Differential Private Federated Learning for Privacy-Preserving Third Party Service Framework in Advanced Metering Infrastructure

Xiao-Yu Zhang [1], Jose Cordoba-Pachon [2], Chris Watkins [2], and Stefanie Kuenzel [2]

[1]Royal Holloway
[2]Affiliation not available

October 30, 2023

Abstract

The advanced metering infrastructure (AMI) is a compulsory component of the future smart grid; it not only provides near real-time two-way communication between the consumers and the utility but also gives an opportunity to third parties to provide relevant value-added services to the consumers to improve the user satisfaction. However, existing services require the consumers share their private energy data with other parties, which has potential privacy risks. To better balance the excellent quality of the services and privacy guarantee, a novel differential private federated learning-based third-party service platform is proposed. Instead of sending the original energy data to the cloud server, the central server in the proposed scheme only collects the model parameters, which are trained locally inside the consumers' houses. Then the collected parameters are aggregated differential privately to eliminate the identity of individuals, and the aggregated parameters are used to update the central model and improve the model performance. Furthermore, a novel attention-based bidirectional long short-term memory neural network model is adopted to make predictions. In the case study, a residential short term load forecasting task is implemented to evaluate the performance of the proposed model; from the simulation results, the conclusion is made that the proposed model can achieve similar accuracy as the typical centralized model and better control the privacy loss flexibly at the same time.

Differential Private Federated Learning for Privacy-Preserving Third Party Service Framework in Advanced Metering Infrastructure

Xiao-Yu Zhang, *Student Member, IEEE,* José-Rodrigo Córdoba-Pachón, Chris Watkins,and Stefanie Kuenzel, *Senior Member, IEEE*

***Abstract*— The advanced metering infrastructure (AMI) is a compulsory component of the future smart grid; it not only provides near real-time two-way communication between the consumers and the utility but also gives an opportunity to third parties to provide relevant value-added services to the consumers to improve the user satisfaction. However, existing services require the consumers share their private energy data with other parties, which has potential privacy risks. To better balance the excellent quality of the services and privacy guarantee, a novel differential private federated learning-based third-party service platform is proposed. Instead of sending the original energy data to the cloud server, the central server in the proposed scheme only collects the model parameters, which are trained locally inside the consumers' houses. Then the collected parameters are aggregated differential privately to eliminate the identity of individuals, and the aggregated parameters are used to update the central model and improve the model performance. Furthermore, a novel attention-based bidirectional long short-term memory neural network model is adopted to make predictions. In the case study, a residential short term load forecasting task is implemented to evaluate the performance of the proposed model; from the simulation results, the conclusion is made that the proposed model can achieve similar accuracy as the typical centralized model and better control the privacy loss flexibly at the same time.**

***Index Terms*—Federated learning, differential privacy, advanced metering infrastructure, attention based deep learning, edge computing, Internet of Things (IoT)**

I. Introduction

THE large-scale smart meter roll-out plans and the fast development of advanced metering infrastructure (AMI) pave the way for the next generation smart grid. As the essential component in the smart grid that is installed at end-users, the smart meter continuously monitors and reports the individual power consumption data in near real-time. The high-resolution power consumption data provided by the smart meter benefits both the consumers, the power system operators, as well as third parties [1].Third parties have a keen interest in this personal data since it has excellent commercial value [2]. A variety of third-party services (TPSs) is available to the consumers, including demand response, non-intrusive load monitoring (NILM), energy awareness, load forecasting, etc. As for traditional TPS in AMI, the power consumption data collected by the smart meters is uploaded to a centralized server. The server can use the data to train machine learning/ deep learning models, and then the trained model can make predictions. However, these centralized TPSs and the data collected by the smart meter are subject to severe privacy concerns. Firstly, most TPSs require consumers to send detailed power consumption profiles of the house or specific appliances with timestamps. Attacks such as NILM attacks [3] can extract detailed behaviour patterns of the consumers by disaggregating the power consumption into detailed appliance usages. Secondly, referring to data privacy legislation such as the European Commission's General Data Protection Regulation (GDPR) [4], data collected by the smart meter belongs to the personal data, the collection or storage of such information is strictly limited by the data minimization principle and the consent principle [5]. Moreover, the European Commission also suggested the TPSs should have separate communication channels where the type of data to be collected and stored should be specified [6].

However, little attention has been paid to privacy issues on TPSs; moreover, these services are one of the main motivations for energy suppliers or third parties to get involved in the roll-out plans. Although a few privacy-preserving techniques , such as rechargeable battery [7], noise-adding method [8], data anonymization method [9], provide a strong privacy guarantee to the consumers, few words emphasise how the TPSs are conducted under these proposed frameworks. In [10], M. Asghar *et al.* provides several suggestions and outlooks for the future privacy-preserving TPSs; these suggestions can be concluded as follows: (1) implement TPSs on the customers' private computing platforms (such as mobile phones, personal computers). (2) develop new privacy-preserving distributed machine learning algorithms to provide better privacy guarantees to the consumers. Differential private federated learning (FL) is a suitable technique that can satisfy all suggestions proposed in [10]; this decentralized machine learning scheme enables clients to train local models without

This work is supported by the Leverhulme Trust.

X. Zhang and S. Kuenzel are with the Department of Electronic Engineering, Royal Holloway, University of London, TW20 0EX, U.K. (e-mail: stefanie.kuenzel@rhul.ac.uk)

J. R. Cordoba-Pachon is with the School of Management, Royal Holloway, University of London, TW20 0EX, U.K.

C. Watkins is with the Department of Computer Science, Royal Holloway, University of London, TW20 0EX, U.K.

sharing the private data with the server. Moreover, differential privacy (DP) provides a stronger privacy guarantee when the cloud server collects model parameters from the clients [11].

