new bindings have to be done between the same data set points and the nearest new centroids. This process should be repeated several times, and in each time, the centriods' locations are changed until convergence occurs when the centroids' locations are not changed any more. This iterative process can be summarized in Fig. 4. Basically, the clustering algorithm aims at minimizing an objective function, e.g., a squared error function, as follows:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i - c_j \right\|^2$$

where $\|x_i - c_j\|^2$ is the Euclidean distance between a data point x_i and the cluster center c_j, for a given data set of n data points.

We will use the K-means clustering technique to label the consumers' records as "absent" and "present," to be used in training our models.

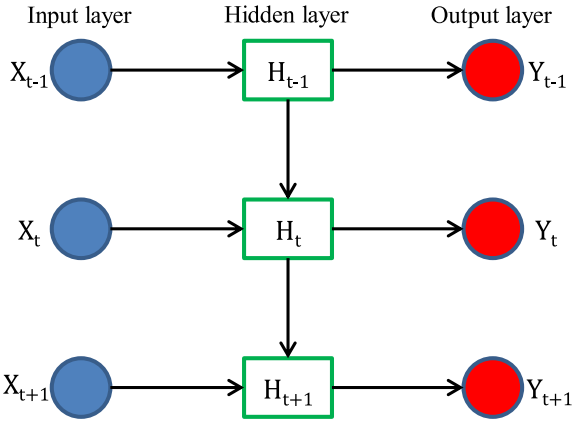

Fig. 3. Typical architecture of deep RNN.

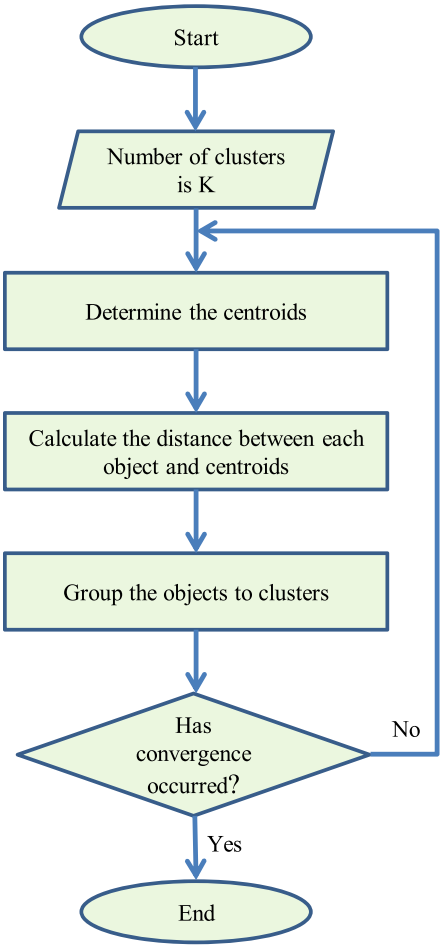

Fig. 4. Flowchart of K-means clustering technique.

G. Activation Functions

The activation functions are used in the machine learning models to transform the summed weighted input from a neuron into the activation of that neuron. They have a major impact on the model accuracy and convergence speed. In the following, we explain the widely used activation functions and their usage.

1) *ReLU:* It outputs the same input if the input is a positive value, otherwise, the output is zero [37]. ReLU is

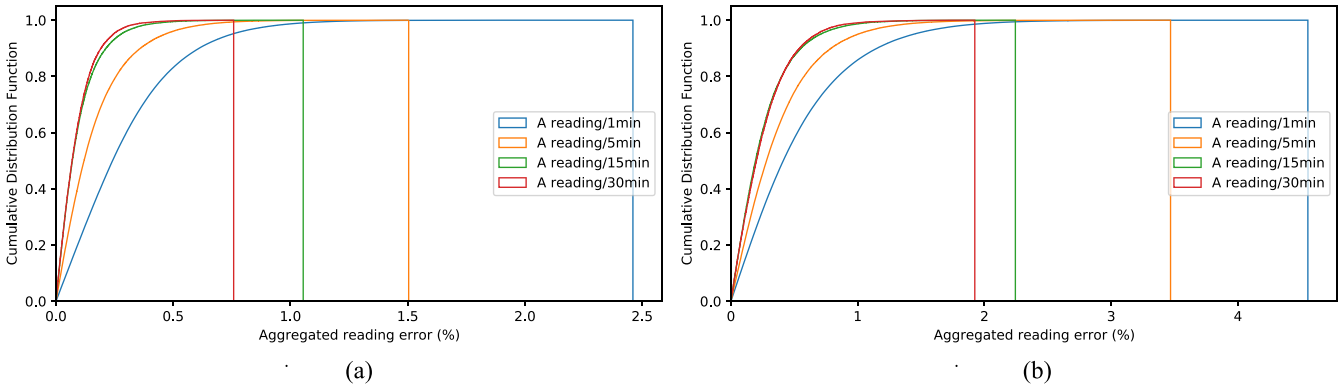

Fig. 5. Comparison of CDF of aggregated reading error for different transmission rates at (a) 5% and (b) 10% thresholds.

TABLE I
EFFICIENCY (%) ACHIEVED BY USING CAT APPROACH AT DIFFERENT THRESHOLDS AND TRANSMISSION RATES

		Threshold (%)			
		1	4	7	10
Transmission rate	1/min	59.69	72.19	75.81	77.96
	1/5min	19.99	31.72	36.48	39.75
	1/15min	6.91	13.91	18.24	21.64
	1/30min	4.75	11.92	17.17	21.77

computationally efficient since it only requires a simple max() function as follows:

$$\text{ReLU}(x) = \max(0, x).$$

2) *Softmax:* It is commonly used in the output layer for multiclass classification problems. For a given input vector, Softmax calculates a probability vector, i.e., for an input vector $\mathbf{z} = [\mathbf{z}[1], \ldots, \mathbf{z}[M]] \in \mathbb{R}^M$ of length M, where M is the number of classes, the Softmax function is defined as follows [37]:

$$\text{Softmax}(z[i]) = \frac{e^{z[i]}}{\sum_{j=1}^{M} e^{z[j]}}, \text{ for } i = \{1, \ldots, M\}.$$

V. DATA SET PREPARATION

In this article, we used a real SM data set, which is available online and released by the smart project [38]. The data set contains real power consumption readings of 114 single-family apartments over 349 days during the year 2016 from January 1st to December 14th at one-minute granularity. Hence, the total number of records is 39 786, where each record corresponds to power consumption readings of one consumer in a single day. The data set is then divided into two sets: 1) 80% for training and 2) 20% for testing. From the 1 min granularity (one reading/minute) data set, we have created other data sets with different transmission rates, including a reading every 5, 15, and 30 min, by aggregating the power consumption readings. Note that we use "1/x min" to refer to transmission rate, which means one reading is transmitted every x minutes, where x is the transmission interval. Using these data sets with thresholds of 1%, 4%, 7%, and 10%, we create data sets for the CAT approach, in which an SM reports its power consumption reading if the absolute value of the percentage of the change in the consumption compared to the last reported reading exceeds the threshold.

