# Privacy-Preserving and Communication-Efficient Energy Prediction Scheme Based on Federated Learning for Smart Grids

Badr, Mahmoud M.; Mahmoud, Mohamed M. E. A.; Fang, Yuguang; Abdulaal, Mohammed; Aljohani, Abdulah Jeza; Alasmary, Waleed; Ibrahem, Mohamed I.

**Published in:**
IEEE Internet of Things Journal

**Published: 01/05/2023**

**Document Version:**
Post-print, also known as Accepted Author Manuscript, Peer-reviewed or Author Final version



**Published version (DOI):**
10.1109/JIOT.2022.3230586

**Publication details:**
Badr, M. M., Mahmoud, M. M. E. A., Fang, Y., Abdulaal, M., Aljohani, A. J., Alasmary, W., & Ibrahem, M. I. (2023). Privacy-Preserving and Communication-Efficient Energy Prediction Scheme Based on Federated Learning for Smart Grids. *IEEE Internet of Things Journal*, *10*(9), 7719-7736. https://doi.org/10.1109/JIOT.2022.3230586

Badr, M. M., Mahmoud, M. M. E. A., Fang, Y., Abdulaal, M., Aljohani, A. J., Alasmary, W., & Ibrahem, M. I. (2023). Privacy-Preserving and Communication-Efficient Energy Prediction Scheme Based on Federated Learning for Smart Grids. *IEEE Internet of Things Journal, 10*(9), 7719-7736. https://doi.org/10.1109/JIOT.2022.3230586.

# Privacy-Preserving and Communication-Efficient Energy Prediction Scheme Based on Federated Learning for Smart Grids

Mahmoud M. Badr, Mohamed Mahmoud, *Senior Member, IEEE,* Yuguang Fang, *Fellow, IEEE,* Mohammed Abdulaal, Abdulah J. Aljohani, *Senior Member, IEEE,* Waleed Alasmary, *Senior Member, IEEE,* and Mohamed I. Ibrahem

***Abstract*—Energy forecasting is important because it enables infrastructure planning and power dispatching while reducing power outages and equipment failures. It is well-known that federated learning (FL) can be used to build a global energy predictor for smart grids without revealing the customers' raw data to preserve privacy. However, it still reveals local models' parameters during the training process, which may still leak customers' data privacy. In addition, for the global model to converge, it requires multiple training rounds, which must be done in a communication-efficient way. Moreover, most existing works only focus on load forecasting while neglecting energy forecasting in net-metering systems. To address these limitations, in this paper, we propose a privacy-preserving and communication-efficient FL-based energy predictor for net-metering systems. Based on a dataset for real power consumption/generation readings, we first propose a multi-data-source hybrid deep learning (DL)-based predictor to accurately predict future readings. Then, we repurpose an efficient inner-product functional encryption (IPFE) scheme for implementing secure data aggregation to preserve the customers' privacy by encrypting their models' parameters during the FL training. To address communication efficiency, we use a change and transmit (CAT) approach to update local model's parameters, where only the parameters with sufficient changes are updated. Our extensive studies demonstrate that our approach accurately predicts future readings while providing privacy protection and high communication efficiency.**

***Index Terms*—Energy prediction, privacy preservation, communication efficiency, federated learning, and Smart grids.**

## I. Introduction

Smart grid is a contemporary paradigm for a clean, reliable, and intelligent power system. It includes an advanced metering infrastructure (AMI) consisting of a set of smart meters (SMs) deployed at the customers' premises. The electric utility server uses this infrastructure to collect fine-grained energy consumption readings for billing, load monitoring, and energy forecasting [1]. Moreover, to reduce greenhouse gas emissions in smart grid and generate more green energy, a utility server may incentivize its customers to install renewable resources, such as solar panels on their rooftops, and sell the excess generated energy by injecting it directly into its grid [2]. This system is called *net-metering*, in which the SMs report *net readings* which are the difference between the consumed and generated energy instead of reporting only the energy consumption readings [3], [4]. Hence, the reported reading can be positive, i.e., when the energy consumed by a customer is higher than the generated energy, or negative, i.e., when the generated energy is higher than the consumed energy. The net-metering system is adopted in various countries, such as the USA, Italy, and Brazil, due to its advantages [3].

Energy prediction has a pivotal role in the development of a smart grid since it enables effective and efficient management of the generated energy [5]. This is because predicting the future energy demand helps in properly fine-tuning the plan and operation mode of the power system by determining the amount of energy that should be generated and distributed to end customers [6]. Moreover, it does not only provide a good vision for infrastructure planning, marketing, and power dispatching, but also reduces the economical loss, power outages, and equipment damage/failures by optimizing the system's operations. Hence, to achieve the aforementioned improvements, *an accurate energy forecasting model is needed, which requires collecting the fine-grained meter readings of the customers* [5].

In order to predict future energy demand, different solutions have been proposed in the literature, including statistical methods, such as Auto-Regressive Integrated Moving Average (ARIMA) [7], and machine learning methods [5], [8]–[12]. Among existing solutions, it has been found that deep learning (DL) solutions are the most accurate because they can extract and learn the temporal and non-linear correlations within the energy readings, and hence provide accurate energy prediction [11], [13]. Besides, to create an accurate energy predictor, the data used to train the model should be collected from different households to increase its size and diversity. To do that, most of the existing solutions in the literature use the energy readings of different customers to train the predictor [14], [15]. However, *this approach may violate the customers' privacy*

Corresponding author: Mohamed Mahmoud.

M. Badr is with the Department of Network and Computer Security: Cybersecurity, College of Engineering, SUNY Polytechnic Institute, Utica, NY 13502, USA, and the Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11629, Egypt (e-mail: badrm@sunypoly.edu).

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

Y. Fang is with the Department of Electrical and Computer Engineering at University of Florida, Gainesville, FL 32611, USA (e-mail: fang@ece.ufl.edu).

M. Abdulaal and A. Aljohani are with the Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mails: mjabdulaal@kau.edu.sa and ajaljohani@kau.edu.sa).

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

M. Ibrahem is with the Department of Cyber Security Engineering, George Mason University, Fairfax, VA 22030, USA, and the Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo 11629, Egypt (e-mail: mibrahem@gmu.edu).

*as the fine-grained readings may be used to reveal sensitive information about the customers* [16]–[18]. This information can be the appliances being used, when occupants sleep, return home, or whether they are on vacation, etc, which may be abused, e.g., for criminal activities such as burglary [16].

Very few works in the literature have investigated this privacy problem by using the emerging federated learning (FL) [6], [17]. The fundamental idea of FL is to build a global model without disclosing the customers' raw data. More specifically, in FL, edge nodes, at the customer-side, use their data to train local machine learning models. Then, they send the updated local models' parameters to a centralized utility server to aggregate them and update the global model for energy prediction. The design goal is to achieve the following properties: (1) the customers' private data are not directly revealed; (2) the amount of data transmitted between the customers' edge nodes and the utility server is lowered since only the updated model's parameters are sent instead of sending a massive amount of customers' readings; and (3) the burden of storing a massive amount of customers' data by the utility server to train the model is reduced.

Despite the aforementioned advantages achieved by using the FL, *the proposed solutions in [6], [17] are still vulnerable to privacy attacks* because the model's parameters may enable adversaries to learn private information about customers by launching attacks such as membership inference [19] and model inversion [20]. Although privacy-preserving data aggregation schemes [21] can be used to hide the local models' parameters and enable the utility server and customers to compute a global model, the existing schemes are not communication-efficient because they produce large-sized ciphertexts. Moreover, *the proposed solutions in [6], [17] are not communication-efficient* because they adopt the basic FL approach in which all the updated model's parameters are sent to the aggregation server in every round.

Moreover, all the above works [5]–[15], [17] only investigate energy predictors for consumption metering systems, where renewable energy generators are not deployed at the customer-side and SMs only report energy consumption readings, and *none of them investigate net-energy predictors for net-metering systems*. The problem is different and more challenging in net-metering systems because the predictor, in this case, not only depends on the customers' consumption behavior as in consumption metering systems, but it also depends on other factors such as the solar irradiance. In other words, net readings depend on both energy consumption and generation patterns simultaneously which should be taken into consideration in the predictor design. Although Razavi *et. al.* [22] have compared the performance between consumed energy and net-energy forecasting, the customers' privacy preservation is largely ignored in their work. In addition, the correlation between net readings and solar irradiance has not been investigated.

Therefore, in this paper, we address the aforementioned limitations by investigating *how to enable the utility server to efficiently build a global predictor for net-metering systems to accurately predict future net readings without learning neither the customers' training data nor the parameters of the local models* to preserve the customers' privacy. To do that, we design a multi-data-source hybrid DL-based predictor that learns the correlation between net readings and solar irradiance to make more accurate prediction. we repurpose an efficient inner-product functional encryption (IPFE) cryptosystem [23] for implementing a communication-efficient and privacy-preserving data aggregation scheme and using it in training a global predictor. The idea is that the edge nodes, at the customer-side, train local models using their net readings and the corresponding solar irradiance values, and then, they encrypt the models' parameters and send the ciphertexts to the utility server. Next, our scheme allows the utility server to use the ciphertexts of the local models' parameters to build the global model without learning the parameters of the customers' models. Finally, we leverage a change and transmit (CAT) approach to train the global model in a communication-efficient way. The idea of this approach is that instead of sending all the updated models' parameters in each round, only the parameters that have sufficiently been changed from previous round are transmitted. Our privacy-preserving data aggregation scheme reduces the size of the encrypted model's parameters by producing shorter chiphertexts while the CAT approach lowers the number of transmitted parameters.

To the best of our knowledge, *this is the first work that investigates energy forecasting in net-metering systems using FL.* Our main contributions in this paper are highlighted as follows.

- We prepared a dataset for net-metering systems to be used for training and evaluating our net-energy predictor by processing real power consumption and generation readings from the Ausgrid dataset [24] and solar irradiance records from SOLCAST website [25].
- We design a multi-data-source hybrid DL-based net-energy predictor to forecast future net readings. The proposed predictor not only uses net readings, but also exploits solar irradiance information to enhance the performance of the net-energy prediction.
- We repurpose an efficient IPFE cryptosystem [23] for implementing a secure data aggregation scheme to preserve customers' privacy during the FL training process of the global predictor by thwarting model inversion and membership inference attacks. Comparing to the existing aggregation schemes based on the Paillier cryptosystem, our aggregation scheme is more efficient in terms of the amount of exchanged data between the utility server and the customers' edge nodes during the FL process.
- To further reduce the communication overhead, we propose a CAT approach when transmitting the updated model's parameters.
- Extensive experiments are conducted and the results indicate that the proposed predictor is accurate in predicting the future net-energy while preserving the customers' privacy. Moreover, our privacy-preserving data aggregation scheme results in about 96% reduction in the communication overhead compared to the existing schemes based on the Paillier cryptosystem and the CAT approach leads to about 90% saving in the communication bandwidth

with a slight impact on the global model performance.

The rest of this paper is organized as follows. The existing works in the literature are discussed in Section II. Then, Section III presents the network/threat models and design goals. Some preliminaries are explained in Section IV. Our proposed approach is discussed in Section V. Section VI presents the evaluation results. Finally, conclusions are drawn in Section VII.

## II. Related Works

In this section, we overview the previous works on energy prediction in the smart grid based on two categories: non-FL-based approaches and FL-based approaches.

### A. Non-FL-Based Approaches

The energy forecasting methods proposed in the literature can be classified into statistical methods, e.g., ARIMA [7], and machine learning methods such as support vector machine (SVM) [8], decision tree (DT) [9], convolutional neural networks (CNNs) [10], and long short-term memory (LSTM) [5], [11], [12]. Among these methods, it has been found that DL-based methods outperform other methods because they can extract and learn the temporal and non-linear correlations within the energy readings, and hence provide more accurate energy predictions [11], [13].

Moreover, predicting the future required energy by each individual customer is challenging due to the dynamic and stochastic nature of the energy consumptions of individual customers [22], [26]. For the predictor to accurately estimate the future required energy of individual customers, it should be trained on a large-sized and diverse energy consumption history [6], [17]. Therefore, customer-specific predictors may not be accurate and effective because this causes an energy prediction latency and hampers the energy prediction for new customers. To overcome this challenge, several papers [14], [15], [22] have proposed generalized energy predictors trained on the historical data of many customers to increase the size of the dataset and ensure its diversity. However, in these works, the utility server collects the customers' data and uses it to train a prediction model. Revealing customers' fine-grained readings does cause a serious privacy concern because the data can be analyzed for sensitive information about the customers. Examples of this information include the appliances being used, when customers sleep and return home, or whether they are on travel, etc.