To the best of the author's knowledge, little work focuses on PPDL-based TPSs before. The most relevant works are the applications of federated learning in power system discipline. Federated learning has been adopted to solve problems such as solar irradiation forecasting [12], electricity consumer characteristics identification [13], energy management [14]. In [12], a federated BayesLSTM-based probabilistic solar irradiation forecasting is introduced, distributed PV stations from different locations collaborate to make accurate irradiation forecasting. This work also combined the conventional LSTM model with the variational Bayesian inference to provide uncertainty estimation with quantified confidence. Y. Wang *et al.* [13] proposed an FL-based electricity consumer characteristics identification framework; the FL framework enables retailers who are not willing to share their smart meter data work together to train model to identify sociodemographic characteristics (e.g., employment status, retirement status, number of residents). However, these applications are limited on the interactions between retailers/ PV stations and the server; little work emphasizes the customer-level application.

As for consumer-level FL application, based on the DP-SGD algorithm, X. Zhang *et al*. designed a demand-side management framework. The consumers will upload their private electricity data to the cloud server while a random Gaussian noise is added to protect the dataset, then the model trained by the server is sent to IoT devices where the consumers can send a query and obtain feedback. The limitation of this work is the consumers need to send the private data to the server, while the malicious servers still can infer data. Moreover, as the model is only trained by the server, the model cannot be personalized to different consumers. S. Lee et al. introduced a federated reinforcement learning-based house-level energy management system [14]. The FL-based system can train a reinforcement model that can better manage the distributed generation, energy storage system and appliance usages inside smart homes. Moreover, privacy-preserving appliance load scheduling methods are proposed based on differential privacy [15] and MPC [16], respectively.

Based on the knowledge gaps discussed above, the significant novelties can be summarized as follows.

(1) A privacy-preserving AMI TPS platform based on differential private federated learning (DPFL) scheme is proposed. The platform can provide multiple services to consumers without sharing their personal data (e.g., load demand data) to cloud server and other parties.
(2) An attention bidirectional long short-term memory (ATT-BLSTM) algorithm, which is one of the newest RNN model, is utilized as the local/central model to train the data and make predictions.
(3) K-means clustering is used to cluster the clients into the normal clients and the malicious clients by using the local model weights only.

II. The Preliminaries

A. Privacy and Data Ethics Requirement

From the privacy and data ethics aspect, the proposed system should guarantee data privacy at first; this means that both the cloud server as well as any other parties cannot observe and learn any information from the clients' data [18]. More specifically, the privacy definition of federated learning can be classified into global privacy and local privacy. Global privacy relies on a trusted server and ensures the model updates are private to other third parties [19], while local privacy assumes the server is also untrusted and the updates are private to the server as well. Taking global differential privacy and local differential privacy as an example, in global differential privacy, the data aggregator in the cloud server is trustworthy and can access the actual raw data, and the data aggregator is responsible for adding noise into the output of the database. In contrast, local differential privacy requires individuals to add random noise to their own database before sending it to the server. In this way, even the server or aggregation cannot access the actual database.

B. Adversary Model

In this paper, the central server is trustworthy, and the behavior of the server is honest. There are two adversarial models are taken into consideration in this research, which is a malicious client and an external adversary. The external adversary can use membership inference attacks [20] and model inversion attacks [21] to infer private information from the system. For model inversion attacks, once the adversary has already obtained a part of the personal data that belongs to the training data, the adversary can further infer more private data by only observing the inputs and outputs of a machine learning model. When it comes to membership inference attacks, the adversary can determine whether a given individual's data is inside the training data even without any previous private information. As for the malicious client, these clients may send bad and low-quality model updates to the global model. As a result, the accuracy of the global model is influenced, and even worse, the overall system may collapse. In the proposed system, the system should defend the attacks from both malicious servers as well as malicious clients.

III. Methodology

In this section, the methodologies to construct the privacy-preserving third-party service scheme are introduced. The methodologies adopted in this paper include attention-based recurrent neural networks and differential private federated learning.

A. Attention-Based Bidirectional Long Short-Term Memory Recurrent Neural Network

The main disadvantage of the conventional LSTM model is it can only utilize information from the past. To overcome the drawback, a bidirectional LSTM (BLSTM) is proposed by Schuster & Paliwal in 1997 [24]. In a BLSTM structure, given a minibatch input $\boldsymbol{X}_t \in \Re^{n\times d}$ (n is the number of examples, and d is the sequence size of each example), the forward hidden

state $\overrightarrow{\boldsymbol{h}_t} \in \Re^{n\times h}$ and backward hidden state $\overleftarrow{\boldsymbol{h}_t} \in \Re^{n\times h}$ (h denotes the number of hidden units) at time step t can be expressed as (7) and (8):

$$\overrightarrow{\boldsymbol{h}_t} = \phi(\boldsymbol{X}_t \boldsymbol{W}_{xh}^{(f)} + \overrightarrow{\boldsymbol{h}}_{t-1} \boldsymbol{W}_{hh}^{(f)} + \boldsymbol{b}_h^{(f)}) \quad (1)$$