To measure the efficiency of the CAT approach in transmitting the fine-grained readings to the utility, we use the following formula:

$$\text{Efficiency} = \frac{N_P - N_C}{N_P} \times 100$$

where N_P is the number of transmitted readings in the case of periodic transmission of the readings and N_C is the number of transmitted readings in case of the CAT approach. Therefore, this formula computes the percentage of saving in the number of transmitted readings when using the CAT approach compared to periodic transmission of readings. Table I gives the efficiency, which can be considered the bandwidth saving achieved, at different thresholds and transmission rates. We observe that as the threshold increases, the efficiency increases since the probability that the absolute value of the percentage of the change in the power consumption exceeds the threshold decreases, and hence, there are fewer transmissions. On the other hand, as the transmission interval increases, the efficiency decreases because the probability that the absolute value of the percentage of the change in the power consumption exceeds the threshold increases, and hence, there are more transmissions.

Using the CAT approach, there may be a reading error because the reading that is considered by the EU may be less/more than the actual reading that is measured by the SM due to the use of a threshold. Therefore, the reading error at time t is $(m_{\text{SM}}[t] - m_{\text{EU}}[t])$, where $m_{\text{SM}}[t]$ is the actual power consumption reading measured by the SM at time t and $m_{\text{EU}}[t]$ is the reading that is used by the EU. Since the EU receives the aggregated reading of a number of SMs, we measured the error in the aggregated reading using our data sets assuming that the AMI network has 114 SMs. Then, we plot, in Fig. 5, the cumulative distribution function (CDF) of the aggregated reading error at 5% and 10% thresholds using different transmission rates. It can be seen that the aggregated reading error in the worst case does not exceed half of the threshold, which indicates that the error in the aggregated reading is much smaller than the error of the individual readings because the positive

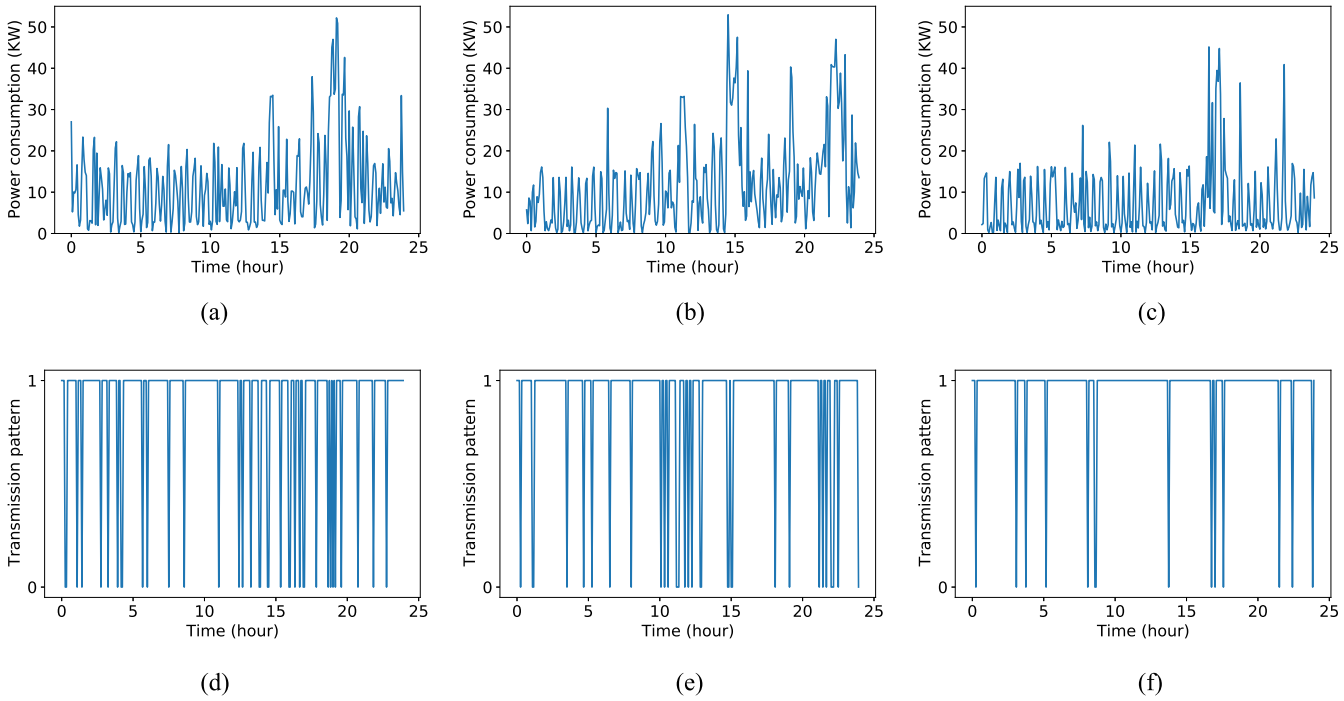

Fig. 6. Consumption and transmission patterns of different consumers when they are present at home, where in (d)–(f), 1 and 0 refer to transmission and no transmission events, respectively. (a) Consumption pattern of consumer 1. (b) Consumption pattern of consumer 2. (c) Consumption pattern of consumer 3. (d) Transmission pattern of consumer 1. (e) Transmission pattern of consumer 2. (f) Transmission pattern of consumer 3.

errors in the individual readings counter the negative errors, which can reduce the aggregated reading error. Moreover, we observe that as the transmission interval increases, the error of the aggregated reading decreases because the probability that the absolute value of the percentage of the change in the power consumption exceeds the threshold increases, and hence, there are more transmissions, which result in smaller error of the individual readings.

On the other hand, the existing power consumption data sets do not include the presence and absence of the consumers. Hence, in this section, we aim at labeling the consumers' records per day as absent or present, to be used in training our models, by using the following two approaches.

1) *Clustering:* We used the K-means clustering algorithm [28] for each consumer's record to label it absent or present. The intuition behind using K-means clustering is that when the dwellers are not at home, their transmission pattern differs from the pattern when they are at home. So a clustering approach can classify the data set records into two classes; one for "present" when the house dwellers are at home and "absent" when they are not at home. We clustered each consumer separately since each home has its own transmission pattern that depends on the lifestyle of its dwellers, the number of dwellers, etc.
2) *Periods' Comparison:* In this approach, we divide the day into three periods, t_1, t_2, and t_3, which correspond to the following time periods, 8 A.M.–4 P.M., 4 P.M.–12 A.M., and 12 A.M.-8 A.M., respectively. Then, if a consumer is present at home, the consumption at period t_2 is usually different from the consumption at other periods because the consumer does many activities at this period such as cooking, as confirmed by [8]. First, we calculate the total consumption of each period t_i, where $i = \{1, 2, 3\}$, and then, we compute the following formula for each consumer's record that has daily power consumption readings

$$\left|\frac{C_1[d] - C_2[d]}{C_1[d]}\right| + \left|\frac{C_3[d] - C_2[d]}{C_3[d]}\right|$$

where $C_i[d]$ is the total power consumption of the ith period in day d. The result of this formula is low when the consumption at periods t_1 and t_3 is close to the consumption at period t_2, which can indicate that the consumer is absent.

Finally, the records in our data sets are labeled absent if both approaches label them absent, otherwise, they are labeled present.