Furthermore, energy forecasting can be classified into consumed energy and net-energy forecasting [22]. Unlike consumed energy forecasting that only concerns about the prediction of future consumption readings, net-energy forecasting is concerned about predicting the difference between the consumed and generated energy (i.e., future net readings). Razavi *et. al.* [22] have compared the performance between the consumed energy and net-energy forecasting, and shown that the inclusion of the generated energy deteriorates the predictor's performance because it adds another source of uncertainty in addition to the energy consumption behavior. Although it has been shown [22] that net-energy forecasting is more challenging than consumed energy forecasting, preserving the customers' privacy is largely neglected in this paper. In addition, the correlation between net readings and solar irradiance has not been investigated in [22]. Unlike [22], in our paper, we have investigated this correlation and demonstrated that the solar irradiance can enhance the prediction performance of our net-energy predictor.

### B. FL-Based Approaches

As discussed in the previous subsection, a privacy issue arises if the utility server collects customers' fine-grained readings to train a global model. The schemes proposed in [6], [17] attempt to resolve this privacy issue by using an FL approach to train an energy forecasting model on the readings of several customers without revealing their readings. The idea is that the customers' edge nodes train local models using their energy consumption readings, and then, they send the parameters of their trained local models to the utility server to build the global prediction model using the local models' parameters without accessing the customers' readings. However, it has been found that the disclosure of the model parameters can still result in privacy leakages by launching attacks, such as membership inference [19] and model inversion [20]. Therefore, *to preserve the customers' privacy, both meter readings and models' parameters should not be disclosed by using FL and encryption, respectively*.

Although in FL, only the model parameters are sent instead of the training data, the global model needs multiple training rounds to reach convergence. Therefore, unlike the existing papers [6], [17] that use the basic FL approach when sending model parameters, in our paper, we use a CAT approach to achieve communication efficiency. In [6], [17], all the local model parameters are sent in every communication round; however, in our paper, only the model parameters that sufficiently change from the previous round are sent. Moreover, the existing works [6], [17] have investigated forecasting the future required energy only for consumption metering systems, and *none of the existing works have investigated the use of FL to forecast the future net-energy in net-metering systems*. Also, *none of the existing works including [6], [17] have designed a multi-data-source predictor to accurately forecast the future required energy*. Table I summarizes the differences between our paper and the existing works. It can be seen that our work is more privacy-preserving as it can achieve the confidentiality of both data and local models' parameters by using FL and encryption to hide the customers' data and models' parameters, respectively. Our work is also more communication-efficient than the existing works as it can reduce both the size of transmitted data and the number of transmitted model's parameters by using FL and CAT approach, respectively.

## III. Network/Threat Models and Design Goals

### A. Network/Threat Models

As shown in Fig. 1, the following three main entities are considered in our network model:

TABLE I: Comparison between our work and the existing works.

| Work | Net-metering system | Multi-data-source | Privacy preservation | | Communication efficiency | |
|---|---|---|---|---|---|---|
| | | | Data confidentiality | local model's parameters confidentiality | Messages' size reduction | Messages' number reduction |
| [5], [7]–[15], [26] | × | × | × | × | × | × |
| [22] | √ | × | × | × | × | × |
| [6], [17] | × | × | √ | × | √ | × |
| Ours | √ | √ | √ | √ | √ | √ |

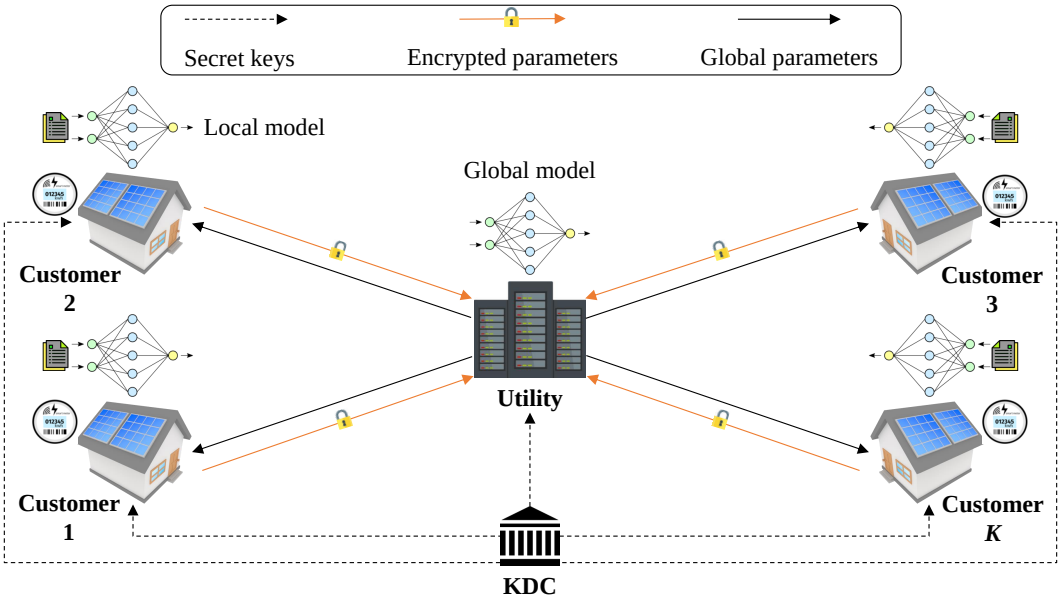



Fig. 1: Network model.

**Key Distribution Center (KDC).** The KDC bootstraps the system by generating the public parameters and secret keys for both the utility server and customers.

**Utility server.** The utility server needs to predict the future required energy by the customers for energy management. It adopts the FL approach to build a global machine learning model for energy prediction. It first initializes the parameters of the model and broadcasts it to the customers to train their local models on their local data. Then, the utility server aggregates the local models' parameters received from the customers to update the parameters of the global model and sends it back to the customers for another round of training. This process is repeated until the global model converges.

**Customers.** The customers have local renewable energy generators, such as rooftop solar panels, to generate energy. They also have net meters to periodically measure fine-grained readings representing the difference between their consumed and generated energy. These readings are used by customers to locally train machine learning models for net-energy forecasting. In each round of FL, customers train the local models based on their net readings, encrypt the updated local models' parameters, and send them to the utility server.

Our threat model considers both external and internal adversaries as follows:

- **External adversaries**. An adversary may eavesdrop on the communication channels between customers and the utility server, capture the transmitted data, and attempt to infer sensitive information about the customers.
- **Internal adversaries**. The utility server is honest-but-curious, i.e., it runs the FL approach honestly, but may be curious to learn sensitive information about the customers. The customers participating in FL are also honest-but-curious, i.e., they follow the protocol of FL training honestly, but may be curious to infer sensitive information about others. Moreover, some customers may collude with each other and/or with the utility server to infer sensitive information about customers.

Adversaries may launch membership inference and model inversion attacks to infer sensitive information about the customers if their local models' parameters are disclosed.

### *B. Design Goals*

*1) Privacy preservation:*

- **Data confidentiality:** The scheme should not leak information about the customers' readings, which are used to train their local models, to eavesdroppers, the utility server, and other customers.
- **Local model's parameters confidentiality:** The scheme should not also leak information about the customers'

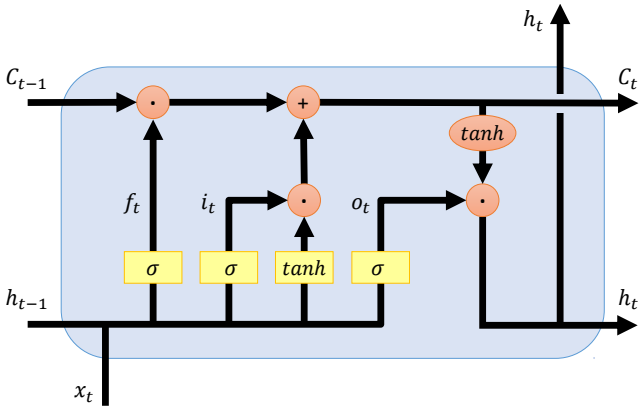



Fig. 2: The typical architecture of an LSTM cell.

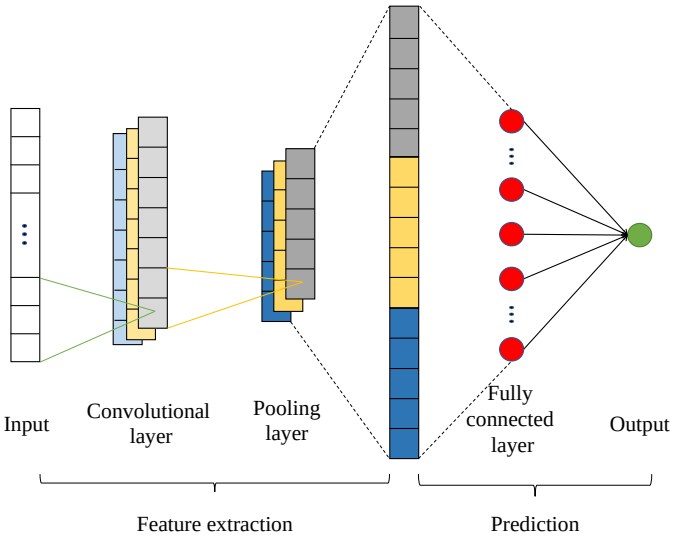



Fig. 3: The typical architecture of 1-D CNN.

local models' parameters to be secure against the membership inference and model inversion attacks.

*2) Efficiency:* The scheme should efficiently use the communication bandwidth by reducing the number and size of the exchanged messages between the utility server and customers during the FL training.

*3) Accuracy:* The net-energy forecasting model obtained from FL should accurately predict the future net-energy and achieve comparable performance to a corresponding model trained directly on the combined net readings of all customers.

## IV. Preliminaries

In this section, we briefly discuss the basics of some deep neural networks which will be used in our predictor. In addition, we briefly explain the IPFE scheme proposed by Kim *et. al.* [23].

### A. Long Short-Term Memory (LSTM) Networks

Net-meter readings are time-series data, which is an example of sequential data because the readings form an ordered sequence of values at different time steps. If we want to predict the next net reading in a sequence given the previous net readings, our prediction model should be capable of capturing the time correlations (time dependencies) between these readings. A recurrent neural network (RNN) is a type of neural networks suitable for handling sequential data because it has a sort of memory that can memorize the inputs and states from the past and use them to predict the output at the current time [27]. However, traditional RNNs suffer from the vanishing gradient problem that prevents them from learning the long-term dependencies between the data points in a sequence. A long short-term memory (LSTM) network is a variant of RNNs that can handle the vanishing gradient problem via an advanced cell structure with gates to regulate the flow of information through the cell [28].

The typical architecture of an LSTM cell is shown in Fig. 2. As we observe from the figure, the output of the LSTM cell at a time step $t$, namely $h_t$, is dependent not only on the current input, $x_t$, but also on the previous output of the cell, $h_{t-1}$. We can also see that each LSTM cell has three gates, namely, forget, input, and output gates that control the flow of information through the cell (i.e., selectively add/remove information to/from the internal cell state). First, the forget gate removes the irrelevant history by controlling which parts of the previous cell state, $C_{t-1}$, are kept and which parts are thrown away. Next, the input gate determines the important parts of the new information to be retained and the parts to be neglected. The outputs from the forget and the input gates are added together to update the cell state, $C_t$. Finally, the output gate determines the parts of the cell state to be outputted to the next LSTM cell.

The equations 1 to 5 [29] mathematically illustrate the operations of an LSTM cell, where $\sigma$ and Tanh are the Sigmoid and hyperbolic tangent activation functions, respectively, $f_t$, $i_t$, and $o_t$ are the Sigmoid activation outputs for the forget, input, and output gates, respectively, $W_f$, $W_i$, $W_o$, and $W_c$ are the weights for the input $x_t$, $U_f$, $U_i$, $U_o$, and $U_c$ are the weights of the previous output $h_{t-1}$, $b_f$, $b_i$, $b_o$, and $b_c$ are the biases, and $\odot$ is the Hadamard product, which is an element-wise multiplication.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{2}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{4}$$

$$h_t = o_t \odot \tanh\left(C_t\right) \tag{5}$$

### B. Convolutional Neural Networks (CNNs)

A convolutional Neural Network (CNN) is a type of neural networks with an automatic feature extraction capability [30]. In other words, we do not need to manually extract features

from the raw data before feeding them to the CNN but we can directly feed the raw data and let the CNN implement the feature extraction task. Although CNNs are very popular for processing two-dimensional data like images, there is a variant of CNNs, called one-dimensional (1-D) CNN, that can process 1-D data such as the univariate time series of net meter readings [30].

The typical architecture of 1-D CNN is shown in Fig. 3. As we can see from the figure, the 1-D CNN consists of convolutional, pooling, and fully connected layers. The convolutional layer consists of a group of parallel trained filters that extract different features from the input data through convolution operations. In particular, each filter is moved over the input data with a certain stride so that the filter's weights are convoluted with a corresponding patch of the input data [30]. The output of the convolutional layer is a group of feature maps corresponding to the group of filters. The feature maps are fed to the pooling layer, which is responsible for reducing the dimensionality of these maps through pooling operations. One common pooling operation is the maximum pooling, where for each patch from a feature map, the maximum value in the patch is chosen to represent the patch. The pooling layer produces distilled versions of the feature maps. The output of the pooling layer is flattened (i.e., transformed to 1-D output) and passed to a fully connected layer that learns the relations among the extracted features to compute the predicted value.