$$\overleftarrow{\boldsymbol{h}_t} = \phi(\boldsymbol{X}_t \boldsymbol{W}_{xh}^{(b)} + \overleftarrow{\boldsymbol{h}}_{t-1} \boldsymbol{W}_{hh}^{(b)} + \boldsymbol{b}_h^{(b)}) \quad (2)$$

where $\boldsymbol{W}_{xh}^{(f)}, \boldsymbol{W}_{xh}^{(b)} \in \Re^{d\times h}$, $\boldsymbol{W}_{hh}^{(f)}, \boldsymbol{W}_{hh}^{(b)} \in \Re^{h\times h}$ represent weights of the model, and $\boldsymbol{b}_h^{(f)}$, $\boldsymbol{b}_h^{(b)} \in \Re^{1\times h}$ are the biases of the model. Then by integrating the forward and backward hidden state, the hidden state is obtained as $\boldsymbol{h}_t \in \Re^{n\times 2h}$. Finally, H_t is fed to the output layer to compute the output $\boldsymbol{O}_t \in \Re^{n\times q}$ (q is the number of outputs):

$$\boldsymbol{h}_t = \left[\overrightarrow{\boldsymbol{h}_t}; \overleftarrow{\boldsymbol{h}_t}\right]^T \quad (3)$$

$$\boldsymbol{O}_t = \boldsymbol{h}_t \boldsymbol{W}_{hq} + \boldsymbol{b}_q \quad (4)$$

where $W_{hq} \in \Re^{2h\times q}$ is the weight, and $b_q \in \Re^{1\times q}$ is the bias of the output layer. The attention mechanism is a probability weighting mechanism that was first proposed in 2014 [25]. Attention-based BLSTM architecture improves its accuracy by assigning the probability weights to each previous hidden state to find the most informative for the output at the current time step [26] (Fig. 1). Hence, the utilization of the attention mechanism can improve the output of the BLSTM and better solve the long-term memory problem [26]. Denoting the current hidden state as $\boldsymbol{h}_t$, and the previous hidden state as $\boldsymbol{h}_i$ ($1 \le i < t$). Referring to the definition in [25], a context vector c_t is computed, which is the weighted sum of all hidden states:

$$\boldsymbol{c_t} = \sum_{i=1}^{t-1} \alpha_{t,i} \boldsymbol{h_i} \quad (5)$$

where $\alpha_{t,i}$ is the weight for the hidden state $\boldsymbol{h_i}$ at timestep t. An attention matrix $\alpha_{t,i}$ is obtained by adopting softmax function, as shown in (11) and (12):

$$\boldsymbol{\alpha_t} = [\alpha_{t,1}, \alpha_{t,2}, \dots, \alpha_{t,(t-1)}] \quad (6)$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{T} \exp(e_{t,k})} \quad (7)$$

In the above equations, $e_{t,i}$ represents the score (or energy) of a feedforward neural network (denoted as function a), the purpose of $e_{t,i}$ is to capture the influence of previous hidden state $\boldsymbol{h}_i$ to the current hidden state $\boldsymbol{h}_t$. Three a functions are introduced in [27], which are location-based attention function (*location*), general attention function (*general*), and concatenation-based attention function (*concat*) [25]. Detailed functions are illustrated below:

$$e_{t,i} = a(e_{t,k}) = \begin{cases} \boldsymbol{W}_e^{\top} \boldsymbol{h}_i + b_e & Location \\ \boldsymbol{h}_t^{\top} \boldsymbol{W}_e \boldsymbol{h}_i & General \\ \boldsymbol{v}_e^{\top} \tanh(\boldsymbol{W}_e[\boldsymbol{h}_t; \boldsymbol{h}_i]) & Concat \end{cases} \quad (8)$$

where $\boldsymbol{v}_e$ is the parameter to be learned by the neural network. Referring to the experiment implemented by [28], attention-based BLSTM achieves excellent performance in processing power consumption data as its characteristic in allocating the importance to the overall power consumption data points that corresponding to the state changes of appliances. As a result, the model can better extract relevant features from the collected data.

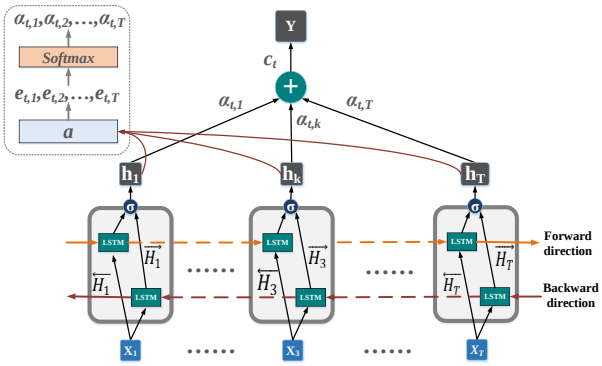

Fig. 1. Structure of attention based bidirectional LSTM.

B. Differential Privacy

Proposed by C. Dwork in 2006, differential privacy is a technology to protect an individual's identification information by adding random noise over the original aggregated data; every individual has little effect on the result [29, 30]. In this case, the adversary cannot distinguish the change of the aggregated data with/without one individual data. There are several noise addition mechanisms available in the literature [31], including the Laplace mechanism, Exponential mechanism, and Gaussian mechanism. The privacy level, ε, is guaranteed via the above noise addition mechanism, and the lower ε, the higher the privacy level can be achieved.

Definition 1. $\Re$ is a random function that transforms input β to a random output $\Re(\beta)$.

Definition 2. $d(\beta, \beta')$, which is the distance between two neighbouring datasets, represents the minimum number of individual samples required to shift dataset β to β'.