In this article, we study two use cases at 10% threshold: one using 1/5 min (transmitting a reading every 5 min) and another using 1/30 min (transmitting a reading every 30 min) transmission rates. Based on this threshold and transmission rates, we converted the data sets that have power consumption readings into data sets that have transmission patterns, containing zero or one in each time slot indicating no transmission event (when the absolute value of the percentage of the change in the consumption is below the threshold) or transmission event (when the absolute value of the percentage of the change in the consumption is above the threshold), respectively. Therefore, we have a transmission data set that contains consumers' records, where each record is labeled *absent* or *present*, and has 288 and 48 binary features, for 1/5 and 1/30 min transmission rates, respectively. Figs. 6 and 7 show the power consumption and transmission patterns for three different consumers in our data

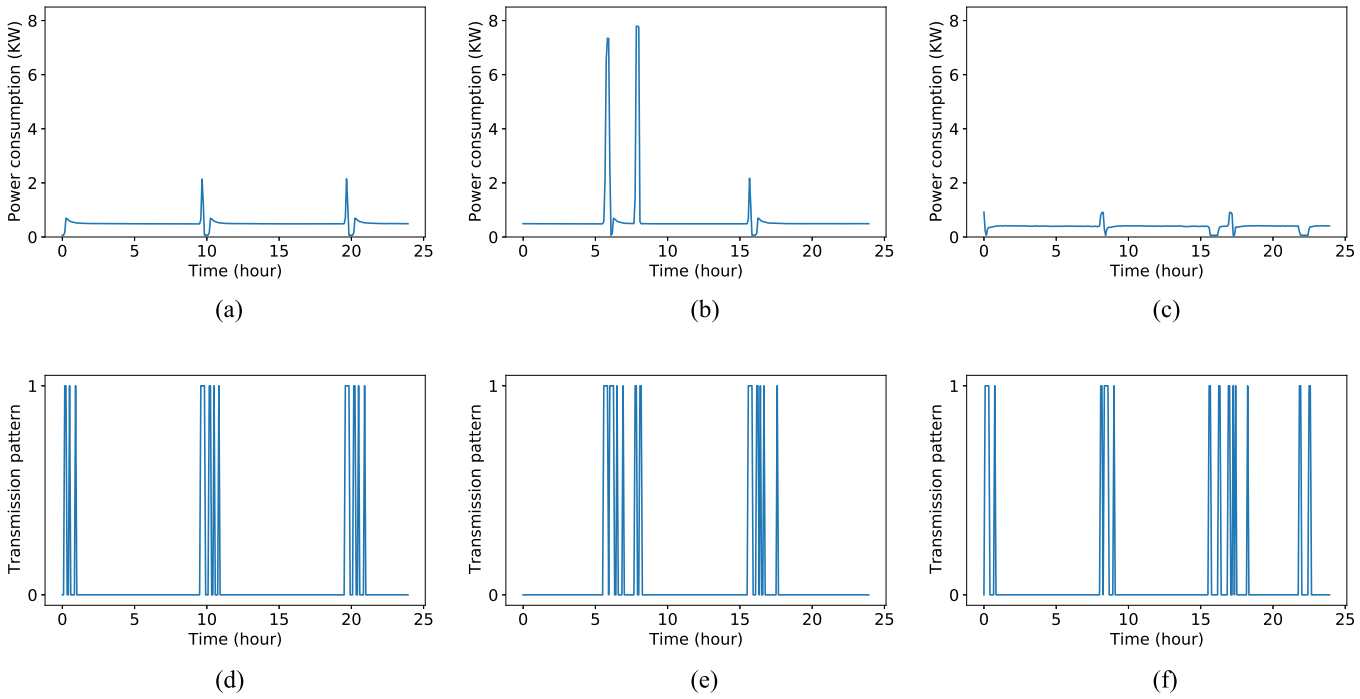

Fig. 7. Consumption and transmission patterns of different consumers when they are not at home, where in (d)–(f), 1 and 0 refer to transmission and no transmission events, respectively. (a) Consumption pattern of consumer 1. (b) Consumption pattern of consumer 2. (c) Consumption pattern of consumer 3. (d) Transmission pattern of consumer 1. (e) Transmission pattern of consumer 2. (f) Transmission pattern of consumer 3.

sets when they are present at home and absent, respectively. As can be seen in the figures, there is a substantial difference in the transmission patterns based on the presence/absence of the consumers. Attackers can exploit this difference to use the transmission patterns to learn whether a consumer is present or absent. In the next section, we will use this data set for training and evaluating both the attacker and defense models.

VI. Proposed Scheme

In this section, we first use the data set created in Section V to train an attacker model to launch PPA. The model takes the transmission pattern of a consumer as input and detects if the consumer is absent or present. Then, we train a deep-learning-based defense model that can generate spoofing transmissions (redundant real readings) to thwart PPA. Finally, we will discuss our privacy-preserving reading collection scheme.

A. Attacker Model

As explained in Section V, when a consumer is present at home, the transmission pattern is distinguishable due to the use of appliances comparing to the transmission pattern when the consumer is absent. In this section, we explain how attackers can make use of this fact by training a deep-learning model to launch PPA to learn whether a consumer is on travel (absent) or at home (present). The transmission pattern is the input of the model, and the output is either the house dwellers are present at home or absent.

1) Training the Attacker's Model: A CNN-based model is used by the attacker to capture the temporal correlation in the input sequence. While training the attacker's model, $\ell 2$-regularization is used to limit overfitting. An overfitted model is a model that performs well on the training data and it does not perform well on new/unseen data. Also, we adjust the hyperparameters of the CNN-based attacker model using hyperopt tool [31] on a validation data set to tune the number of units in each layer, batch size, learning rate, and select an activation function for each layer. Moreover, the *Adam* [30] optimizer is used to train the model for 60 epochs, 128 batch size, 0.001 learning rate, and categorical cross entropy as the loss function. To train the attacker's model, we used Python3 libraries, such as Scikit-learn [39], Numpy, TensorFlow [40], and Keras [41], which are installed on a high-performance cluster (HPC) of the Tennessee Technological University using a NVIDIA Tesla K80 GPU. The best performance of the attacker's model can be achieved using the hyperparameters given in Table II.

2) Attacker's Deep-Learning Model Architecture: In the following, we present the detailed architecture of the attacker's CNN-based model. 1-dimensional CNN model (Conv1D) is used, which consists of four convolution layers, a max pooling layer, four fully connected layers, and a softmax output layer. Table II gives the detailed structure of the attacker's model in the case of using 1/5 and 1/30 min transmission rates, including the number of layers, hidden units, and activation functions. The structure can be described as follows.

1) The input layer is the first layer of the model, and it consists of nodes, called neurons. It receives the input data and passes them to the following layers. The number of neurons in the input layer is equal to the number of readings per day (or transmission pattern size), e.g., 288