### C. Inner-Product Functional Encryption (IPFE)

Unlike other encryption schemes, a functional encryption (FE) scheme allows the holder of a decryption key to decrypt an encrypted data and only obtain a function of the data but not the data itself. For example, given the ciphertext of a message $m$ and a secret key for a function $f$, a decryptor only knows the value $f(m)$ but he cannot know $m$. Inner-product FE (IPFE) is a specific type of FE, where the function is the inner-product of two vectors, including an input vector $\mathbf{x}$ and another vector $\mathbf{y}$ associated with the function $f$. Thus, in an IPFE scheme, a decryptor only knows the value of the inner-product of $\mathbf{x}$ and $\mathbf{y}$ ($\langle \mathbf{x}, \mathbf{y} \rangle$) but nothing about $\mathbf{x}$. In [23], Kim *et. al.* have constructed an efficient IPFE scheme that consists of the following algorithms:

***Setup***$(1^\lambda) \rightarrow (\mathcal{PP}, \mathcal{MSK})$: This algorithm takes a security parameter $\lambda$ as an input and outputs public parameters $\mathcal{PP}$ and a master secret key $\mathcal{MSK}$. First, the algorithm selects three multiplicative groups $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ of prime order $q$ and an asymmetric bilinear pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ that takes two elements from $\mathbb{G}_1$ and $\mathbb{G}_2$ and maps them to an element in $\mathbb{G}_T$. Then, it chooses generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$. After that, it samples the matrix $\mathbf{B} \leftarrow \mathbb{GL}_n(\mathbb{Z}_q)$, where $\mathbb{GL}_n(\mathbb{Z}_q)$ is the general linear group of $(n \times n)$ matrices over $\mathbb{Z}_q$, and sets the matrix $\mathbf{B}^\star = \det(\mathbf{B}) \cdot \left(\mathbf{B}^{-1}\right)^\top$, where $\det(\mathbf{B})$ is the determinant of $\mathbf{B}$ and $\left(\mathbf{B}^{-1}\right)^\top$ is the transpose of $\mathbf{B}^{-1}$. Finally, the algorithm outputs $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$ as the public parameters and $(g_1, g_2, \mathbf{B}, \mathbf{B}^\star)$ as the master secret key.

***KeyGen***$(\mathcal{MSK}, \mathbf{x}) \rightarrow \mathrm{sk}$: This algorithm takes as input $\mathcal{MSK}$ and a vector $\mathbf{x} \in \mathbb{Z}_q^n$, and outputs the secret key $\mathrm{sk}$ as follows. First, it chooses a uniformly random element $\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_q$. Then, $\mathrm{sk}$ is calculated as $\mathrm{sk} = (K_1, K_2) = \left(g_1^{\alpha \cdot \det(\mathbf{B})}, g_1^{\alpha \cdot \mathbf{x} \cdot \mathbf{B}}\right)$, where the second component of $\mathrm{sk}$ is a vector of group elements.

***Encrypt***$(\mathcal{MSK}, \mathbf{y}) \rightarrow \mathrm{ct}$: This algorithm takes as input $\mathcal{MSK}$ and a vector $\mathbf{y} \in \mathbb{Z}_q^n$, and outputs the ciphertext $\mathrm{ct}$ as follows. First, it chooses a uniformly random element $\beta \xleftarrow{\mathrm{R}} \mathbb{Z}_q$. Then, $\mathrm{ct}$ is calculated as $\mathrm{ct} = (C_1, C_2) = \left(g_2^{\beta}, g_2^{\beta \cdot \mathbf{y} \cdot \mathbf{B}^*}\right)$, where the second component of $\mathrm{ct}$ is a vector of group elements.

***Decrypt***$(\mathcal{PP}, \mathrm{sk}, \mathrm{ct}) \rightarrow z$: This algorithm takes as input the public parameters $\mathcal{PP}$, a secret key $\mathrm{sk} = (K_1, K_2)$, and a ciphertext $\mathrm{ct} = (C_1, C_2)$, and outputs $z = \langle \mathbf{x}, \mathbf{y} \rangle$ as follows. First, it computes $D_1 = e(K_1, C_1)$ and $D_2 = e(K_2, C_2)$. Then, it checks whether there exists a $z$ such that $(D_1)^z = D_2$ by computing a discrete logarithm in $\mathbb{G}_T$. If there is such a $z$, it outputs $z = \langle \mathbf{x}, \mathbf{y} \rangle$. Otherwise, it outputs $\perp$, which implies that a valid $z$ cannot be found.

## V. Net-Energy Forecasting in Net-Metering System using Federated Learning (FL)

Our proposed approach in this paper enables the utility server to train a global model efficiently using FL to accurately forecast the future net-energy of a net-metering system while preserving the privacy of its customers.

### A. Overview

The proposed approach consists of the following few steps. First, the KDC initializes the approach by selecting the cryptographic public parameters, generating the secrets, and sending the public parameters along with the secrets to the utility server and the customers participating in the FL process, respectively. Second, the utility server initiates the FL process by agreeing with the customers on the DL architecture of the global net-energy forecasting model, initializing the model's parameters, and sending them to customers. Third, through several rounds, the utility server and customers use our data aggregation scheme to update the global model's parameters while hiding the parameters of the customers' local models to preserve privacy. In each round, upon receiving the model's parameters from the utility server, the customers train the model locally on their data, encrypt the updated model's parameters after training, and send them to the utility server. The utility server averages the aggregated models' parameters to update the global model's parameters without the need to know the parameters of the individual (local) models of the customers to preserve privacy.

### B. Net-Energy Forecasting Model

In this subsection, we propose a hybrid DL-based net-energy forecasting model for a net-metering system. First, we prepare a dataset for the net-metering system and analyze it. Then, based on the analysis, we design the DL architecture of our model.

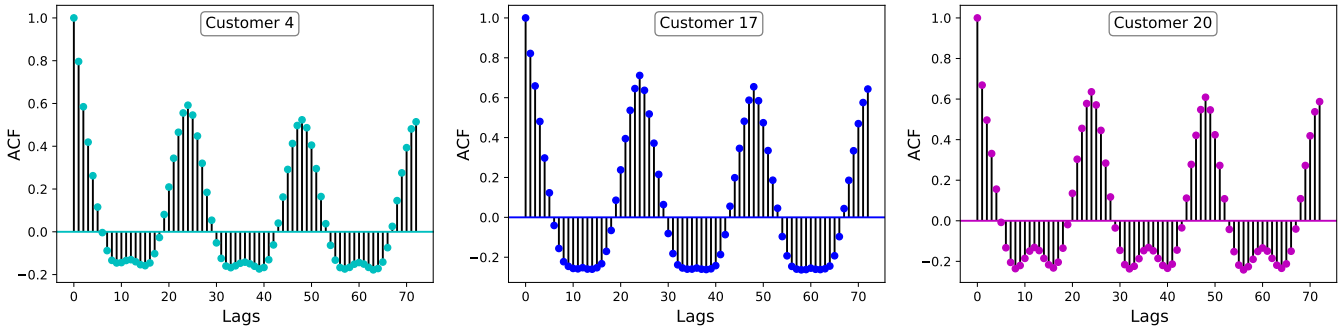



Fig. 4: The ACF of the net readings of three randomly selected customers.

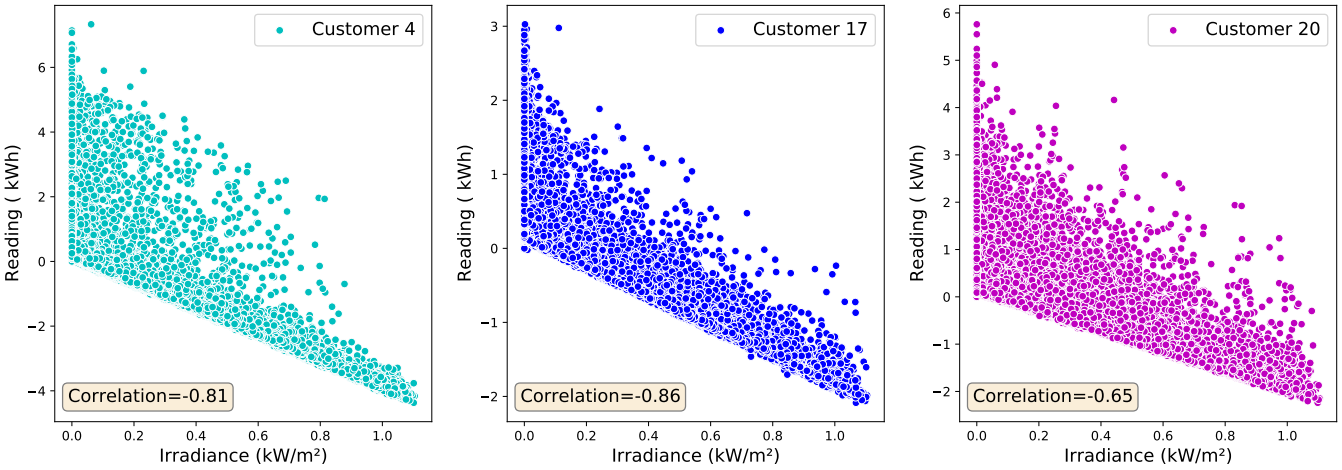



Fig. 5: The correlation between the net readings and the corresponding solar irradiance values for three randomly selected customers.

*1) Dataset Preparation:* We exploit the publicly available dataset of real power consumption and generation readings released by Ausgrid, the largest electricity distributor in the east coast of Australia [24], to prepare our dataset. The customers included in Ausgrid dataset had homes with rooftop solar panels and SMs to record their power consumption and generation readings on half-hour granularity. The readings in the dataset were recorded from 2010 to 2013. Besides the readings, the Ausgrid dataset contains the location of each customer. Having the Ausgrid dataset, we have conducted the following preprocessing steps. First, we have removed the outliers to create a clean dataset using the same approach in [31]. Second, we have converted the individual consumption and generation readings to net readings. The range of the net readings in our dataset is from -11.33 to 11.22 kWh. Third, we have used SOLCAST [25], a website that provides historical weather information for any location in the world at any specified time, to obtain the solar irradiance values corresponding to the locations of the customers and the times of their readings.

*2) Dataset Analysis:* The correlation is a statistical means used to measure the relationship between two time-series data in a quantitative manner. The values of correlation are in the range of $[-1, 1]$. If the correlation value is close to 1 or $-1$, this indicates that the two time-series data have a perfect positive or negative correlation, respectively. On the other hand, if the correlation value is close to zero, this indicates that the two time-series data are uncorrelated. A special type of correlation, called the autocorrelation function (ACF), is used to measure the relationship (correlation) between a time-series data and a lagged version of this time-series with different time lags. Consequently, the ACF can be used to characterize how the time-series data values are related to each other. In other words, the ACF can measure how the current value in a time-series data is dependent on the previous values in the time-series data.

Fig. 4 shows the ACF of the net readings of three randomly selected customers from the dataset with different time lags. As we can see from the figure, there are correlations between the current net reading and the previous net readings for all customers. Moreover, the ACF, for any customer, shows periodicity every 24 hours. Therefore, in this paper, we use the last 24 net readings of a customer to predict the next net reading of the customer. Fig. 5 shows the correlation between the time-series net readings and the corresponding time-series solar irradiance for the same customers given in Fig. 4. As we can see from the figure, the net reading decreases with the increase of the solar irradiance, which indicates a negative correlation between them. This is because when the solar irradiance increases, the amount of generated power increases, which in turn decreases the net reading. In addition, the correlation values are given in Fig. 5. The sign of the

values indicates the direction of the correlation, while the magnitude indicates the significance of the correlation. It can be concluded that the net readings are significantly correlated with the solar irradiance values. Therefore, the solar irradiance can enhance the net reading forecasting.

*3) Proposed Model Architecture:* The DL architecture in our multi-data-source net-energy forecasting model is shown in Fig. 6. Given the time-series nature of the net readings, we use an LSTM module as the main building block in our net-energy predictor because it can capture the correlation between the time-series data. However, to enhance the accuracy of our predictor, the net readings are first fed to a CNN module to extract the important features from the readings before feeding them to the LSTM module. Moreover, the analysis of our dataset has revealed the existence of correlation between the net readings and the corresponding solar irradiance values. Thus, for better prediction accuracy, we develop a multi-data-source model that takes both the net readings and solar irradiance values as inputs as shown in Fig. 6. The net readings serve as the main input to perform the net-energy forecasting and the solar irradiance serves as an auxiliary input to further enhance the forecasting capability of the model by learning the dependencies between the net readings and solar irradiance values.