Definition 3. For a random function f, the global sensitivity, S_f, is the maximum difference between the outputs of two neighbouring datasets β and β'. S_f also determines the overall noise to be added into the DP mechanism.

$$\Delta f = \max_{d(\beta,\beta')=1} \|f(\beta) - f(\beta')\| \quad (9)$$

Definition 4. The Gaussian privacy mechanism denoted $\Re$, is defined as f plus noise term $\mathcal{N}$.

$$\Re(\beta) \triangleq f(\beta) + \mathcal{N}(0, \Delta f^2 \sigma^2) \quad (10)$$

where $\mathcal{N}$ is the Gaussian distribution with mean 0 and standard deviation ${S_f}^2\sigma^2$. And the scale σ is computed as

$$\sigma = \sqrt{2\ln\left(\frac{1.25}{\delta}\right)} \Delta_2 / \varepsilon \quad (11)$$

Definition 5. A randomized function $\Re$ satisfies (ε, δ) privacy $\mathbb{P}_{\mathbb{R}}$ for any two neighboring datasets β and β':

$$\mathbb{P}_{\mathbb{R}}[\Re(\beta) \in \varepsilon] \le e^{\varepsilon} \mathbb{P}_{\mathbb{R}}[\Re(\beta') \in \varepsilon] + \delta \quad (12)$$

where ε denotes all possible outcomes in range $\Re$, and δ is the possibility that the differential privacy is broken, in this paper, we select 10^{-5} as δ.

The overall privacy cost throughout the whole learning process is computed by following composition theorem:

Theorem 1. *(Composition Theorem) If f is (ε_1, δ_1)-differential privacy and g is (ε_1, δ_2)-differential privacy, then*

$f(D), g(D)$ is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-Differential Privacy (13)

with the composition theorem, the overall privacy cost is calculated by accumulating the privacy cost at each training step. Hence, the overall privacy cost after T steps is:

$$\varepsilon_{total} = T\varepsilon;\ \delta_{total} = T\delta \tag{14}$$

As a deep neural network may contain a large number of training steps, the privacy cost generated by the naïve composition is too significant, and overall privacy cannot be guaranteed. Hence, a moments accountant theorem is proposed by M. Abadi *et al.* [32] to minimize the value of $\varepsilon_{total}, \delta_{total}$, the moments accountant method offers much tighter bounds by random down-sampling and tracking a bound on the moments of the privacy loss random variable.

Theorem 2. *(Moments Accountant) The overall learning process with T learning steps provides a ($q\varepsilon\sqrt{T}$, δ)-differential privacy guarantee with a moments accountant, where q is the fraction of data.*

Proof of Theorem 2. A detailed proof is given in [32].

C. Federated learning with differential privacy

An FL model contains $K \in \mathcal{N}^*$clients indexed by k and one cloud server denoted as S. The target of the FL algorithm is to minimize a local objective function that can be expressed as:

$$\min_{w\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m} f_i(w) \tag{15}$$

For client $k \in K$, a local model will be trained with their private data on an edge device (such as smartphone or laptop),

$$\forall k, w_{t+1}^k \leftarrow w_t - \eta\nabla\mathcal{L}(w_t) \tag{16}$$

The parameters of the local model w_{t+1}^k for a client are then sent to S, and the parameters of all local models are aggregated, and a data-weighted average over all parameters is performed to update the global model w_{t+1}:

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \tag{17}$$

where n_k is the number of samples of client k, and n is the number of samples of all clients. Then the new global model is broadcasted to clients, and clients will retrain the local model with their data. The above steps will be repeated until convergence.

Although federated learning models avoid sharing private data with a cloud server or third parties, privacy is still a significant concern. By continuously sharing parameters of local models, the adversary can still infer some sensitive information from the parameters [33]. Differential private federated learning provides a strong privacy guarantee and reduces the communication cost simultaneously [34]. Hence, in this work, a DPFL algorithm is adopted to provide a stronger privacy guarantee to the system. The DPFL adopted in this paper is based on the randomized Gaussian mechanism, which was introduced in [35]. Denote the global model at timestep t as w_t, then the model is optimized locally by the local model of client k, we denote the optimized model as w^k. The mismatch between w_t and w^k is the client k's update and can be expressed as:

$$\triangle w^k = w^k - w_t \tag{18}$$

To reduce the sensitivity of $\triangle w^k$ with a considerable value, a scaling function is applied to $\triangle w^k$ to ensure the second norm $\|\triangle w^k\|_2$ is limited by sensitivity S. Hence, the scaled version of the updates is obtained as:

$$\triangle \bar{w}^k = \triangle w^k / max\left(1, \frac{\zeta^k}{S}\right) \tag{19}$$

where $\zeta^k = \|\triangle w^k\|_2$, S is the median of norms of clients' update and can be expressed as:

$$S = median\{\zeta^k\} \tag{20}$$

By adding random Gaussian noise scaled to S, $\mathcal{N}(0, S^2\cdot\sigma^2)$ into the sum of all scaled updates from K clients $\sum_{k=1}^{K} \triangle \bar{w}^k$, the Gaussian mechanism approximating the sum of updates is obtained. The new global model w_{t+1} is computed by adding the original global model with averaged approximation:

$$w_{t+1} \leftarrow w_t + \frac{1}{K}\left(\sum_{k=1}^{K} \triangle \bar{w}^k + \mathcal{N}(0, S^2\cdot\sigma^2)\right) \tag{21}$$

IV. System Model

In this section, a privacy-preserving third-party service framework in AMI based on differential private federated learning is introduced. To simplify the system, we adopt the following assumptions for the rest of the paper: (1) The sampling frequency, computation ability, types of data of all smart meters are the same; (2) latency and the communication delay is neglected; (3) All clients will upload the parameters at the same pace.