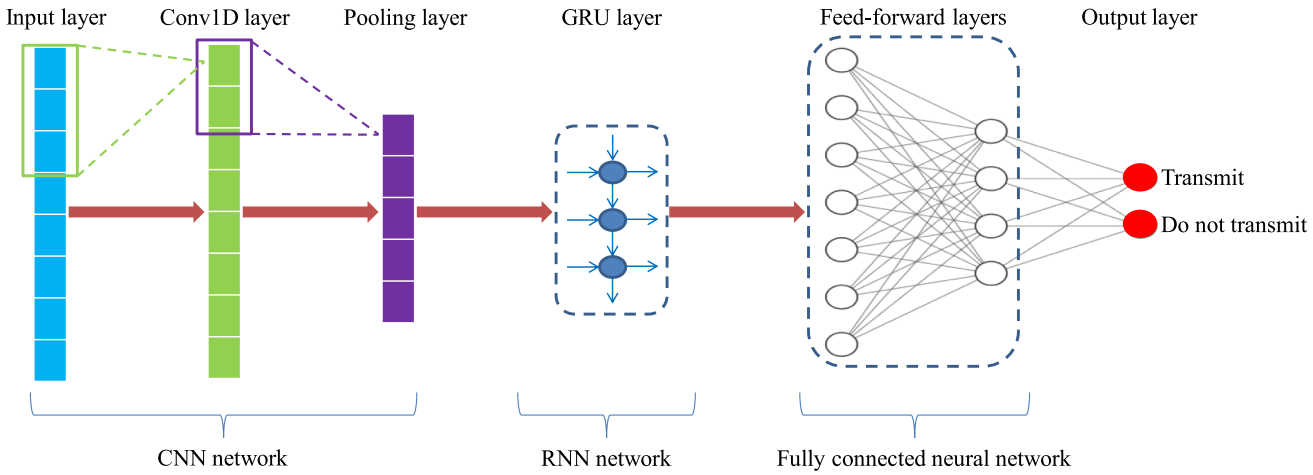

Fig. 8. Illustration of the CNN-RNN-based defense architecture.

TABLE II
ATTACKER'S MODEL ARCHITECTURE, WHERE AF STANDS FOR ACTIVATION FUNCTION, AND h_i IS THE iTH FULLY CONNECTED HIDDEN LAYER

Layer	No. of units using		AF using	
	1/5min	1/30min	1/5min	1/30min
Input	288	48	Linear	Linear
Conv1D	150	80	ELu	ReLU
Conv1D	85	32	ReLU	ReLU
Conv1D	45	20	ReLU	ELu
Conv1D	25	–	ReLU	–
MaxPooling1D	2	2	–	–
h_1	512	256	ELu	ELu
h_2	512	512	ReLU	Sigmoid
h_3	128	64	Sigmoid	ELu
h_4	64	64	ELu	ReLU
Output	2	2	Softmax	Sigmoid

readings per day if the transmission rate is one reading every 5 min.

2) 1-dimensional convolution layers are used to capture the temporal correlation in the input sequence using a number of convolution filters.
3) A max pooling layer is used to combine the features extracted from the previous layer.
4) Fully connected hidden layers are used to extract more features and make complex decisions.
5) The softmax output layer is used to classify whether the dwellers are present or absent. Since we only have two classes, the output layer has two units.

B. Defense Model

The objective of this article is to thwart transmission analysis attacks, including PPA, caused by using the CAT approach to preserve the consumers' privacy while using their fine-grained power consumption readings for load monitoring and energy management. No one, including eavesdroppers, the utility, and the aggregator, should be able to learn whether the house dwellers are present at home or absent. The key challenge is how to send spoofing transmissions so that the attacker cannot distinguish between the transmission pattern generated by our defense model when a consumer is absent and the transmission pattern when the consumer is at home. The attacker should not also identify the spoofing transmissions because they are sent only when the consumer is absent. Therefore, in this section, we explain how the defense model is trained to generate spoofing transmissions using deep-learning to countermeasure PPA.

1) Data Set Preparation for Training the Defense Model: We used the generated data set in Section V to create another data set to train the defense model as follows. First, we create a sliding window of size n elements and slide it on the transmission pattern of the consumer, so the new data set contains n elements from the transmission pattern and the label is the next element in the pattern. Note that each element in the transmission pattern is either one or zero in case there is a transmission or no transmission, respectively, in the corresponding time slot. In other words, the input of the defense model is a sequence of n binary features, zeros, and ones, and the label is the transmission decision in the next time slot, where one corresponds to transmit and zero corresponds to no transmit. By this way, our defense model acts as a predictor that takes n elements from a transmission pattern and predicts next element. Since the model is trained on transmission patterns when the consumer is present at home, the model outputs an element that makes the pattern looks like the patterns generated when the consumer is present at home.

2) Defense Model Architecture: In the following, we present the detailed architecture of the defense model. As shown in Fig. 8, the defense model is a hybrid deep-learning model that combines a CNN with GRU followed by a fully connected neural network and a softmax output layer. The input of the defense model is the transmission pattern of the last n time slots and the output is the decision of either the SM needs to send a redundant power consumption reading (spoofing transmission) or not so that the generated transmission pattern looks like the patterns transmitted when the house dwellers are present at home. The same configuration, mentioned in Section VI-A1, which is used in training the

TABLE III
DEFENSE MODEL STRUCTURE, WHERE AF STANDS FOR ACTIVATION FUNCTION, AND h_i IS THE iTH FULLY CONNECTED HIDDEN LAYER

Layer	No. of units using		AF using	
	1/5min	1/30min	1/5min	1/30min
Input	100	35	Linear	Linear
Conv1D	150	128	ReLU	ReLU
Conv1D	–	64	–	ReLU
Conv1D	–	32	–	ReLU
MaxPooling1D	4	2	–	–
GRU	200	128	Tanh	Tanh
h_1	128	128	ReLU	ReLU
h_2	32	32	ReLU	ReLU
Output	2	2	Softmax	Softmax

attacker's model, is also used in training the defense model, but with a batch size of 400 and the learning rate of 0.0001. The best performance of the defense model can be achieved using the hyperparameters given in Table III. Note that we optimized the hyperparameters of the defense model so that the generated patterns look like the patterns generated when the consumer is present at home, which reduces the attacker's success rate. However, if the hyperparameters of the defense model are not optimized, the generated patterns are not close to the present patterns, which results in higher attacker's success rate. To reduce the attacker's success rate in this case, more redundant readings (spoofing transmissions) need to be generated, which reduces the efficiency.

The main reason of using a deep-learning model is that comparing with the traditional shallow learning methods, such as logic regression, support vector machine, decision tree, etc., it can well capture the complex patterns within the data, which can improve the performance of the model. Moreover, the combination of CNN and GRU improves the extraction of the correlations between the previous transmission pattern and the current transmission decision, which is the key to improve the decision accuracy. Hence, our defense can generate transmission patterns that look like the patterns transmitted when the consumer is at home. As a result, regardless of the data set that is used, the design of our defense approach can generate similar patterns to the present ones.

3) Defense Methodology: Fig. 9 shows the defense methodology that should be run in each time slot when the consumer is absent to decide whether a spoofing transmission (redundant reading) should be sent to defend against PPA. When the absolute value of the percentage of the change in the power consumption exceeds a threshold, regardless of the presence of dwellers, the SM should send the updated power consumption reading to the aggregator. On the other hand, when the dwellers are not at home, they run the defense model to decide whether the SM should transmit redundant readings to prevent the attacker from distinguishing the generated pattern by the defense model from the transmission patterns transmitted when the consumer is present at home to preserve the consumers' privacy. A memory, as shown in Fig. 9, is used to store the transmission decision of each time slot, and it needs to keep only the last n transmission decisions.