To predict a future net reading (e.g., $r_{t+24}$), the previous 24 net readings $(r_t, r_{t+1}, \ldots, r_{t+23})$ and the corresponding 24 solar irradiance values $(i_t, i_{t+1}, \ldots, i_{t+23})$ are inputted to our model. Internally, the 24 net readings sequence is divided into multiple groups of non-overlapped readings and fed to a CNN module, where the size of each group is a hyper-parameter. Each CNN is responsible for extracting the important features from a corresponding group. Similarly, the 24 solar irradiance sequence is divided into multiple groups and fed to a CNN module to extract the important features from these groups. The extracted features from each net readings group and the corresponding solar irradiance group are concatenated together and fed as a single time step features to an LSTM unit of the LSTM module. The LSTM module correlates the features of the different time steps and outputs the predicted net reading.

### C. Privacy-Preserving Data Aggregation Scheme

In this subsection, we repurpose the IPFE scheme proposed by Kim *et. al.* [23] for implementing an efficient privacy-preserving data aggregation. The IPFE scheme [23] is designed so that the master secret key is used for encryption. However, in our scenario, we have multiple users, and it is not secure to give the same secret key for multiple users. Therefore, we have split the master secret key used for encryption into multiple unique keys and we give a different secret key for each user. For encryption, each user uses his own secret key and an encryption algorithm similar to that of [23] to produce a ciphertext. For decryption, after aggregating the ciphertexts of the different users, we obtain a ciphertext that can be decrypted using a decryption algorithm similar to that of [23]. Our data aggregation scheme consists of three main phases, including *initialization*, *encryption*, and *aggregation and decryption*.

*1) Initialization [23]:* The KDC runs the following algorithms to generate the public parameters and secrets for the utility server and customers, respectively.

***Setup***$(1^\lambda) \rightarrow (\mathcal{PP}, \mathcal{MSK})$: This algorithm takes a security parameter $\lambda$ as an input and outputs public parameters $\mathcal{PP}$ and a master secret key $\mathcal{MSK}$. First, the algorithm selects two cyclic additive groups $\{\mathbb{G}_1, \mathbb{G}_2\}$ and a multiplicative group $\mathbb{G}_T$ of prime order $q$ and a bilinear pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Then, the algorithm selects two generators $P$ and $Q$ of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, where $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$. Note that the bilinear paring satisfies the following property: $e(aP, bQ) = e(P, Q)^{ab}$, where $a, b \in Z_q$. Further, the algorithm computes three invertible $(n \times n)$ matrices over $Z_q$, namely, $\{\boldsymbol{B}, \boldsymbol{N}_1, \boldsymbol{N}_2\}$, where $Z_q$ is a finite field of prime order $q$. Finally, the algorithm outputs $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, P, Q)$ as the public parameters and $(B, N_1, N_2)$ as the master secret key. Note that the public parameters are shared with the utility server and customers while the master secret key is never shared with any entity, i.e., it is kept secret at the KDC.

***KeyGenCustomer***$(\mathcal{MSK}, C_i) \rightarrow (\Delta_i, \mathcal{SK}_{C_i})$: This algorithm takes $\mathcal{MSK}$ as an input and for each customer $C_i$, it outputs $\Delta_i$, which is an $n$-dimensional vector over $Z_q$, used to mask the local model parameters, and a secret key $\mathcal{SK}_{C_i}$ used to encrypt them. Each customer $C_i$ is given a different $\Delta_i$ from the KDC for each round of the FL training. $\mathcal{SK}_{C_i}$ is computed as $\mathcal{SK}_{C_i} = (\boldsymbol{N}_1^{-1}\boldsymbol{B}_i', \boldsymbol{N}_2^{-1}\boldsymbol{B}_i'')$, where $\boldsymbol{B}_i'$ and $\boldsymbol{B}_i''$ are two $(n \times n)$ matrices selected randomly such that $\boldsymbol{B}_i' + \boldsymbol{B}_i'' = \boldsymbol{B}^{-1}$.

To enable the KDC to distribute the secret keys for each customer, we follow the public-key cryptography approach. In particular, each entity, including the KDC, utility server, and customers, has a unique public/private key pair. After generating $(\Delta_i, \mathcal{SK}_{C_i})$ for a customer $C_i$, the KDC encrypts them using $C_i$'s public key and sends them to $C_i$. Upon receiving the encrypted secrets, $C_i$ uses his/her own private key to decrypt them.

***KeyGenUtility***$(\mathcal{MSK}) \rightarrow (\mathcal{SK}_U, \Delta)$: This algorithm takes $\mathcal{MSK}$ as an input and outputs the utility server secret key $\mathcal{SK}_U$ used to aggregate the received encrypted models' parameters and $\Delta$, which is an $n$-dimensional vector over $Z_q$, used to unmask the aggregated parameters. $\mathcal{SK}_U$ is computed as $\mathcal{SK}_U = (\boldsymbol{BN}_1, \boldsymbol{BN}_2)$ and $\Delta$ is computed as $\Delta = \sum_{i=1}^{K} \Delta_i$, where $K$ is the number of customers participating in the FL. In each round, the utility server informs the KDC with the customers who have sent their models' parameters and the KDC responds with $\Delta$ corresponding to those customers.

For the KDC to distribute the secrets $(\mathcal{SK}_U, \Delta)$ to the utility server, it encrypts them using the utility server's public key and sends them to the utility server. Upon receiving the encrypted secrets, the utility server uses its own private key to decrypt them.

*2) Encryption:* In this phase, after each customer trains the model locally on his/her private data, he/she runs the following algorithm to mask and encrypt the updated model parameters.

***Encrypt***$(\Delta_i, \mathcal{SK}_{C_i}, X_i) \rightarrow CT_i$: This algorithm takes as input the two secrets, $\Delta_i$ and $\mathcal{SK}_{C_i}$, of customer $C_i$ and $X_i$,

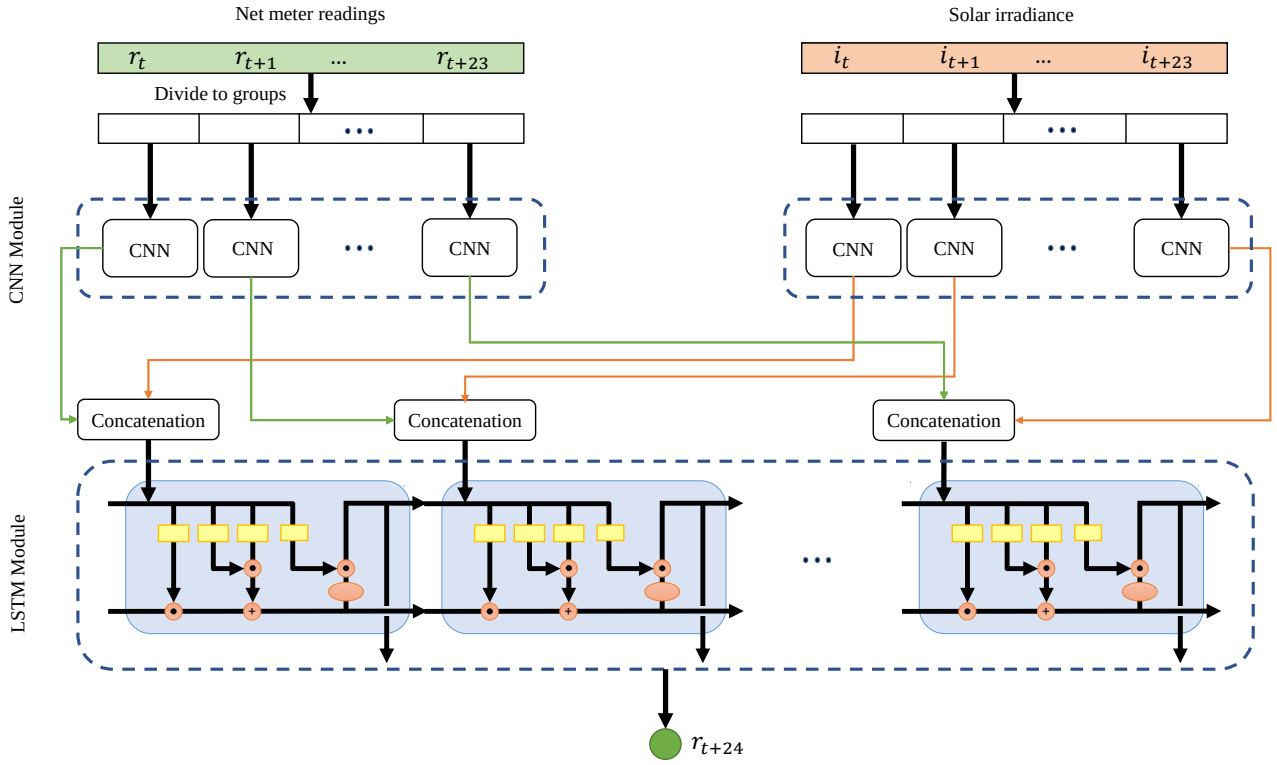



Fig. 6: The proposed net-energy forecasting model.

which is an $n$-dimensional vector representing the customer's local model parameters. The algorithm outputs the ciphertext $CT_i$ corresponding to $X_i$ as follow. First, $X_i$ is masked by computing $\hat{X}_i = X_i + \Delta_i$. Then, the ciphertext $CT_i$ is computed as follows:

$$CT_i = (\hat{X}_i \boldsymbol{N}_1^{-1} \boldsymbol{B}_i' Q,\ \hat{X}_i \boldsymbol{N}_2^{-1} \boldsymbol{B}_i'' Q) \tag{6}$$

*3) Aggregation and decryption:* Upon receiving the ciphertexts from all the customers participating in the FL, the utility server runs the following algorithm to aggregate the local models' parameters of the customers.

***Decrypt***$(\mathcal{SK}_U, \Delta, \{CT_1, CT_2, \ldots, CT_K\}) \rightarrow X$: This algorithm takes as input the two secrets, $\mathcal{SK}_U$ and $\Delta$, of the utility server and $\{CT_1, CT_2, \ldots, CT_K\}$, which is the set of ciphertexts corresponding to the local models' parameters of the participating customers. This algorithm outputs $X = \sum_{i=1}^{K} X_i$, which is an $n$-dimensional vector representing the sum of the individual local models' parameters of the participating customers. $X$ is computed as follows.

$$\begin{aligned} CT &= \sum_{i=1}^{K} CT_i = (CT(1), CT(2)) \\ &= (\sum_{i=1}^{K} \hat{X}_i \boldsymbol{N}_1^{-1} \boldsymbol{B}_i' Q,\ \sum_{i=1}^{K} \hat{X}_i \boldsymbol{N}_2^{-1} \boldsymbol{B}_i'' Q) \end{aligned} \tag{7}$$

$$e(\boldsymbol{B}\boldsymbol{N}_1 P, CT(1)) = e(P,Q)^{(\sum_{i=1}^{K} \hat{X}_i \boldsymbol{B}_i')\boldsymbol{B}} \tag{8}$$

$$e(\boldsymbol{B}\boldsymbol{N}_2 P, CT(2)) = e(P,Q)^{(\sum_{i=1}^{K} \hat{X}_i \boldsymbol{B}_i'')\boldsymbol{B}} \tag{9}$$

$$\begin{aligned} (8) * (9) &= e(P,Q)^{(\sum_{i=1}^{K} \hat{X}_i (\boldsymbol{B}_i' + \boldsymbol{B}_i''))\boldsymbol{B}} = e(P,Q)^{\sum_{i=1}^{K} \hat{X}_i} \\ &= e(P,Q)^{\Delta + \sum_{i=1}^{K} X_i} \end{aligned} \tag{10}$$

$$e(\Delta P, Q) = e(P,Q)^{\Delta} \tag{11}$$

$$(10)/(11) = e(P,Q)^{\sum_{i=1}^{K} X_i} = e(P,Q)^{X} \tag{12}$$

Finally, $X$ can be computed from $e(P,Q)^X$ by computing the discrete logarithm in $\mathbb{G}_T$, which can be efficiently computed using the baby-step giant-step algorithm [32] given that the elements of $X$ are small-sized numbers.