A. System Overall

The overall system is demonstrated in the flowchart shown in Fig. 2. The clients in this framework are the consumers who install smart meters at home; they use IoT devices such as smartphones, personal computers to train local models and communicate with the cloud server. The proposed framework contains six procedures that can be concluded as follows:

- *Procedure 1. Global model initialization.* In the beginning, the global model at the TP cloud server is initialized by allocating random values to its parameters. Then the model parameters are downloaded by clients and are broadcasted to local models.
- *Procedure 2. Local model training.* After receiving the parameters from the cloud server, the local model is updated in the IoT device; then, the IoT device will train the new model with private data locally.
- *Procedure 3. Local model parameters upload*. After the training process, the parameters of all local models are uploaded to the cloud server.
- *Procedure 4. Aggregation with differential privacy*. An aggregator is responsible for the secure aggregation once it received a response from the required number of clients. It aggregates the data with a random mechanism to maintain client-level differential privacy. After the aggregation of each round, the collected local model parameters are discarded.
- *Procedure 5. Global model update*. The global model is updated with the output of the aggregator.
- *Procedure 6. Model broadcast*. Parameters of the new global model are broadcasted to all local models which run at IoT devices.

B. Core Deep Neural Network Model

As shown in Fig. 3, the structure of the local and central model consists of seven layers:

- *The input layer*: The power consumption data collected by the smart meter is fed into the model.

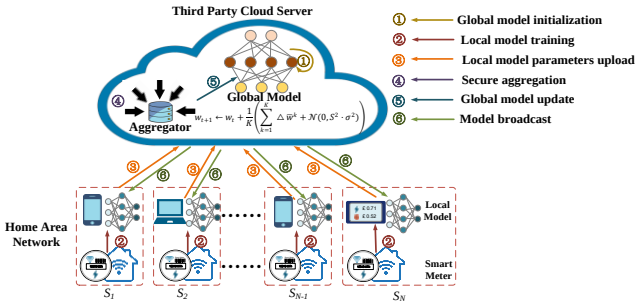

Fig. 2. Overall differential private federated third-party service scheme.

- *Two BLSTM layers*: BLSTM is adopted to extract high-level representation from the input data. Although more BLSTM layers enable the model to better extract nonlinear features from the input sequences, too many BLSTM layers will cause overfitting problems and the training time is also highly extended. Considering the above issues, two BLSTM layers are easier to implement with high efficiency.
- *An attention layer*: As introduced in Section 3.1, the attention layer utilizes the attention mechanism to rank the importance of the previous hidden states and selects the most informative hidden state to predict the output values.
- *A concatenated layer*: As the optional layer, the function of the concatenated layer is to load data from external databases that are related to the evaluation of desired TPS. The external databases could be the meteorological database, the calendar database, the electricity market database, etc.
- *A fully connected layer*: The fully connected layer links the recurrent layers with the output layer. The purpose of the layer is to fully extract the nonlinear correlation between all input variables and outputs.
- *The output layer*: As for classification tasks, the probability of each category is generated as the output; as for regression tasks (such as load forecasting or NILM), the prediction value at the current timestep is generated by the output layer.

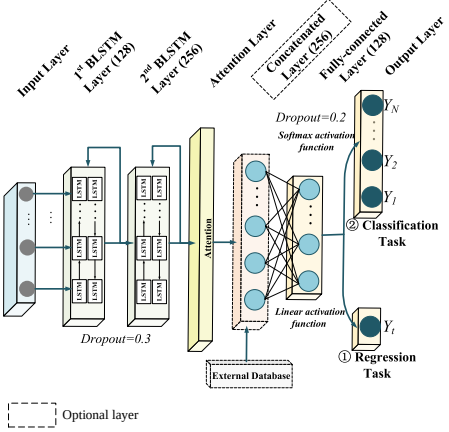

Fig. 3. Structure of local neural network model.

C. Cloud server

The central cloud server is responsible for malicious data detection, secure aggregation and central model update. Detailed description is presented as follows.

1) Malicious client detection

Malicious clients sent bad and low-quality updates to the cloud server that cloud fail the FL system, the distributions of the parameters from these malicious models are distinctive from the data of regular clients. The cloud server should detect these malicious clients efferently to prevent the system collapse. The clustering mechanism outputs two clusters, which are the normal client group and the malicious clients group.

2) Secure aggregation with differential privacy

In each communication round, once the server receives the uploaded local models from all clients, it will implement a secure aggregation with differential privacy. As introduced in Section 3.4, random Gaussian noise is added to the sum of the clipped updates. Then the aggregated updates are utilized to update the global model on the server. See Algorithm 1.

V. Result and Discussion

In this section, the accuracy and efficiency of the proposed DPFL Attention-BLSTM TPS framework are validated by using the scheme for a typical TPS residential STLF task. Both the proposed scheme and the traditional centralized framework are tested with real-world datasets. Moreover, the impacts of exogenous meteorological and calendar features are also investigated. Finally, the privacy performance, as well as the communication cost, is studied as well.