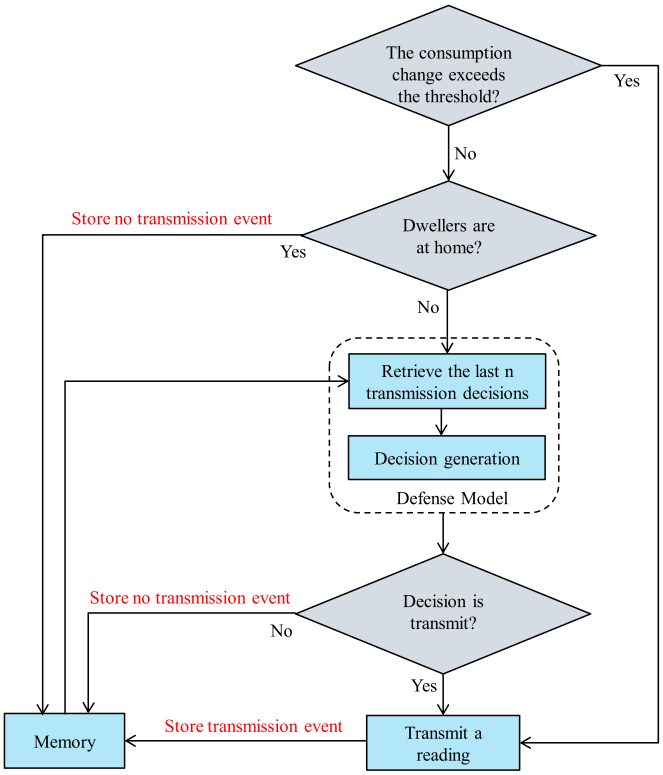

Fig. 9. Illustration of the defense methodology.

Our defense model works as follows. First, it retrieves the last n transmission decisions from the SM's memory and inputs them to the defense model, which is presented in Fig. 8. Then, the model outputs a decision, which is either "transmit" or "do not transmit" a redundant power consumption reading (a spoofing transmission) as shown in Fig. 8. We have tried different values of n and we found that the defense model gives good results at $n = 100$ and $n = 35$ in case of using 1/5 and 1/30 min transmission rates, respectively.

C. Privacy-Preserving Reading Collection Using CAT Approach

1) System Initialization: An offline KDC should bootstrap the system as follows. First, it generates the Paillier cryptosystem's public key $(n = pq, g)$, and the corresponding private key (λ, μ) where p and q are two large prime numbers with $|p| = |q|$. Moreover, it generates the bilinear pairing parameters $(q_1, \mathbb{G}, \mathbb{G}_T, P, \hat{e})$. Furthermore, it chooses a secure hash function H, where $H : \{0,1\}^* \rightarrow \mathbb{G}$. Finally, it publishes the system parameters as pubs $= \{n, g, q_1, \mathbb{G}, \mathbb{G}_T, P, \hat{e}, H\}$. In addition, each SM_i chooses a secret key $x_i \in \mathbb{Z}_q^*$, and computes the corresponding public key $Y_i = x_i P$. Similarly, the aggregator possesses private/public key pairs x_{gw}/Y_{gw}, respectively.

2) Reporting the Power Consumption Readings by SMs: When an SM_i transmits a reading either due to a change in the consumption or a redundant reading (spoofing transmission) to preserve privacy, it encrypts the power consumption reading m_i and sends it to the aggregator by performing the following steps.

1) *Step 1:* SM_i chooses a random number $r_i \in \mathbb{Z}_n^*$ and encrypts m_i using Paillier cryptosystem as follows:
$$C_i = g^{m_i} \cdot r_i^n, \quad \mathrm{mod} n^2.$$
2) *Step 2:* SM_i uses its private key x_i to generate a signature σ_i as
$$\sigma_i = x_i H(C_i \| \mathrm{TS})$$
where TS is a timestamp.
3) *Step 3:* SM_i sends the following tuple to the aggregator:
$$C_i \| \mathrm{TS} \| \sigma_i.$$

3) Aggregation by the Aggregator: The aggregator should store the last encrypted power consumption reading sent by each SM to reuse it in case that the SM does not send an encrypted reading because there is no enough change in the consumption. The aggregator should do the following steps to verify the SMs' messages and calculate the ciphertext of the total aggregated reading of all SMs.

1) *Step 1 (Verification of the Received Messages):* After receiving messages from the SMs, the aggregator first checks whether the timestamps are fresh or not. Then, it uses a batch verification approach to efficiently verify the received signatures instead of verifying the signatures individually as follows, where w is the number of messages and $w \leq |\mathbb{SM}|$:
$$\hat{e}\left(\sum_{i=1}^{w} \sigma_i, P\right) \stackrel{?}{=} \prod_{i=1}^{w} \hat{e}(H(C_i \| \mathrm{TS}), Y_i).$$
Proof of Signature Verification:
$$\begin{aligned}\hat{e}\left(\sum_{i=1}^{w} \sigma_i, P\right) &= \prod_{i=1}^{w} \hat{e}(\sigma_i, P) \\ &= \prod_{i=1}^{w} \hat{e}(x_i H(C_i \| \mathrm{TS}), P) \\ &= \prod_{i=1}^{w} \hat{e}(H(C_i \| \mathrm{TS}), x_i P) \\ &= \prod_{i=1}^{w} \hat{e}(H(C_i \| \mathrm{TS}), Y_i).\end{aligned}$$
2) *Step 2 (Privacy-Preserving Readings Aggregation):* The aggregator should compute the encrypted aggregated fine-grained reading C_{gw} as follows:
$$C_{gw} = \prod_{i=1}^{|\mathbb{SM}|} C_i, \quad \mathrm{mod} n^2$$
3) *Step 3:* The aggregator uses its private key x_{gw} to compute the signature as follows:
$$\sigma_{gw} = x_{gw} H\big(C_{gw} \| \mathrm{TS}\big).$$
4) *Step 4:* The aggregator composes a message containing the encrypted aggregated reading and a signature and sends it to the EU. The message should have the following tuple:
$$C_{gw} \| \mathrm{TS} \| \sigma_{gw}.$$

4) Aggregated Reading Recovery by the EU: Upon receiving the message from the aggregator, the EU checks the freshness of the timestamp and verifies the signature by checking the following:
$$\hat{e}\big(\sigma_{gw}, P\big) \stackrel{?}{=} \hat{e}\big(H\big(C_{gw} \| \mathrm{TS}\big), Y_{gw}\big).$$
Proof of Signature Verification:
$$\begin{aligned}\hat{e}\big(\sigma_{gw}, P\big) &= \hat{e}\big(x_{gw} H\big(C_{gw} \| \mathrm{TS}\big), P\big) \\ &= \hat{e}\big(H\big(C_{gw} \| \mathrm{TS}\big), x_{gw} P\big) \\ &= \hat{e}\big(H\big(C_{gw} \| \mathrm{TS}\big), Y_{gw}\big).\end{aligned}$$

Then, the EU uses the secret key (λ, μ) to decrypt C_{gw}, and recover the total power consumption reading of the SMs R_T as follows:
$$R_T = D\big(C_{gw}\big) = L\Big(C_{gw}^{\lambda} \bmod n^2\Big) \cdot \mu \bmod n = \sum_{i=1}^{|\mathbb{SM}|} m_i.$$

VII. Evaluations

In this section, we first discuss the performance metrics and then evaluate the attacker's success rate in determining the absence of the house dwellers by analyzing the transmission pattern without using our defense. Next, we evaluate the success of the attacker while using our defense model assuming that the attacker does not know the defense model. Finally, we evaluate the attacker's success assuming that the attacker knows the defense model.