### D. Change and Transmit (CAT) Approach

In order to efficiently use the available communication bandwidth, it is desirable to reduce the size and the number of transmitted messages from the customers to the utility server. Our data aggregation scheme leads to short-size encrypted model's parameters. However, to reduce the number of transmitted model's parameters, we use a CAT approach instead of the basic FL approach. In the basic FL approach, each customer transmits all the updated model's parameters in each round of the FL process. However, it is not necessary to update the model's parameters that do not sufficiently change from previous round. Therefore, we use the CAT approach when transmitting the updated model's parameters. In each round, the updated model's parameters are compared to the last transmitted model's parameters. Only the parameters whose

absolute percentage change exceeds a predefined threshold are transmitted to the utility server, i.e., when:

$$\left|\frac{x_i - {x'}_i}{{x'}_i}\right| * 100\% \geq \text{Threshold},$$

where $x_i$ is the updated value of the $i$-th parameter after the local training and ${x'}_i$ is the last transmitted value of the same parameter. For the other model's parameters that are not transmitted, the utility server uses the latest received versions of these parameters. This indicates that the CAT approach leads to communication efficiency. The higher the threshold, the higher the communication efficiency that can be achieved. However, higher thresholds introduce larger errors in the transmitted parameters, which may affect the global model prediction performance. Therefore, in Section VI we will investigate the selection of a threshold value that can produce high communication efficiency without significantly deteriorating the global model performance.

## VI. Evaluations

In this section, we first analyze the security and privacy preservation of our proposed scheme. Then, we evaluate its overhead. After then, we assess the performance of our net-energy forecasting model. Finally, we study the communication saving that can be achieved by using the CAT approach and the impact on the prediction performance.

### A. Security and Privacy Analysis

*1) Security of the Aggregation Scheme:* The security of the aggregation scheme is critical to our FL-based net-energy forecasting approach because it allows the utility server to build the global model without the need for knowing the local models' parameters of the individual customers. In building our aggregation scheme, we have repurposed the IPFE scheme [23] to perform privacy-preserving data aggregation. Specifically, instead of using the master secret key $\mathcal{MSK}$ by all customers to encrypt their models' parameters, a different secret key derived from $\mathcal{MSK}$ is used by each customer while the $\mathcal{MSK}$ is only known by the KDC. This can secure our FL approach because if the encrypted model's parameters of a customer are intercepted by a curious customer, he/she is not able to decrypt them because the two customers use different secret keys. However, the core methods used for encryption and decryption are similar to those of the IPFE scheme [23]. Consequently, our privacy-preserving data aggregation scheme has the same security logic as the IPFE scheme, which is proved to be secure in [23]. For detailed security analysis and formal security proof in the generic bilinear group model, see Appendix A.

*2) Privacy of Participating Customers:* The proposed approach follows the Privacy-by-Design principle to preserve the privacy of the participating customers in the FL process against external adversaries, curious customers, and curious utility server. Instead of sending the local model's parameters in clear to the utility server, each customer encrypts them using a unique secret key. This thwarts the membership inference and model inversion attacks [19], [20], which have been demonstrated to reveal private information about the customers whose data have been used for training if the model's parameters are known. Moreover, each customer $C_i$ masks his/her local model's parameters with a random vector $\Delta_i$ before encrypting it, which is important to prevent unlinkability of the ciphertexts. Specifically, this ensures that the ciphertexts of the same parameter values look different if they are sent in different FL rounds.

Furthermore, the random vector $\Delta_i$ plays an important role in preventing the curious utility server from decrypting a single customer' local model parameters ($CT_i$). If the utility server passes $CT_i$ to the ***Decrypt***() algorithm instead of $\{CT_1, CT_2, \ldots, CT_K\}$, it obtains $e(P,Q)^{\Delta_i + X_i}$ at step (10) of the scheme. For the utility server to obtain $\Delta_i + X_i$, it has to compute the discrete logarithm, which is computationally intractable in this case because $\Delta_i$ is an $n$-dimensional vector over $Z_q$, i.e., the elements of $\Delta_i$ are large-size numbers. Finally, our FL approach thwarts collusion attacks between the utility server and customers. The only way for the utility server to obtain the model parameters of a single customer $C_i$ is to collude with all the other customers, which is not practical given the large number of customers.

### B. Overhead

To evaluate the communication overhead in our approach, we have used 224-bit elliptic curve (secp224k1) as recommended by the National Institute of Standards and Technology (NIST). The communication overhead is measured in terms of the number and size of messages exchanged between the utility server and the participating customers in the FL. In each round, each customer sends an encrypted vector representing the updated local model's parameters. Using our aggregation scheme, the length of this vector is $2n$ group elements, where $n$ is the number of the model's parameters. Each group element is a point on the elliptic curve, which can be represented by $56$ bytes. Therefore, the communication overhead needed to send the encrypted model's parameters is $112n$ bytes. Furthermore, the elliptic curve points can be compressed, i.e., represented by a smaller number of bits [33]. Thus, according to [33], the communication overhead in our aggregation scheme is approximately equal to $57n$ bytes.

On the other hand, some works in the literature [34] proposed secure data aggregation using a threshold homomorphic encryption (THE) based on the Paillier cryptosystem [21]. Using the Paillier cryptosystem, the size of each encrypted model parameter is $512$ bytes. Therefore, the communication overhead needed to send the encrypted model's parameters is $512n$ bytes. Moreover, using THE, the utility server is not able to directly decrypt the aggregated vectors from the participating customers because it does not know the decryption key that is divided into shares and distributed among the customers. Therefore, the utility server sends the aggregated vectors back to the customers to partially decrypt and send the decryption shares. Upon receiving a threshold of decryption shares, the utility server is able to decrypt the aggregated vectors. This means that using THE requires three communication rounds in each round of the FL process. Thus, the communication

overhead in the aggregation scheme based on the Paillier cryptosystem is equal to $1536n$ bytes. This indicates that *our aggregation scheme achieves about* $96\%$ *reduction in the communication overhead*.

To evaluate the computation overhead in our approach, we have implemented it using the Python Charm library [35]. The computation overhead is measured in terms of the time required to compose the messages exchanged between the utility server and the participating customers in the FL. In each round, each customer computes an encrypted vector of $n$ elements. From our experiments, computing an encrypted element using our approach takes less than $1.9$ msec, while it takes more than $26.5$ msec using the Paillier cryptosystem. This is because the encryption in our approach is dominated by scalar multiplication operations, while it is dominated by modular exponentiation operations in the Paillier cryptosystem. Therefore, *using our approach to encrypt the updated model parameters lowers the computation overhead on customers compared to the Paillier cryptosystem*.

For the utility, in each round of the FL, it aggregates $K$ encrypted vectors from $K$ participating customers to update the global model parameters. In our approach, the aggregation is dominated by pairing operations, while it is dominated by modular exponentiation operations in THE based on the Paillier cryptosystem. Although pairing is harder to compute than modular exponentiation given the same security parameter, our experiments' results show that the computation overhead of a modular exponentiation operation in the Paillier cryptosystem is more than the computation overhead of 2 pairing operations in our approach. This is because our approach achieves the same security level as the Paillier cryptosystem using smaller security parameter. Besides, in THE, customers are involved in the aggregation process. In particular, each customer computes a decryption share in each FL round, which requires computing $n$ modular exponentiation operations. In contrast, in our approach, the aggregation process is solely executed by the utility server. It is not a problem to increase the computation overhead on utility server given its high computational capabilities. However, it is problematic to increase the computation overhead on customers. Finally, using THE introduces a delay in the aggregation process because it requires three communication rounds between the utility server and customers in each round of the FL.

### C. Performance of Net-energy Forecasting Model

In this subsection, we discuss the results obtained from the three experiments we conducted to assess the performance for our net-energy forecasting model. The first experiment assesses how good is the DL architecture of our model. The second one assesses how accurate the model after the FL process is. The last one evaluates the impact of the CAT approach on both the communication overhead and the model performance.

*1) Experimental Setup and Metrics:* To evaluate the accuracy of our net-energy forecasting approach, we have implemented it using the TensorFlow Federated framework running by the Tennessee Technological University's high-performance cluster with one NVIDIA Tesla K80 GPU. We have used five metrics to evaluate the performance in terms of prediction error, including mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), normalized mean absolute error (nMAE), and mean arctangent absolute percentage error (MAAPE).

- **MSE** is computed as follows:
$$\text{MSE} = \frac{\sum_{i=1}^{\mathcal{N}} (r_i - \hat{r}_i)^2}{\mathcal{N}},$$
where $\mathcal{N}$ is the number of test samples and $r_i$ and $\hat{r}_i$ are the actual and predicted net readings in kWh, respectively.
- **RMSE** is computed as follows:
$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{\mathcal{N}} (r_i - \hat{r}_i)^2}{\mathcal{N}}}.$$
- **MAE** is computed as follows:
$$\text{MAE} = \frac{\sum_{i=1}^{\mathcal{N}} |r_i - \hat{r}_i|}{\mathcal{N}}.$$
- **nMAE** is computed as follows:
$$\text{nMAE} = \frac{\sum_{i=1}^{\mathcal{N}} |r_i - \hat{r}_i|}{\sum_{i=1}^{\mathcal{N}} |r_i|} * 100\%.$$
- **MAAPE** is computed as follows:
$$\text{MAAPE} = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \arctan\left(\left|\frac{r_i - \hat{r}_i}{r_i}\right|\right) * 100\%.$$

RMSE and MAE measure the prediction error in kWh, while nMAE and MAPPE are scale-free metrics that measure the prediction error as a percentage. The lower MSE, RMSE, MAE, nMAE, and MAAPE, the better the predictor. It is noteworthy to mention that the popular mean absolute percentage error (MAPE) metric cannot be used to evaluate the prediction performance in the net-metric system. This is because the actual net readings can be zero, in which case, MAPE tends to approach the infinity regardless of the prediction error. Therefore, in this paper we use MAPPE, which is a modified version of MAPE proposed in [36], to overcome the limitation of MAPE.

*2) Dataset Preprocessing:* We have selected a group of 31 customers from the dataset prepared in Section V-B1 to form the dataset used to train and evaluate our net-energy forecasting model. Then, we have reduced the granularity of the dataset from half-hour to one-hour to make the training process efficient. After that, we have transformed the net readings of each customer into sliding windows with look-back of 24 readings size and look-ahead of one reading size, i.e., given 24 net readings, the net-energy forecasting model predicts the 25th reading. Similarly, we have transformed the solar irradiance values into sliding windows of size 24. Consequently, each sample of the dataset consists of 48 values (24 net readings and the corresponding 24 solar irradiance values) and has a label representing the future net reading after the 24 net readings in the sample. Further, our dataset has been divided into training and test subsets with the ratios of 70% and 30%, respectively. Finally, these subsets have been

TABLE II: The optimal hyper-parameters of our predictor.

| Layers | Number of units | Activation Function |
|---|---|---|
| Input$_r$ | 24 | Linear |
| Conv1D$_r$ | 64 | ReLU |
| Input$_i$ | 24 | Linear |
| Conv1D$_i$ | 16 | ReLU |
| LSTM | 256 | Sigmoid |
| Output | 1 | Linear |

Note: $r$ and $i$ denote net-reading and solar irradiance, respectively.

TABLE III: Comparison between the prediction performance of our predictor and other DL-based predictors.

| Predictor | Metrics | | | | |
|---|---|---|---|---|---|
| | MSE | RMSE | MAE | nMAE | MAAPE |
| CNN | 0.39 | 0.62 | 0.37 | 3.25 | 3.18 |
| LSTM [22] | 0.37 | 0.61 | 0.34 | 2.97 | 2.91 |
| CNN-LSTM | 0.36 | 0.60 | 0.33 | 2.89 | 2.83 |
| Ours | 0.32 | 0.57 | 0.32 | 2.80 | 2.72 |

normalized so that the values of all variables are in the same range. It is important to prevent the large-value variables from unnecessarily dominating the model.

*3) Experiments and Discussions:* Here we explain the conducted experiments and discuss the results.

**Experiment 1.** In this experiment, we evaluate the accuracy of the DL architecture of our predictor in net-energy forecasting. For this purpose, we have trained three DL-based predictors, with CNN, LSTM, and CNN-LSTM architectures, to compare with ours. It is noteworthy that our predictor is a multi-data-source predictor that considers both the previous net readings and the corresponding solar irradiance values to make a prediction, while the CNN-LSTM predictor that we compare to is a single-data-source predictor that only considers the previous net readings to make a prediction. For fair comparison, we have used the Hyperopt optimization tool [37] to optimize the hyper-parameters of the four predictors. The optimal hyper-parameters of our predictor in terms of the layers, the number of units in each layer, and the activation function used in each layer are given in Table II. The mean square error and Adam were the chosen loss function and optimizer, respectively.