A. Data description

In this paper, a real-world dataset provided by Pecan Street Dataport [36] is used to evaluate the forecasting performance. The dataset contains over 1200 houses and is collected in Austin, Texas, the United States (N 30° 15', W 97° 43') between 1st January 2018 and 31st December 2018. Both household and appliance power consumption in each house is recorded with the sampling frequency of 1 min and 15 min, respectively. In this paper, 15 min interval smart meter data from 50 houses are selected as the simulation dataset. The dataset is split into training data (1st January 2018 to 30th September 2018) and testing data (1st October 2018 to 31st December 2018). And the training data is split into 36-week data, one-week data is adopted for each communication round; when the communication round reaches 36, it will start dragging data from the first week again at the next communication round until it reaches the threshold of δ.

B. Implementation

1) Simulation environment

The case study is implemented on a workstation with a Core i7-7700HQ CPU, NVIDIA GTX 1060 GPU (8 cores), and 8GB RAM. The DPFL ATT-BLSTM is operated on Python 3.6 with Pytorch [37], and the privacy loss is computed via the Tensorflow-Privacy library [38].

2) Evaluation metrics

The performance of the scheme is evaluated with Normalized Mean Absolute Error (nMAE), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE). The smaller value of MAE, MAPE, RMSE, the better performance the model provides.

$$nMAE = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N P_{max}} \quad (22)$$

$$MAPE = \frac{\sum_{i=1}^{N} |(y_i - \hat{y}_i)/y_i|}{N} \times 100\% \quad (23)$$

$$\mathrm{n}RMSE = \sqrt{\frac{(\sum_{i=1}^{N} [y_i - \hat{y}_i]^2)}{N}} \quad (24)$$

3) Benchmark model

To better demonstrate the accuracy and robustness of the proposed method, several benchmark models are designed. Firstly, the proposed model is compared with three different service frameworks, such as centralized framework, localized framework, as well as FL framework without adding noise during the aggregation process:

(1) Conventional centralized ATT-BLSTM model (denote as Centralized model).
(2) FL ATT-BLSTM model without DP (denote as FL model).
(3) Localized ATT-BLSTM model (denote as Localized model). In the Localized model, the smart meter can only train the DNN model with minimal data (we assume the smart meter can only store the data in the last week due to data regulation in this paper).

Then three benchmark models under the DPFL framework but utilizing different DNN algorithms (MLP, LSTM, BLSTM) are selected. By comparing the proposed model with the models listed below, the efficiency of ATT-BLSTM can be validated.

(4) DPFL model utilizes LSTM as a training algorithm (denote as DPFL-LSTM model).
(5) DPFL model utilizes BLSTM as a training algorithm (denote as DPFL-BLSTM model).
(6) DPFL model utilizes MLP as a training algorithm (denote as DPFL-MLP model).

4) Hyperparameters configuration

The hyperparameters of the pre-training model and the proposed DPFL Attention-BLSTM TPS model are summarized in Table 1. The pre-training model is a shallow MLP with only one dense layer. The number of the layer contains 16 cells, and the activation function of the dense layer is the Rectified Linear Unit (ReLU), which enables the model to learn nonlinear correlations better. The optimizer is SGD with the learning rate 1×10^{-3}.

As shown in Fig. 3 and Table I, the DPFL ATT-BLSTM model contains two BLSTM layers, with 128 and 256 cells, respectively. Followed by an attention layer with size 28 and one dense layer with 128 cells. The activation function of hidden layers is ReLU, and the optimizer is Adam with the learning rate 1×10^{-4}. As the STLF task is a regression task, the size of the output layer is one. Moreover, dropout and L2 regularization are used to avoid overfitting problems. 0.3 and 0.2 are selected as the dropout rates of the BLSTM layer and

Algorithm 1: Differential Private Federated Learning-based Third-Party Service. Clients number $\widetilde{K}$ indexed by k; communication round t; the maximum communication round T; the maximum pre-train communication round T_p; B is the mini-batch size; q is the fraction of clients; ε is the target differential privacy; σ is the Gaussian Mechanism parameter; δ represents the probability that ε-DP is broken, and Q is the threshold for δ.

Procedure Pretraining $(\widetilde{K}, w_t)$
  for client k in $\widetilde{K}$ **do**
    $w^k \leftarrow Local(k)$ ▷ Pretraining the local models to obtain the weights
  $C \leftarrow$ K-MeansClustering(2,$\triangle w$) ▷ Cluster clients into normal/abnormal clusters
  return $C_1, C_2, K, \widetilde{K}$ ▷ return the normal clients cluster and client number C_1 and K, and abnormal clients cluster and client number C_2 and $\widetilde{K}$