A. Performance Metrics

In order to evaluate our models' performance, we consider the following metrics. The success rate (SR) is the probability that the attacker successfully detects that the house dwellers are not at home (absent). The false alarm rate (FA) is the probability that the attacker thinks that the dwellers are absent but they are actually present at home. The defense model performance is better when SR is low. These metrics are measured as follows:
$$\mathrm{SR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad \mathrm{FA} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FN}}$$
where, TP, TN, FN, and FP stand for true positive, true negative, false negative, and false positive, respectively. Moreover, we measure also the efficiency of the CAT approach with and without our defense in terms of the percentage of readings that are not transmitted, which is related to the saving in the communication bandwidth, compared to the continuous transmission approach. The efficiency is computed as follows:
$$\mathrm{Efficiency}(\%) = \frac{P_t - R_t}{P_t} \times 100$$
where P_t is the number of readings that are sent when SMs send the readings continuously (without using the CAT approach) and R_t is the number of readings sent by our scheme that uses the CAT approach.

B. Attacker's Success Rate Without Our Scheme

As mentioned in Section III-B, the attacker has old absent/present records of the transmission patterns of the consumers. These records are used to train a deep-learning model, which is discussed in Section VI-A, to infer the absence of house dwellers by running the model using the consumer's transmission pattern. Using this model and without using our defense model, the attacker's success rates are 91.1% and 88.5% in the case of 1/5 and 1/30 min transmission rates, respectively, as can be seen in Table IV. The reduction in the success rate with the increasing of the transmission interval is because as the transmission interval increases, the number of transmissions decreases, which results in hiding detailed information from the consumption pattern. From the given results, we can conclude that the attacker can infer the absence of the dwellers with high confidence.

C. Attacker's Success Rate With Our Scheme if the Defense Model Is Unknown

In this section, we evaluate the performance of our defense model assuming that the attacker does not know the defense model. The attacker uses the transmission patterns generated by our defense model as input to the attacker model, discussed in Section VI-A, to know whether the house dwellers are absent. As shown in Table IV, the attacker's success rate is significantly reduced from 91.1%, without using our defense model, to only 3.15% using our defense with 1/5 min transmission rate, while the attacker's success rate is reduced from 88.5% to 2.25% with 1/30 min transmission rate. This big reduction in the success rate of the attacker indicates that our defense model can generate patterns that look like the patterns transmitted when the consumer is at home, and that is why the attacker's model was not able to classify the patterns generated by our defense model as absent.

Moreover, using 1/5 min transmission rate at 10% threshold, our scheme can achieve efficiency of 40.8% compared to 69.67% in the case that privacy is not preserved. Also, using 1/30 min transmission rate, the efficiency is reduced from 44.6% to 19.01% using our defense, as shown in Table IV. This reduction in efficiency is due to sending redundant readings to preserve privacy, and this can be considered the cost for privacy preservation. Note that better efficiency can be achieved by reducing the number of redundant readings but with higher attacker's success, i.e., there is a tradeoff between the efficiency and attacker's success rate. From our experiment results, we can see that the less the transmission rate (longer transmission interval), the more reduction in efficiency since the likelihood that the power consumption change exceeds the threshold increases, which causes more transmissions.

D. Attacker's Success Rate With Our Scheme if the Defense Model Is Known

In this section, we evaluate the performance of our scheme when the attacker knows the defense model. In Sections VII-B and VII-C, we assume that the attacker has old absent/present records of the transmission patterns of the consumers, and using these records, the attacker trained a deep-learning model to learn whether a consumer is absent. In this section, we assume a stronger attacker who knows not only old transmission patterns but also the defense model. Using the available knowledge to the attacker, he can launch powerful attacks as follows. First, the attacker uses the absent records and the defense model to create "spoofing" patterns. Then, using absent, present, and spoofing patterns, the attacker trains an advanced model to classify the input patterns into "absent," "present," or spoofing. Hence, if the classification is spoofing, the attacker thinks that the victim consumer is absent and the defense model is used to generate patterns with spoofing transmissions.

TABLE IV
EVALUATIONS WITH AND WITHOUT OUR DEFENSE MODEL FOR 1/5 AND 1/30 MIN TRANSMISSION RATES AT 10% THRESHOLD

Transmission rate	1/5min	1/30min
Attacker's success rate without our scheme	91.1%	88.5%
Attacker's success rate with our scheme	**3.15%**	**2.25%**
Efficiency without our scheme	69.67%	44.6%
Efficiency with our scheme	**40.8%**	**19.01%**

TABLE V
ATTACKER'S MODEL ARCHITECTURE WHEN THE DEFENSE MODEL IS KNOWN TO THE ATTACKER, WHERE AF STANDS FOR ACTIVATION FUNCTION, AND h_i IS THE iTH FULLY CONNECTED HIDDEN LAYER

Layer	No. of units using		AF using	
	1/5min	1/30min	1/5min	1/30min
Input	288	48	Linear	Linear
Conv1D	150	128	ReLU	ReLU
MaxPooling1D	4	4	–	–
GRU	–	32	–	Tanh
h_1	128	64	ReLU	ReLU
h_2	35	32	ReLU	ReLU
Output	3	3	Softmax	Softmax

The best performance of the attacker's model, when the defense model is known in case of using 1/5 and 1/30 min transmission rates, can be achieved using the hyperparameters given in Table V. By knowing the defense model by the attacker, the success rate of detecting the absence of the house dwellers increases from 3.15% to 13.52% and from 2.25% to 12.84% using 1/5 and 1/30 min transmission rates, respectively. These results are good given the strong assumptions that the attacker knows the defense model and has old transmission patterns.

In addition, Figs. 10 and 11 show the receiver operating characteristic (ROC) curves of the attacker model in case of 1/5 and 1/30 min transmission rates, respectively, assuming that the defense model is known. The ROC curve is often used to evaluate the classification accuracy, which is measured by the area under the curve (AUC). This area indicates how the model can distinguish between the classes, where a bigger AUC indicates a better performance. We assume that a false alarm rate of 0.05 is acceptable to the attacker, and then, we need to determine the success rate at this false alarm rate. As

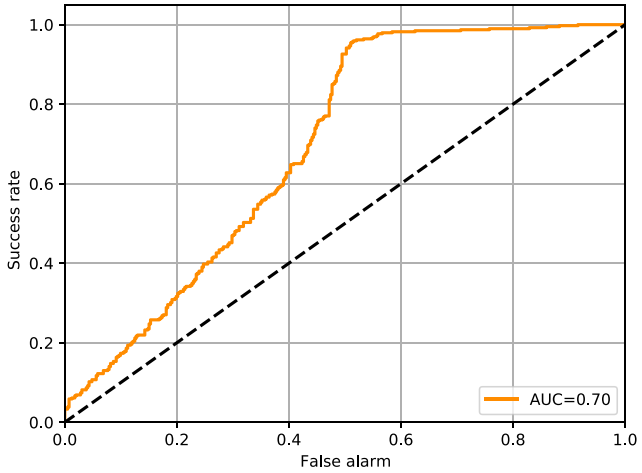

Fig. 10. ROC curve of the attacker model when the defense model is known using 1/5 min transmission rate at 10% threshold.