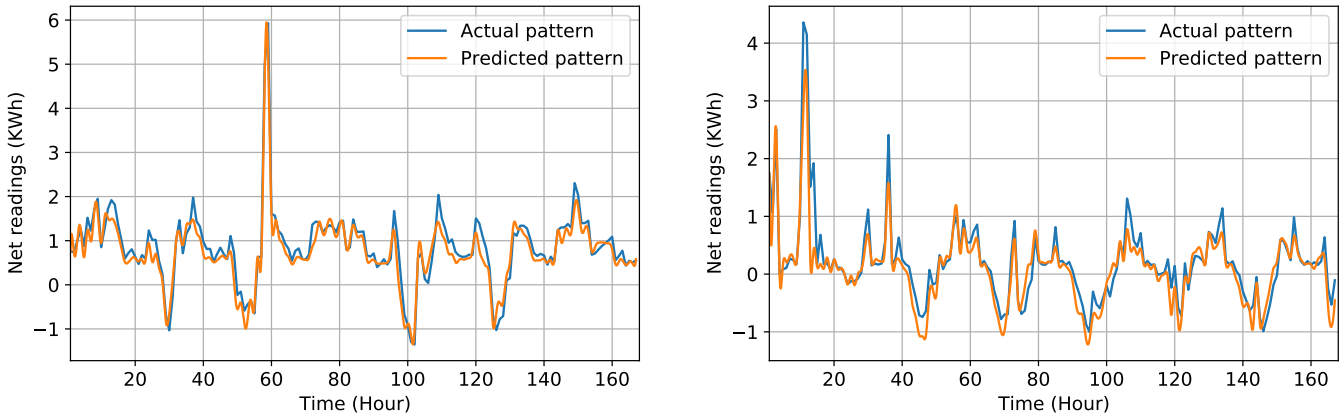
Table III compares the performance of our predictor to the other DL-based predictors in term of MSE, RMSE, MAE, nMAE, and MAAPE. It is clear from Table III that the predictor based on hybrid CNN-LSTM architecture gives better performance than either the CNN-based or the LSTM-based [22] predictors. This is because it can combine the advantages of both CNN and LSTM. The CNN module extracts the important features from the net readings and the LSTM module correlates the extracted features at different time steps to make the prediction. Moreover, it can be noted that our multi-data-source predictor surpasses a single-data-source predictor that has the same DL architecture. This is because our predictor learns the correlation between the net readings and the solar irradiance, which results in an improved prediction.

TABLE IV: Comparison between the prediction performance with and without FL.

| Predictor | Metrics | | | | |
|---|---|---|---|---|---|
| | MSE | RMSE | MAE | nMAE | MAAPE |
| Without FL (Baseline) | 0.33 | 0.58 | 0.32 | 2.80 | 2.76 |
| With FL (Participants) | 0.45 | 0.67 | 0.38 | 3.34 | 3.37 |
| With FL (Non-participants) | 0.50 | 0.70 | 0.39 | 3.37 | 3.45 |

**Experiment 2.** In this experiment, we evaluate our FL-based predictor. For this purpose, the 31 customers of the dataset have been divided into two groups. The first group consists of 25 customers who participate in the FL training. The second group consists of 6 customers whose data are used for evaluating the generalization of the global model, i.e., evaluating the model performance for the non-participating customers. In this experiment, we have trained two predictors; one with FL and the other without FL. The predictor trained without FL is a theoretical global model obtained if all the customers' data are available to the utility server. It is just used here as a baseline (i.e., a reference to assess how good is the prediction of our FL-based predictor). To train a predictor using FL, the utility server randomly initializes the global model's parameters and sends it to the 25 customers to train it locally using their own data. The customers send the updated models to the server that aggregates them and returns a global model to the customers to train again on their local data. This process continues until the convergence of the global model is reached.

Table IV compares the performance of the two predictors in terms of MSE, RMSE, MAE, nMAE, and MAAPE. It is evident from Table IV that the FL-based predictor achieves a slightly lower performance than the baseline. *This demonstrates that our FL-based predictor can accurately predict the customers' net-energy while preserving their privacy*. Also, in Table IV, we compare the performance of our predictor for participating and non-participating customers in the FL process. It is evident from the table that the predictor's performance for non-participating customers is very close to its performance for the participating customers. This indicates that our predictor has good generalization. Furthermore, we have randomly selected a participating customer and a non-participating customer from the dataset to visualize the prediction accuracy of our predictor. Figs. 7a and 7b compare between the actual net readings and the predicted readings by our predictor for a week for a participating customer and a non-participating customer, respectively. Fig. 7a does not only show that the predicted readings are very close to the actual readings, but also demonstrates the ability of our predictor to accurately follow the actual pattern of the participating customer. Fig. 7b demonstrates the same fact for the non-participating customer, which indicates that our predictor has




(a) A participating customer.

(b) A non-participating customer.

Fig. 7: Comparison between the actual and predicated net readings for a week.

good generalization.

**Experiment 3.** In this experiment, we study the impact of the CAT approach used by our work on both the communication overhead and the model performance compared to the basic FL approach used by [6], [17]. To provide a fair comparison, we encrypt the local models' parameters in our work, [17] and [6]. The global model's parameters have been randomly initialized and it has been trained for 40 rounds using the basic FL approach. After each round, the global model performance has been evaluated using all the customers of the dataset. Then, the global model obtained after the first round of training has been used as an initial global model in the CAT approach and trained for additional 39 rounds. We have experimented the CAT approach with different thresholds. At each threshold, we have evaluated the prediction performance of the resultant global model and measured the achieved communication saving.

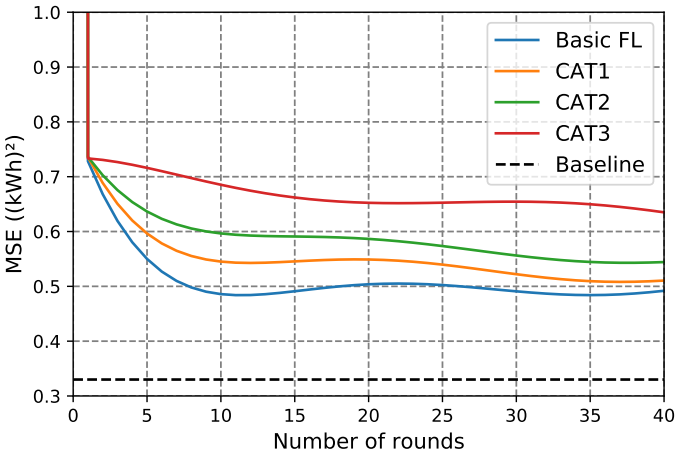
In Fig. 8 and Table V, we present the results of 3 cases; CAT1 (with 2% threshold), CAT2 (with 4% threshold), and CAT3 (with 10% threshold). Figs. 8a-8e compare the performance of the predictor obtained by using the basic FL approach and predictors obtained by using the CAT approach in terms of MSE, RMSE, MAE, nMAE, and MAAPE, respectively after each training round. On the other hand, Fig. 8f shows the communication saving that can be achieved in each round using the CAT approach. We can observe that the performance of the CAT1-based predictor is slightly lower than the performance of the basic FL-based predictor. However, the CAT1-based predictor provides more than 80% saving in the communication bandwidth compared to the predictor trained using the basic FL approach. Moreover, we can observe from Fig. 8 that the higher the threshold, the higher the communication saving and the lower the global model performance. This trend continues until the threshold reaches 10%. At 10% threshold, we observe that the performance of the model obtained after 39 rounds is slightly higher than the performance of the initial model. Using thresholds higher than 10% results in a global model that is not as good as the initial model after 39 rounds of training. To conclude, using the CAT approach can produce a predictor with acceptable performance and about 90% saving in the communication bandwidth. Finally, Table V summarizes the comparison between the basic FL and CAT approach.

TABLE V: Comparison between basic FL and CAT approach.

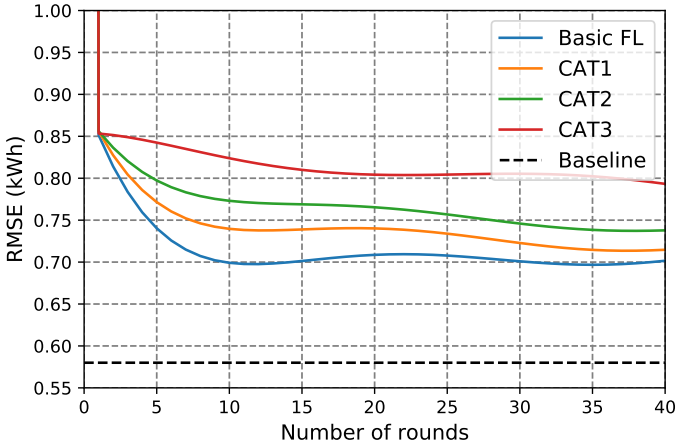
| Approach | Metrics | | | | | Average Saving/round |
|---|---|---|---|---|---|---|
| | MSE | RMSE | MAE | nMAE | MAAPE | |
| Basic FL | 0.47 | 0.69 | 0.38 | 3.35 | 3.40 | 0 |
| CAT1 | 0.51 | 0.72 | 0.42 | 3.65 | 3.70 | 81.5 |
| CAT2 | 0.54 | 0.74 | 0.45 | 3.85 | 3.90 | 84.5 |
| CAT3 | 0.65 | 0.80 | 0.57 | 4.85 | 4.90 | 91.2 |

## VII. Conclusion

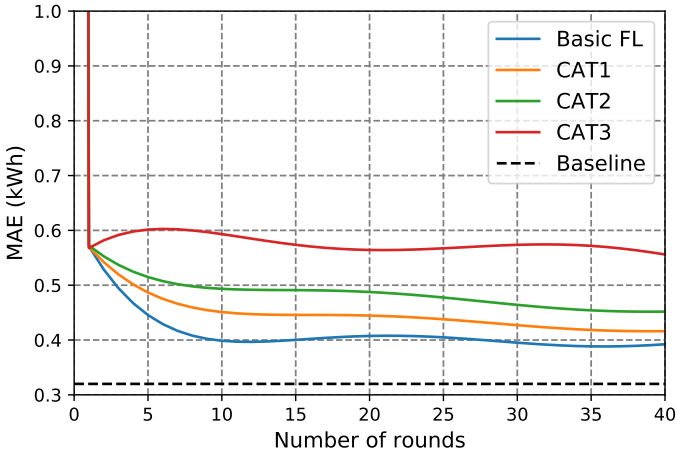
In this paper, we have proposed a privacy-preserving and communication-efficient FL-based approach for energy prediction in net-metering systems. In particular, we have proposed a multi-data-source hybrid DL-based predictor that considers both the customers' historical net readings and solar irradiance values to accurately predict future net readings. This predictor is trained through multiple rounds of FL. To preserve the customers' privacy, they encrypt their local models' parameters before sending them to the utility server while allowing the utility server to only obtain the aggregated parameters to build the global predictor. To save the communication bandwidth, we have used a CAT approach during the transmission of the updated model's parameters. Specifically, only the parameters whose values have sufficiently changed from previous round are transmitted. For the non-transmitted parameters, the previous values are used instead. Our evaluations indicate that the proposed predictor accurately predicts future net readings for both participating and non-participating customers in the FL training. Also, the evaluations indicate that the proposed approach preserves the customers' privacy while



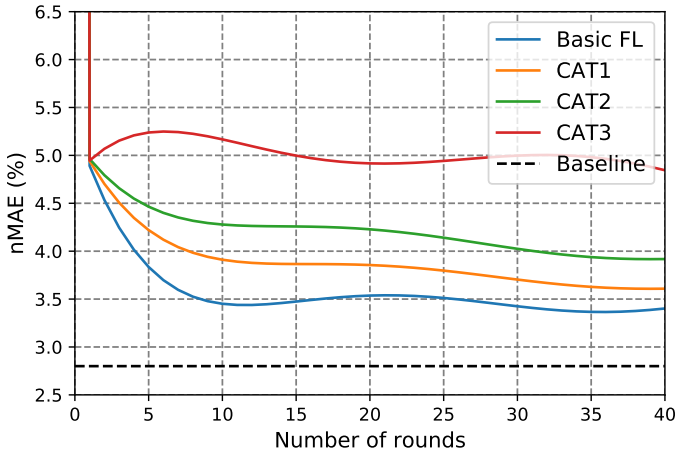

(a) MSE.



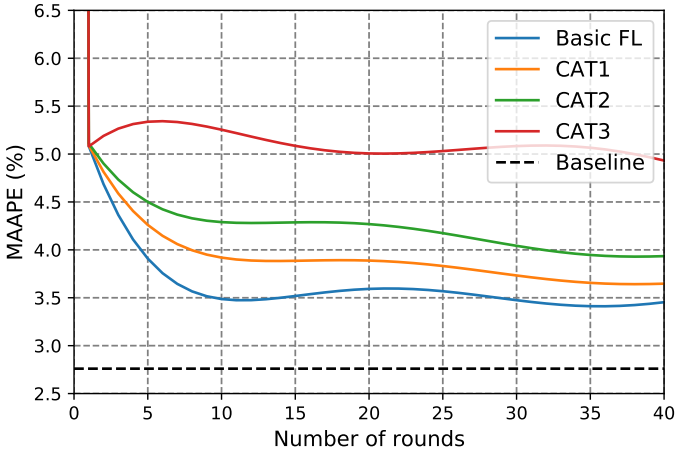

(b) RMSE.