Procedure DPFL(K, w_t)
  initialize the global model w_0 ▷ initialize weights of the global model on the server
  initialize Accountant (ε, K) ▷ initialize the privacy accountant on the server
  while $r < R$ **do**
    $\delta \leftarrow$ Accountant (ε, q) ▷ accumulate the privacy loss
    if $\delta > Q$ **then** return w_t ▷ return the model when the privacy threshold reached
    for client k in qK **do**
      $\triangle w_{t+1}^k, \zeta^k \leftarrow$ ClientUpdate(k, w_t) ▷ the client k's update and norm update on local model
    $S = median\{\zeta^k\}$ ▷ compute the median of norms of clients' update as sensitivity
    $w_{t+1} \leftarrow w_t + \frac{1}{m}\left(\sum_{k=1}^{K} \triangle w_{t+1}^k / max\left(1, \frac{\zeta^k}{S}\right) + \mathcal{N}(0, S^2 \cdot \sigma^2)\right)$ ▷ update the global model by adding averaged approximation
  return w_{t+1}
Procedure ClientUpdate(k, w_t) ▷ perform on client k
  $w \leftarrow w_t$
  while $r < r_{max}$ **do**
    for $b \in$ B **do**
      $w \leftarrow w - \eta \nabla \mathcal{L}(w_t)$ ▷ mini-batch gradient descent
    $\triangle w_{t+1} = w^k - w_t$ ▷ client k's local model update
    $\zeta = \|\triangle w_{t+1}\|_2$ ▷ second norm update
  return $\triangle w_{t+1}, \zeta$
Procedure K-MeansClustering $(X, \triangle w)$
  random place centroids C_1, C_2 across $\triangle w$
  repeat
    for i in K **do**
      $\gamma_{ij} = \begin{cases} 1 \text{ if } j = \text{argmin}_j \|\triangle w_i - C_j\|^2 \\ 0 \text{ otherwise} \end{cases}$ ▷ find the nearest cluster j for model i
    for j in 2 **do**
      $n_j = \sum_{i=1}^{K} \gamma_{ij}$ ▷ assign the data points to clusters
      $C_j = \frac{1}{n_j} \sum_{i=1}^{K} \gamma_{ij} \triangle w_i$ ▷ assign the average of points to cluster j
  until Convergence
  return C_1, C_2 ▷ assign the regular clients to C_1 and the malicious clients to C_2
Procedure Accountant (ε, q)
  $\delta = 2q\delta\sqrt{t}$ ▷ moments accountant
  return δ

the dense layer, respectively. And 1×10^{-3} is selected as the weight decay value.

TABLE I
HYPERPARAMETER CONFIGURATION

Pre-training model	
Hyperparameter	Value/range
Layers	1 Fully connected layer with 16 cells
Batch size	32
Activation function	ReLU
Epochs	3
Optimizer	SGD
Learning rate	1e-3
Dropout rate	0.3
Differential privacy federated learning model	
Hyperparameter	Value/range
Lookback	4
Optimizer	Adam
Loss	MSE
Activation function	ReLU
Layers	2 BLSTM layers with 128 and 256 cells, respectively; 1
Epochs for each client in every communication round	5
Privacy budget ε	1, 2, 4, 6, 8, 10, 12
δ	1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8
The GM parameter σ	1.12
Number of batches per client B	128
Dropout rate	0.5
Weight decay	1e-3
Attention size	28
Learning rate	1e-4
Total clients	5, 10, 50
Percentage of clients selected each round q	30%

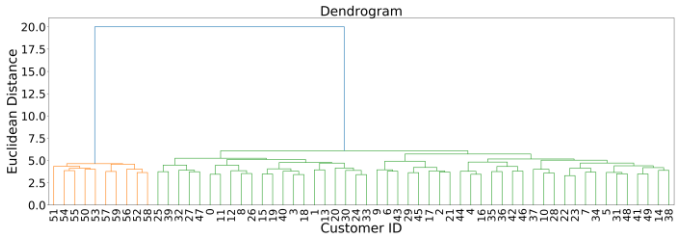

Fig. 4. Structure of local neural network model.

C. Clustering the consumers and detecting the malicious clients

Before we are implementing the DPFL algorithm, a pre-training process is evaluated to filter out the malicious clients based on the similarity of the weights. In this case study, 50 regular consumers (ID number between 0 and 49) and 10 abnormal consumers (ID number from 50 to 59) are included. The malicious clients will upload fake weights to the central server; the generated fake weights follow the Gaussian Distribution. The purpose of introducing the shallow neural network is to reduce the communication and computation cost during the pre-training process. Moreover, the simple network has limited learning ability that can reduce the sensitivity of the shared weights. As shown in Fig. 4, the central server applies the K-means clustering algorithm to the collected weights, and the algorithm detects all the malicious clients and classifies these models into the same group. Meanwhile, the rest of the clients is clustered into another group. The Euclidean Distance of the regular consumers is below 7, which is considerably small compared to the distance between the malicious clients and the normal clients.

D. Comparison of the Proposed Model with Centralized and Localized Models

After filtering out the malicious clients, the federated model is operating among all regular clients. In the first case study, the proposed DPFL scheme is compared with the conventional Centralized scheme, Localized scheme, as well as the normal FL scheme. To control the variable, all schemes utilize ATT-BLSTM as the DNN algorithm. The forecasting results are concluded in Table II. Fig. 5 plots the predicting load curves by the four schemes as well as the ground truth curve (solid blue line) in three consumers' houses. Considering the accuracy of the forecasting, the centralized scheme has the best performance, as the centralized scheme can access all consumers' data without any constraints. Accessing a more significant amount of the data will help the central model better learn the characteristics of the loads among all houses and avoid the overfitting problems, which will decrease the accuracy significantly. However, the centralized scheme suffers from significant privacy risks as all consumers must send their personal demand data continuously. The regular FL scheme almost achieves equal accuracy as the centralized scheme, especially when client number K increases. From Table II, when $K = 50$, nRMSE of the values forecasted by the FL scheme reaches 6.67%, which is only 2.33% less than the Centralized scheme. This simulation result confirms that the FL can achieve very similar forecasting performance as the Centralized scheme without sharing the real-measured data to the cloud server at all. In other words, the FL scheme can satisfy the functionality requirement without scarifying individuals' privacy.