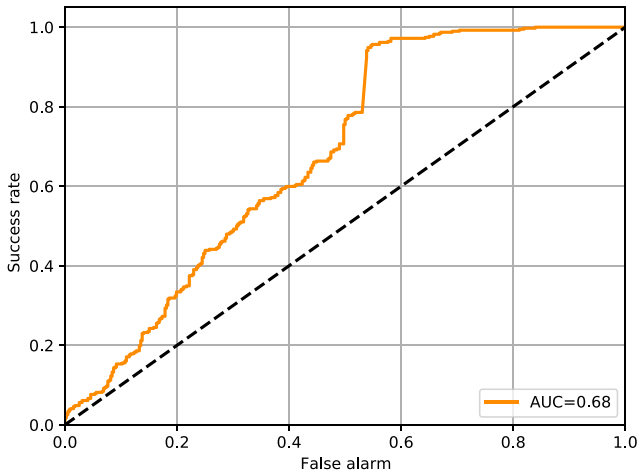

Fig. 11. ROC curve of the attacker model when the defense model is known using 1/30 min transmission rate at 10% threshold.

can be seen in Figs. 10 and 11, at false alarm of 0.05, the attacker's success rate is fairly low (less than 0.1) compared to more than 0.6 using the ASP proposed in [8]. Therefore, the attacker can detect the absence of the house dwellers with a high probability in ASP [8], while the success rate in our scheme is much lower. This is because our scheme is trained on the transmission patterns when the consumer is present at home, and then it uses spoofing transmissions (redundant real readings) when the consumer is absent to generate a pattern that looks similar to the consumer's transmission patterns when he/she is at home.

E. Discussion

In this article, we consider a strong attack model assuming that the attacker has old transmission patterns and knows the defense model. Using this knowledge, the attacker trains a sophisticated deep-learning model to detect the absence of the house dwellers. On the other hand, the design of our defense scheme is based on a deep-learning model, which is trained on the transmission patterns when the consumers are present at home to generate spoofing transmissions so that the attacker cannot distinguish the pattern generated using STDL from the patterns transmitted when the consumers are at home to preserve consumers' privacy. The scheme in [8] is customized, i.e., the scheme depends on the distribution of the power consumption changes (transmission patterns) of each consumer, but our model is general (one model for all consumers) and it can be applied to new consumers who have no old transmission patterns. Furthermore, we assume that the network nodes, including the SMs, EU, and aggregator, are honest but curious, while the scheme in [8] assumes that they are trusted and the defense is limited only to the eavesdroppers.

Moreover, our scheme can achieve the following desirable privacy requirements that can counter the attacks mentioned in Section III-B.

PPA Countermeasure: An adversary can eavesdrop on the wireless transmissions between the SMs and the aggregator and construct the transmission patterns of the consumers, and then launch PPA. On the other hand, the aggregator and EU are curious to learn whether the house dwellers are absent by analyzing their SMs' transmission patterns. However, using our scheme, no one can learn whether the house dwellers are absent because it sends spoofing transmissions (redundant real readings) to make it difficult for the attacker to learn this information.

Consumers' Fine-Grained Readings' Privacy Preservation: To ensure the privacy of the consumers, each SM first encrypts its readings using the public key of the Paillier cryptosystem and no entity is able to learn the individual readings to preserve consumers' privacy. In addition, if the same reading is repeated at different times, the ciphertext looks different because the encryption uses a one-time random number. Also, when an SM sends a redundant reading that is similar to the last reported reading to preserve privacy, the ciphertext looks different and the aggregator cannot know that the reading is redundant. This is important to prevent the aggregator and eavesdroppers from identifying redundant readings (spoofing transmissions) because these readings are sent only when the house dwellers are absent. It is important to use the random number r once because if it is reused, the difference between two readings, i.e., m_1 and m_2, can be obtained by dividing their ciphertexts $g^{m_1} \cdot r^n / g^{m_2} \cdot r^n = g^{m_1 - m_2}$, and this can be solved to obtain $m_1 - m_2$; hence, by knowing one reading, the other one can be obtained.

Confidentiality of AMI's Total Power Consumption: The aggragator should send the ciphertext of the AMI's total power consumption to the EU for load monitoring, after computing it from the encrypted fine-grained power consumption readings of the SMs. The aggregator and the attackers, who eavesdrop on the communications between the aggregator and the EU, learn nothing about the total consumption of an AMI since the private key which is known only to the EU is needed for decrypting Paillier ciphertext to obtain the aggregated power consumption reading.

VIII. Conclusion

In this article, we have proposed STDL, an efficient and privacy-preserving scheme for collecting power consumption

readings in AMI networks. Our scheme is efficient because the SMs do not send the readings if there is no enough change in the power consumption comparing to the last report reading. It also preserves the consumers' privacy by transmitting redundant readings (spoofing transmissions) using a deep-learning-based defense scheme. First, a real data set for power consumption readings data set and clustering technique is used to create a data set for transmission patterns using the CAT approach. Then, we train an attacker model using deep learning, and our evaluations indicate that the attacker's success rate is about 91%. Next, we train a deep-learning-based defense scheme to send spoofing transmissions efficiently to thwart PPA attacks. Extensive evaluations are conducted, and the results indicate that our scheme can reduce the attacker's success rate to 13.52% in case he knows the defense model and to 3.15% in case he does not know the model, while still achieving high efficiency in terms of the number of readings that should be transmitted. This big reduction in the attacker's success rate indicates that our defense can generate patterns that look like the patterns transmitted when the consumer is at home. Moreover, our measurements indicate that the proposed scheme can reduce the number of readings that should be transmitted by about 41% while preserving the privacy of the consumers.

ACKNOWLEDGMENT

The authors, therefore, acknowledge with thanks Deanship of Scientific Research for technical and financial support.

REFERENCES

[1] V. C. Gungor *et al.*, "A survey on smart grid potential applications and communication requirements," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 28–42, Feb. 2013.

[2] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 60–65, Apr. 2011.

[3] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 238–243.

[4] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[5] H. Mohammed, S. Tonyali, K. Rabieh, M. Mahmoud, and K. Akkaya, "Efficient privacy-preserving data collection scheme for smart grid AMI networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[6] M. I. Ibrahem, M. M. Badr, M. M. Fouda, M. Mahmoud, W. Alasmary, and Z. M. Fadlullah, "PMBFE: Efficient and privacy-preserving monitoring and billing using functional encryption for AMI networks," in *Proc. IEEE Int. Symp. Netw. Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–9.

[7] M. I. Ibrahem, M. Nabil, M. M. Fouda, M. Mahmoud, W. Alasmary, and F. Alsolami, "Efficient privacy-preserving electricity theft detection with dynamic billing and load monitoring for AMI networks," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1243–1258, Jan. 2021, doi: 10.1109/JIOT.2020.3026692.

[8] H. Li, S. Gong, L. Lai, Z. Han, R. C. Qiu, and D. Yang, "Efficient and secure wireless communications for advanced metering infrastructure in smart grids," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1540–1551, Sep. 2012.

[9] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security Privacy*, vol. 8, no. 1, pp. 11–20, Feb. 2010.

[10] S. Finster and I. Baumgart, "Privacy-aware smart metering: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 1088–1101, 2nd Quart., 2015.

[11] K. Weaver. *A Perspective on How Smart Meters Invade Individual Privacy*. Accessed: 2020. [Online]. Available: https://skyvisionsolutions.files.wordpress.com/2014/08/utility-smart-meters-invade-privacy-22-aug-2014.pdf

[12] S. Yussof, M. E. Rusli, Y. Yusoff, R. Ismail, and A. A. Ghapar, "Financial impacts of smart meter security and privacy breach," in *Proc. Int. Conf. Inf. Technol. Multimedia*, Nov. 2014, pp. 11–14.