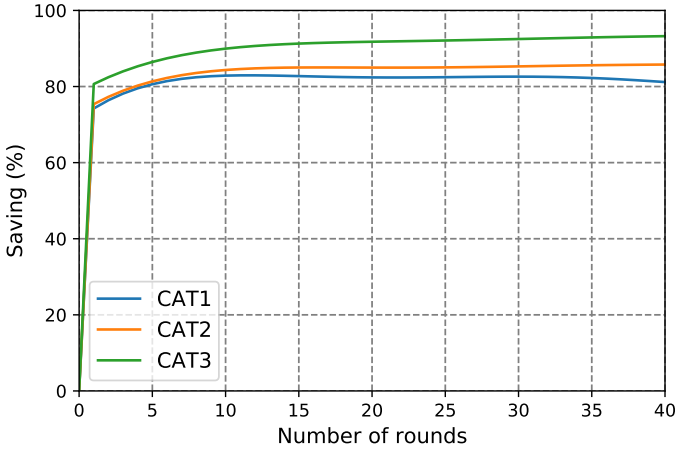

(c) MAE.




(d) nMAE.




(e) MAAPE.




(f) Saving.

Fig. 8: Comparison between basic FL and CAT approaches in terms of MSE, RMSE, MAE, nMAE, MAAPE, and communication saving.

using the communication bandwidth efficiently. Specifically, our approach reduces the communication overhead by over 96% compared to the existing approaches based on the Paillier cryptosystem. Furthermore, the CAT approach provides over 90% saving in the communication bandwidth without significantly deteriorating the predictor performance.

## ACKNOWLEDGEMENT

This research work was funded by Makkah Digital Gate Initiative under grant no. (MDP-IRI-7-2022). Therefore, authors gratefully acknowledge technical and financial support from Emirate of Makkah Province and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

## REFERENCES

[1] 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," *Proc. of IEEE International Symposium on Networks, Computers and Communications (ISNCC'20), Montreal, Canada*, 2020.

[2] V. B. Krishna, C. A. Gunter, and W. H. Sanders, "Evaluating detectors on optimal attack vectors that enable electricity theft and DER fraud," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 4, pp. 790–805, 2018.

[3] M. Ismail, M. F. Shaaban, M. Naidu, and E. Serpedin, "Deep learning detection of electricity theft cyber-attacks in renewable distributed generation," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3428–3437, 2020.

[4] M. M. Badr, M. I. Ibrahem, M. Mahmoud, M. M. Fouda, F. Alsolami, and W. Alasmary, "Detection of false-reading attacks in smart grid net-metering system," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1386–1401, 2022.

[5] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2019.

[6] J. Li, Y. Ren, S. Fang, K. Li, and M. Sun, "Federated learning-based ultra-short term load forecasting in power internet of things," *In Proc, of IEEE International Conference on Energy Internet (ICEI)*, pp. 63–68, 2020.

[7] S.-J. Huang and K.-R. Shih, "Short-term load forecasting via ARIMA model identification including non-gaussian process considerations," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 673–679, 2003.

[8] L. Ghelardoni, A. Ghio, and D. Anguita, "Energy load forecasting using empirical mode decomposition and support vector regression," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 549–556, 2013.

[9] Z. Xie, R. Wang, Z. Wu, and T. Liu, "Short-term power load forecasting model based on fuzzy neural network using improved decision tree," *In Proc. of IEEE Sustainable Power and Energy Conference (iSPEC)*, pp. 482–486, 2019.

[10] K. Amarasinghe, D. L. Marino, and M. Manic, "Deep neural networks for energy load forecasting," in *IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 2017, pp. 1483–1488.

[11] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, vol. 11, no. 7, 2018.

[12] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1087–1088, 2018.

[13] A. Almalaq and J. J. Zhang, "Evolutionary deep learning-based energy consumption prediction for buildings," *IEEE Access*, vol. 7, pp. 1520–1531, 2019.

[14] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, "Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1591–1598, 2017.

[15] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—a novel pooling deep RNN," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2018.

[16] M. I. Ibrahem, M. M. Badr, M. Mahmoud, M. M. Fouda, and W. Alasmary, "Countering presence privacy attack in efficient AMI networks using interactive deep-learning," *Proc. of IEEE International Symposium on Networks, Computers and Communications (ISNCC'21), Dubai, UAE*, 2021.

[17] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[18] I. Yilmaz and A. Siraj, "Avoiding occupancy detection from smart meter using adversarial machine learning," *IEEE Access*, vol. 9, pp. 35 411–35 430, 2021.

[19] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 739–753.

[20] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 2015, p. 1322–1333.

[21] I. Damgård and M. Jurik, "A generalisation, a simpli. cation and some applications of paillier's probabilistic public-key system," in *International workshop on public key cryptography*. Springer, 2001, pp. 119–136.

[22] S. E. Razavi, A. Arefi, G. Ledwich, G. Nourbakhsh, D. B. Smith, and M. Minakshi, "From load to net energy forecasting: Short-term residential forecasting for the blend of load and PV behind the meter," *IEEE Access*, vol. 8, pp. 224 343–224 353, 2020.

[23] S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, and D. J. Wu, "Function-hiding inner product encryption is practical," in *International Conference on Security and Cryptography for Networks*. Springer, 2018, pp. 544–562.

[24] "Ausgrid's solar home electricity data," https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data, last accessed: Sep. 2020.

[25] "Solcast," https://solcast.com/historical-and-tmy/, last accessed: Sep. 2020.

[26] Y. Hong, Y. Zhou, Q. Li, W. Xu, and X. Zheng, "A deep learning method for short-term residential load forecasting in smart grid," *IEEE Access*, vol. 8, pp. 55 785–55 797, 2020.

[27] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," *in Advances in Neural Information Processing Systems*, pp. 2450–2462, 2018.

[28] A. F. Ganai and F. Khursheed, "Predicting next word using RNN and LSTM cells: Stastical language modeling," *Proc. of International Conference on Image Information Processing (ICIIP)*, pp. 469–474, Nov. 2019.

[29] M. Buzau, J. Tejedor-Aguilera, P. Cruz-Romero, and A. Gómez-Expósito, "Hybrid deep neural networks for detection of non-technical losses in electricity smart meters," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1254–1263, 2020.

[30] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 255–258, 1995.

[31] E. L. Ratnam, S. R. Weller, C. M. Kellett, and A. T. Murray, "Residential load and rooftop PV generation: An Australian distribution network dataset," *International Journal of Sustainable Energy*, vol. 36, no. 8, pp. 787–806, 2017.

[32] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *Advances in Cryptology — EUROCRYPT '97*, W. Fumy, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 256–266.

[33] B. King, "A point compression method for elliptic curves defined over gf (2 n)," in *International Workshop on Public Key Cryptography*. Springer, 2004, pp. 333–345.

[34] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 2019, pp. 1–11.

[35] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

[36] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *International Journal of Forecasting*, vol. 32, no. 3, pp. 669–679, 2016.

[37] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: a Python library for model selection and hyperparameter optimization," *Computational Science & Discovery, doi: https://doi.org/10.1088/1749-4699/8/1/014008*.

[38] J.-H. Im, H.-Y. Kwon, S.-Y. Jeon, and M.-K. Lee, "Privacy-preserving electricity billing system using functional encryption," *Energies*, vol. 12, no. 7, p. 1237, 2019.

## APPENDIX A: SECURITY PROOF

In this appendix, we follow the methodology in [38] to prove the security of the proposed privacy-preserving data aggregation (PPDA) scheme. First, we review the security of the inner-product functional encryption (IPFE) scheme [23]. Then, we prove the security of the PPDA scheme.

### *A. Review of Security for the Inner-Product Function Encryption (IPFE) Scheme*

The existing inner-product encryption schemes, including the IPFE scheme [23], consider an indistinguishability notion of security (IND-security) [23]. In this subsection, we review the security notion for the IPFE scheme. In [23], an experiment between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ that can make key generation and encryption oracles queries is defined as follows:

**Definition 1** (Experiment $\text{Expt}_b^{\text{IPFE}-\text{IND}}$ [23]). *Let $b \in \{0,1\}$. The challenger $\mathcal{C}$ calculates $(\mathcal{PP}, \mathcal{MSK}) \leftarrow$ **Setup**$(1^\lambda)$ and gives $\mathcal{PP}$ to the adversary $\mathcal{A}$. Then, $\mathcal{C}$ responds to each oracle query type made by $\mathcal{A}$ as follows.*

- **Key generation oracle.** *Given a pair of non-zero vectors $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathbb{Z}_q^n \backslash \{0\}$, the challenger $\mathcal{C}$ calculates and returns* $\text{sk} \leftarrow$ ***KeyGen***$(\mathcal{MSK}, \boldsymbol{x}_b)$.
- **Encryption oracle.** *Given a pair of non-zero vectors $\boldsymbol{y}_0, \boldsymbol{y}_1 \in \mathbb{Z}_q^n \backslash \{0\}$, the challenger $\mathcal{C}$ calculates and returns* $\text{ct} \leftarrow$ ***Encrypt***$(\mathcal{MSK}, \boldsymbol{y}_b)$.

*Finally, the adversary $\mathcal{A}$ outputs a bit $b'$, which is the output of the experiment* $\text{Expt}_b^{\text{IPFE}-\text{IND}}(\mathcal{A})$.

Then, the security of the IPFE scheme is defined using an indistinguishability notion as follows:

**Definition 2** (Admissibility of $\mathcal{A}$ [23]). *For an adversary $\mathcal{A}$, let $Q_1$ and $Q_2$ be the total number of key generation and encryption oracle queries made by $\mathcal{A}$, respectively. For $b \in \{0,1\}$, let $\boldsymbol{x}_b^{(1)}, \ldots, \boldsymbol{x}_b^{(Q_1)} \in \mathbb{Z}_q^n \backslash \{0\}$ and $\boldsymbol{y}_b^{(1)}, \ldots, \boldsymbol{y}_b^{(\text{Q}_2)} \in \mathbb{Z}_q^n \backslash \{0\}$ be the corresponding vectors that $\mathcal{A}$ submits to the key generation and encryption oracles, respectively. We say that $\mathcal{A}$ is admissible if for all $i \in [Q_1]$ and $j \in [Q_2]$, we have that:*

$$\left\langle \boldsymbol{x}_0^{(i)}, \boldsymbol{y}_0^{(j)} \right\rangle = \left\langle \boldsymbol{x}_1^{(i)}, \boldsymbol{y}_1^{(j)} \right\rangle.$$

**Definition 3** (IND-Security for the IPFE [23]). *We define an inner-product functional encryption scheme denoted as $\Pi_{IPFE}$ = (**Setup**, **KeyGen**, **Encrypt**, **Decrypt**) as fully-secure if for all efficient and admissible adversaries $\mathcal{A}$,*

$$\left|\Pr\left[\text{Expt}_0^{\text{IPFE}-\text{IND}}(\mathcal{A}) = 1\right] - \Pr\left[\text{Expt}_1^{\text{IPFE}-\text{IND}}(\mathcal{A}) = 1\right]\right| = \text{negl}(\lambda),$$

*where* $\text{negl}(\lambda)$ *is a negligible function in $\lambda$.*

**Theorem 1** ( [23]). *The inner-product functional encryption scheme $\Pi_{IPFE}$ is IND-secure in the generic group model.*

**Remark 1.** *The original statement in Theorem 1 as stated in [23] is that $\Pi_{IPFE}$ is simulation-based secure (SIM-secure) in the generic group model. However, it was also mentioned in Remark 2-5 in [23] that a SIM-secure scheme is also IND-secure. Therefore, we have merged the above two statements into Theorem 1. We refer to [23] for more details about the SIM-security and generic group model.*

### *B. Security for the Proposed Privacy-Preserving Data Aggregation (PPDA) Scheme*

In our threat model, we assume that the utility server is honest-but-curious, i.e., it may try to extract useful information about the local model parameters vector $X_i$ of a customer $C_i$. As our goal is to be secure against the honest-but-curious utility server, we define an IND-security for the PPDA scheme. Then, we prove the IND-security of the PPDA scheme using that of the IPFE scheme.

We begin by defining the following experiment between a challenger and an adversary $\mathcal{A}^*$ that can make encryption oracles' queries. This experiment is designed similarly to that in Definition 1. However, the adversary here queries the $K$ encryption oracles corresponding to the $K$ customers participating in the FL training.