[13] M. Alam, M. Reaz, and M. Ali, "A review of smart homes—Past, present, and future," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1190–1203, Nov. 2012.

[14] K. Samarakoon, J. Ekanayake, and N. Jenkins, "Reporting available demand response," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 1842–1851, Dec. 2013.

[15] K. Carrie Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? The case of electricity," *Energy Policy*, vol. 52, pp. 213–234, Jan. 2013.

[16] S. Werner and J. Lunden, "Event-triggered real-time metering in smart grids," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2015, pp. 2701–2705.

[17] S. Werner and J. Lundén, "Smart load tracking and reporting for real-time metering in electric power grids," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1723–1731, May 2016.

[18] J. Lunden and S. Werner, "Real-time smart metering with reduced communication and bounded error," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Nov. 2014, pp. 326–331.

[19] M. Simonov, G. Chicco, and G. Zanetto, "Event-driven energy metering: Principles and applications," *IEEE Trans. Ind. Appl.*, vol. 53, no. 4, pp. 3217–3227, Aug. 2017.

[20] A. Proano, L. Lazos, and M. Krunz, "Traffic decorrelation techniques for countering a global eavesdropper in WSNs," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 857–871, Mar. 2017.

[21] S. Alsemairi and M. Younis, "Adaptive packet-combining to counter traffic analysis in wireless sensor networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2015, pp. 337–342.

[22] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proc. Adv. Cryptol. CRYPTO*, 2001, pp. 213–229.

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

[24] Z. Zheng, Y. Yang, X. Niu, H. Dai, and Y. Zhou, "Wide and 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.

[25] S. Haykin, *Neural Networks and Learning Machines: A Comprehensive Foundation*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, Nov. 2008.

[26] Y. LeCun *et al.*, "Convolutional networks for images, speech, and time series," *Handbook Brain Theory Neural Netw.*, vol. 3361, no. 10, pp. 255–258, 1995.

[27] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2450–2462.

[28] K. M. Faraoun and A. Boukelif, "Neural networks learning improvement using the *k*-means clustering algorithm to detect network intrusions," *Int. J. Comput. Inf. Eng.*, vol. 1, no. 10, pp. 3151–3158, 2007.

[29] S. Chaturvedi, R. N. Titre, and N. Sondhiya, "Review of handwritten pattern recognition of digits and special characters using feed forward neural network and Izhikevich neural model," in *Proc. Int. Conf. Electron. Syst. Signal Process. Comput. Technol.*, Jun. 2014, pp. 425–428.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–6.

[31] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: A Python library for model selection and hyperparameter optimization," *Comput. Sci. Disc.*, vol. 8, no. 1, 2015, Art. no. 014008. [Online]. Available: https://doi.org/10.1088/1749-4699/8/1/014008

[32] M. Nabil, M. Ismail, M. Mahmoud, M. Shahin, K. Qaraqe, and E. Serpedin, "Deep recurrent electricity theft detection in AMI networks with random tuning of hyper-parameters," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 740–745.

[33] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," 2017. [Online]. Available: arXiv:1702.01923.

[34] A. F. Ganai and F. Khursheed, "Predicting next word using RNN and LSTM cells: Stastical language modeling," in *Proc. Int. Conf. Image Inf. Process. (ICIIP)*, Nov. 2019, pp. 469–474.

[35] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2013, pp. 6645–6649.
[36] B. Dash, D. Mishra, A. Rath, and M. Acharya, "A hybridized *k*-means clustering approach for high dimensional dataset," *Int. J. Eng. Sci. Technol.*, vol. 2, no. 2, pp. 59–66, 2010.
[37] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018. [Online]. Available: arXiv:1811.03378.
[38] Kolter. *Residential Energy Disaggregation Dataset (REDD)*. Accessed: 2020. [Online]. Available: http://traces.cs.umass.edu/index.php/Smart/Smart
[39] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[40] A. Martín *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/
[41] F. Chollet *et al.* (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras

Mohamed I. Ibrahem received the B.S. and M.S. degrees in electrical engineering (electronics and communications) from Benha University, Cairo, Egypt, in 2014 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN, USA.

He also holds the position of Lecturer assistant with Benha University. His research interests include privacy preserving and security schemes for smart grid AMI networks.

Mr. Ibrahem received the Eminence Award for the Doctor of Philosophy Best Paper from Tennessee Technological University.

Mohamed Mahmoud (Senior Member, IEEE) received the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in April 2011.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Tennessee Technological University, Cookeville, TN, USA. He is the author of more than 100 papers published in IEEE conferences and journals, such as INFOCOM conference, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON MOBILE COMPUTING, and IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. His research interests include security and privacy-preserving schemes for smart grid and intelligent transportation systems.

Dr. Mahmoud has received the NSERC-PDF Award. He won the Best Paper Award from IEEE International Conference on Communications, Dresden, Germany, in 2009. He serves as an Associate Editor for IEEE INTERNET OF THINGS JOURNAL and *Peer-to-Peer Networking and Applications* (Springer).

Mostafa M. Fouda (Senior Member, IEEE) received the Ph.D. degree in information sciences from Tohoku University, Sendai, Japan, in 2011.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA. He also holds the position of an Associate Professor with Benha University, Cairo, Egypt. He has served as an Assistant Professor with Tohoku University. He was a Postdoctoral Research Associate with Tennessee Technological University, Cookeville, TN, USA. He has more than 60 publications in international conferences, journal papers, and book chapters. His research interests include cyber security, machine learning, blockchain, IoT, 5G networks, smart healthcare, and smart grid communications.

Dr. Fouda is an Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and an Associate Editor of IEEE ACCESS.

Fawaz Alsolami (Member, IEEE) received the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2008, and the Ph.D. degree in computer science from the King Abdullah University of Science and Technology University, Thuwal, Saudi Arabia, in 2016.

He joined the Computer Science Department, King Abdulaziz University, Jeddah, Saudi Arabia, as an Assistant Professor of Computer Science. Since 2018, he has been the Chairman of the Computer Science Department, King Abdulaziz University. His research interests are artificial Intelligence, machine learning and data mining, and combinatorial optimization.

Waleed Alasmary (Senior Member, IEEE) received the B.Sc. degree (Hons.) in computer engineering from Umm Al-Qura University, Mecca, Saudi Arabia, in 2005, the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2010, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2015.

During his Ph.D. degree, he was a Visiting Research Scholar with the Network Research Laboratory, University of California at Los Angeles, Los Angeles, CA, USA, in 2014. He was a Fulbright Visiting Scholar with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA, from 2016 to 2017. He subsequently joined the College of Computer and Information Systems, Umm Al-Qura University as an Assistant Professor of Computer Engineering, where he currently holds an Associate Professor position.

Dr. Alasmary is currently an Associate Editor for *Array*.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular *ad hoc* and sensor networks.

Dr. Shen received the R.A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) presents in 2019, the Excellent Graduate Supervision Award in 2006 from the University of Waterloo, and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for the IEEE Globecom, IEEE Infocom, and IEEE VTC. He is the elected IEEE Communications Society Vice President for Technical and Educational Activities, the Vice President for Publications, the Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and the member of IEEE Fellow Selection Committee. He was/is the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, *IET Communications*, and *Peer-to-Peer Networking and Applications*. He is a registered Professional Engineer of Ontario, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.