**Definition 4** (Experiment $\text{Expt}_b^{\text{PPDA}-\text{IND}}$). *Let $b \in \{0,1\}$. The challenger calculates $(\mathcal{PP}, \mathcal{MSK}) \leftarrow$ **Setup**$(1^\lambda)$ and gives $\mathcal{PP}$ to the adversary $\mathcal{A}^*$. Then, the challenger responds to each oracle query type made by $\mathcal{A}^*$ as follows.*

- **Encryption oracle for customer $C_1$.** *Given a pair of non-zero vectors $X_{10}, X_{11} \in \mathbb{Z}_q^n \backslash \{0\}$, the challenger calculates and returns $CT_1 \leftarrow$ **Encrypt**$(\Delta_1, \mathcal{SK}_{C_1}, X_{1b})$.*
- **Encryption oracle for customer $C_2$.** *Given a pair of non-zero vectors $X_{20}, X_{21} \in \mathbb{Z}_q^n \backslash \{0\}$, the challenger calculates and returns $CT_2 \leftarrow$ **Encrypt**$(\Delta_2, \mathcal{SK}_{C_2}, X_{2b})$.*

  $\vdots$
- **Encryption oracle for customer $C_K$.** *Given a pair of non-zero vectors $X_{K0}, X_{K1} \in \mathbb{Z}_q^n \backslash \{0\}$, the challenger calculates and returns $CT_K \leftarrow$ **Encrypt**$(\Delta_K, \mathcal{SK}_{C_K}, X_{Kb})$.*

*Finally, $\mathcal{A}^*$ outputs a bit $b'$, which is the output of the experiment* $\text{Expt}_b^{\text{PPDA}-\text{IND}}(\mathcal{A}^*)$.

Then, the security of the PPDA scheme is defined using an indistinguishability notion as follows:

**Definition 5** (Admissibility of $\mathcal{A}^*$). *For an adversary $\mathcal{A}^*$, let $Q_1$, $Q_2$, …, and $Q_K$ be the total number of queries made by $\mathcal{A}^*$ on the encryption oracles for customers 1, 2, …, and $K$, respectively. For $b \in \{0,1\}$, let $X_{1_b}^{(1)}, \ldots, X_{1_b}^{(\text{Q}_1)} \in \mathbb{Z}_q^n \backslash \{0\}$, $X_{2_b}^{(1)}, \ldots, X_{2_b}^{(\text{Q}_2)} \in \mathbb{Z}_q^n \backslash \{0\}$, …, and $X_{K_b}^{(1)}, \ldots, X_{K_b}^{(\text{Q}_K)} \in \mathbb{Z}_q^n \backslash \{0\}$ be the vectors that $\mathcal{A}^*$ submits to the $K$ corresponding encryption oracles, respectively. We say that $\mathcal{A}^*$ is admissible if for all $i_1 \in [Q_1]$, $i_2 \in [Q_2]$, …, and $i_K \in [Q_K]$, we have that:*

$$\sum_{k=1}^{K} X_{k_0}^{(\text{i}_k)} = \sum_{k=1}^{K} X_{k_1}^{(\text{i}_k)}.$$

**Algorithm 1:** Construction of $\mathcal{A}$ using $\mathcal{A}^*$.

**Input:** Public parameters $\mathcal{PP}$.

Give $\mathcal{PP}$ to $\mathcal{A}^*$.
**while** *true* **do**
  **if** $\mathcal{A}^*$ *returns* $b'$ **then**
    **Return** $b'$.
  **end**
  Wait until $\mathcal{A}^*$ submits a query $Q$.
  **if** $Q = (X_{10}, X_{11})$ **then**
    - Set $\boldsymbol{y}_0 \leftarrow X_{10}$ and $\boldsymbol{y}_1 \leftarrow X_{11}$.
    - Submit $(\boldsymbol{y}_0, \boldsymbol{y}_1)$ to the corresponding encryption oracle and receive $CT_1$.
    - Provide $CT_1$ to $\mathcal{A}^*$.
  **else if** $Q = (X_{20}, X_{21})$ **then**
    - Set $\boldsymbol{y}_0 \leftarrow X_{20}$ and $\boldsymbol{y}_1 \leftarrow X_{21}$.
    - Submit $(\boldsymbol{y}_0, \boldsymbol{y}_1)$ to the corresponding encryption oracle and receive $CT_2$.
    - Provide $CT_2$ to $\mathcal{A}^*$.
  $\vdots$
  **else**
    - Set $\boldsymbol{y}_0 \leftarrow X_{K0}$ and $\boldsymbol{y}_1 \leftarrow X_{K1}$.
    - Submit $(\boldsymbol{y}_0, \boldsymbol{y}_1)$ to the corresponding encryption oracle and receive $CT_K$.
    - Provide $CT_K$ to $\mathcal{A}^*$.
  **end**
**end**

**Output:** A bit $b'$.

**Definition 6** (IND-Security for the PPDA scheme). *We define a privacy-preserving data aggregation scheme denoted as $\Pi_{PPDA}$ as fully-secure if for all efficient and admissible adversaries $\mathcal{A}^*$,*

$$\left|\Pr\left[\mathrm{Expt}_0^{\mathrm{PPDA-IND}}(\mathcal{A}^*)=1\right]-\Pr\left[\mathrm{Expt}_1^{\mathrm{PPDA-IND}}(\mathcal{A}^*)=1\right]\right| = \mathrm{negl}(\lambda).$$

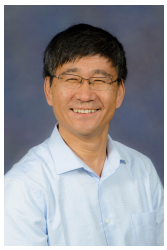
**Theorem 2.** *If $\Pi_{IPFE}$ is IND-secure (according to Definition 3) in the generic group model, then $\Pi_{PPDA}$ that is defined using $\Pi_{IPFE}$ is IND-secure (according to Definition 6) in the generic group model.*

**Proof of Theorem 2.** Following the methodology in [38], we perform a reduction proof as follows. First, we assume that there exists an efficient and admissible adversary $\mathcal{A}^*$. Then, we show that $\mathcal{A}^*$ can be used as a subroutine for the adversary $\mathcal{A}$. We design $\mathcal{A}$ such that it can simulate the customers' encryption oracles. In particular, instead of having a single encryption oracle based on the $\mathcal{MSK}$, we can have multiple encryption oracles, each based on a unique secret key of a certain customer. Thus, $\mathcal{A}$ can simulate the customers' encryption oracles by forwarding $\mathcal{A}^*$'s queries to the corresponding IPFE's encryption oracles. Algorithm 1 shows our construction of $\mathcal{A}$. It is straightforward to observe that if $\mathcal{A}^*$ is an admissible, i.e., a polynomial time algorithm, then Algorithm 1 is also admissible. Moreover, $\mathcal{A}$'s advantage is the same as that of $\mathcal{A}^*$, i.e., $\left|\Pr\left[\mathrm{Expt}_0^{\mathrm{IPFE-IND}}(\mathcal{A})=1\right]-\Pr\left[\mathrm{Expt}_1^{\mathrm{IPFE-IND}}(\mathcal{A})=1\right]\right| = \left|\Pr\left[\mathrm{Expt}_0^{\mathrm{PPDA-IND}}(\mathcal{A}^*)=1\right]-\Pr\left[\mathrm{Expt}_1^{\mathrm{PPDA-IND}}(\mathcal{A}^*)=1\right]\right|$. However, this construction contradicts Theorem 1, which states that an admissible $\mathcal{A}$ with non-negligible advantage does not exist. This completes the proof.

## BIOGRAPHIES

**Mahmoud M. Badr** received the B.S. and M.S. degrees in Electrical Engineering (electronics and communications) from Benha University, Cairo, Egypt in 2013 and 2018, respectively, and the Ph.D. degree in electrical and computer engineering from Tennessee Tech. University, TN, USA in 2022. He is currently an Assistant Professor at the Network and Computer Security: Cybersecurity Department, College of Engineering, State University of New York (SUNY) Polytechnic Institute, USA. He is also holding the position of a Lecturer Assistant at the Faculty of Engineering at Shoubra, Benha University, Egypt. He has been selected as a poster winner in Tennessee Tech. University's annual research and creative inquiry day, 2021. His research interests include machine learning, blockchain, cryptography, 5G networks, and smart grids.

**Dr. Mohamed M. E. A. Mahmoud** received PhD degree from the University of Waterloo in April 2011. From May 2011 to May 2012, he worked as a postdoctoral fellow in the Broadband Communications Research group - University of Waterloo. From August 2012 to July 2013, he worked as a visiting scholar in University of Waterloo, and a postdoctoral fellow in Ryerson University. Currently, Dr. Mahmoud is an associate professor in Department Electrical and Computer Engineering, Tennessee Tech University, USA. The research interests of Dr. Mahmoud include security and privacy preserving schemes for smart grid communication network, mobile ad hoc network, sensor network, and delay-tolerant network. Dr. Mahmoud has received NSERC-PDF award. He won the Best Paper Award from IEEE International Conference on Communications (ICC'09), Dresden, Germany, 2009. Dr. Mahmoud is the author for more than twenty three papers published in major IEEE conferences and journals, such as INFOCOM conference and IEEE Transactions on Vehicular Technology, Mobile Computing, and Parallel and Distributed Systems. He serves as an Associate Editor in Springer journal of peer-to-peer networking and applications. He served as a technical program committee member for several IEEE conferences and as a reviewer for several journals and conferences such as IEEE Transactions on Vehicular Technology, IEEE Transactions on Parallel and Distributed Systems, and the journal of Peer-to-Peer Networking.



**Dr. Yuguang Fang** (S'92, M'97, SM'99, F'08) received an MS degree from Qufu Normal University, Shandong, China in 1987, a PhD degree from Case Western Reserve University in 1994, and a PhD degree from Boston University in 1997. He joined the Department of Electrical and Computer Engineering at University of Florida in 2000 as an assistant professor, then was promoted to associate professor in 2003 and full professor in 2005 and has been a distinguished professor since 2019. He has held multiple professorships including the University of Florida Foundation Preeminence Term Professorship (2019-2022), University of Florida Research Foundation Professorship (2017-2020, 2006-2009), and University of Florida Term Professorship (2017-2021).

Dr. Fang received many awards including the US NSF CAREER Award in 2001, US ONR Young Investigator Award in 2002, 2018 IEEE Vehicular Technology Outstanding Service Award, 2019 IEEE Communications Society AHSN Technical Achievement Award, 2015 IEEE Communications Society CISTC Technical Recognition Award, 2014 IEEE Communications Society WTC Recognition Award, the Best Paper Award from IEEE ICNP (2006), 2010-2011 UF Doctoral Dissertation Advisor/Mentoring Award, and 2009 UF College of Engineering Faculty Mentoring Award. He has served as the Editor-in-Chief of IEEE Transactions on Vehicular Technology (2013-2017) and IEEE Wireless Communications (2009-2012), and has served on several editorial boards of journals including Proceedings of the IEEE (2018-present), ACM Computing Surveys (2017-present), ACM Transactions on Cyber-Physical Systems (2020-present), IEEE Transactions on Mobile Computing (2003-2008, 2011-2016, 2019-present), IEEE Transactions on Communications (2000-2011), and IEEE Transactions on Wireless Communications (2002-2009). He has been actively participating in conference organizations such as serving as the Technical Program Co-Chair for IEEE INFOCOM'2014 and the Technical Program Vice-Chair for IEEE INFOCOM'2005. He is a fellow of the IEEE and AAAS.

**Dr. Waleed Alasmary** (SM'19) received the B.Sc. degree (Hons.) in computer engineering from Umm Al-Qura University, Saudi Arabia, in 2005, the M.A.Sc. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2010, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, Canada, in 2015. During his Ph.D. degree, he was a Visiting Research Scholar with Network Research Laboratory, UCLA, in 2014. He was a Fulbright Visiting Scholar with CSAIL Laboratory, MIT, 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. He is currently an Associate Editor for the Array journal.

**Mohammed Abdulaal** is an assistant professor in the Department of Electrical and Computer Engineering at King Abdulaziz University, Saudi Arabia. He finished his PhD in 2019 at the school of Electrical and Electronic Engineering, the University of Manchester, UK. His PhD research involved design and implementation of a low-level Electroencephalography recognition system and Brain-computer Interface. He has done his MSc in 2014 from King Abdullah University for Science and Technology in Mechanical Engineering and his BEng in Mechatronic Engineering from The University of Manchester in 2012. His research interests include signal processing and machine learning of biomedical systems, Hajj research, traffic control systems, cybersecurity, and various image processing applications.

**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, and the Ph.D. degree in electrical and computer engineering from Tennessee Tech. University, USA, in 2021. He is currently an Assistant Professor with the Department of Cyber Security Engineering, George Mason University, USA. Dr. Ibrahem received Eminence Award for the Doctor of Philosophy Best Paper from Tennessee Tech. University, USA. He is also holding the position of Assistant Professor at the Faculty of Engineering at Shoubra, Benha University, Egypt. His research interests include machine learning, cryptography and network security, and privacy-preserving schemes for smart grid communication and AMI networks.

**Abdulah Jeza Aljohani** received the B.Sc (Eng.) degree in electronics and communication engineering from King Abdulaziz University, Jeddah, Saudi Arabia, in 2006, and the M.Sc. degree with distinction and Ph.D. degree, awarded with no corrections, in wireless communication from the University of Southampton, Southampton, U.K., in 2010 and 2016, respectively. He is currently an Assistance Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include machine learning, and optimization, distributed source coding, free-space Optical Communication, channel coding, cooperative communications, and MIMO systems.