In the Localized scheme, it disconnects communication with the cloud server, and all computation processes are completed within the smart meter and personal devices. As defined in Section V.B., the smart meter can only preserve the last one-week demand data for privacy concerns, so the Localized scheme has minimal data to train the local model. From the predicted curve shown in Fig. 5, the Localized scheme failed to predict the demand load in most situations. Also, the high nRMSE and nMAE errors presented in Table II convinced the conclusion that the Localized scheme does not reach a balance between privacy and accuracy.

The DPFL scheme, which is the privacy-enhanced version of the normal FL scheme, can make a prediction with performance merely worse than the two schemes mentioned above. This is due to the privacy constraints set by DP. The privacy level of the DPFL scheme can be adjusted flexibly by setting the two DP parameters, the privacy budget ε and the probability of information being leaked δ. Typically, smaller ε means a smaller distance between the two neighbouring databases when sending a query. Hence the adversary has difficulty in distinguishing these two databases by observing the query output. Hence, a smaller ε provides better privacy but less accuracy at the same time. From the results shown in Table II, when ε=8 and δ=10^{-5}, the performance of the DPFL scheme is 3.75% and 12.31% worse than the FL scheme from the perspective of nRMSE and MAPE. Although the accuracy of the DPFL scheme stays below non-differentially private schemes, it is significantly better than the Localized scheme that only trains the model with its own data.

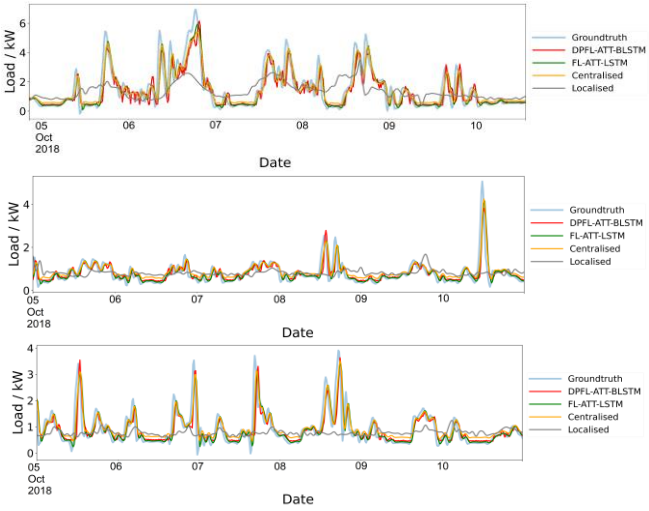

Fig. 5. Short-term load forecasting results of three houses predicted by proposed differential private federated learning scheme and three conventional schemes (ε=8, δ=10^{-5}).

TABLE II
LOAD FORECASTING PERFORMANCES OF THE PROPOSED MODELS AND BENCHMARK MODELS

Model	ε	K	MAPE (%)	nMAE (%)	nRMSE (%)	C.R.	CC	CPC
DPFL-MLP	1	5	221.40	32.99	35.25	1	1	0.71
		10	99.60	26.52	28.97	1	3	0.69
		50	76.51	16.51	20.38	1	15	0.73
	4	5	78.67	25.16	26.16	6	6	13.50
		10	70.82	9.06	11.59	3	9	10.93
		50	70.41	7.68	10.99	3	45	41.33
	8	5	162.87	20.32	21.64	36	25	80.72
		10	69.58	8.08	10.85	15	45	88.04
		50	63.68	8.32	10.50	18	221	332.70
DPFL-LSTM	1	5	257.20	29.51	31.87	1	1	1.45
		10	146.04	15.08	19.39	1	3	1.56
		50	75.12	10.63	13.06	1	15	1.47
	4	5	94.83	14.14	17.41	6	6	30.17
		10	71.55	7.40	10.65	3	9	24.06
		50	71.43	11.61	13.30	3	45	94.08
	8	5	73.88	15.65	21.91	36	35	307.18
		10	68.31	7.57	10.97	15	45	345.70
		50	62.43	7.24	9.94	18	221	1422.23
DPFL-BLSTM	1	5	176.51	21.41	24.60	1	1	2.10
		10	152.10	12.13	17.11	1	3	2.26
		50	102.31	11.54	16.72	1	15	2.26
	4	5	79.17	15.46	16.49	6	6	45.89
		10	72.92	14.64	15.67	3	9	35.63
		50	70.98	9.72	12.13	3	45	150.98
	8	5	69.67	17.21	18.73	36	35	693.44
		10	65.95	10.01	11.68	15	45	718.48
		50	61.37	6.16	9.30	18	221	3159.73
DPFL ATT-BLSTM	1	5	323.89	16.27	20.44	1	1	3.29
		10	400.23	19.89	23.09	1	3	3.29
		50	376.45	29.65	41.20	1	15	4.98
	4	5	51.21	20.73	21.44	7	6	95.39
		10	40.38	7.13	10.53	3	6	42.81
		50	36.35	5.68	8.07	3	45	172.61
	8	5	29.06	4.49	8.04	36	35	307.22
		10	24.67	4.36	7.52	15	45	339.93
		50	14.44	4.32	6.92	18	221	1526.36
FL ATT-BLSTM	—	5	19.62	4.19	7.45	50	50	441.57
		10	17.20	3.76	6.70	50	147	1151.88
		50	12.59	3.70	6.67	50	735	4631.15
Centralised ATT-BLSTM	—	—	10.34	2.87	4.34	—	—	434.83
Localised ATT-BLSTM	—	—	68.73	10.01	10.69	—	—	29.21

E. Comparison of the Proposed Model with Other Algorithms

In the first case study, the proposed DPFL ATT-BLSTM model is compared with DPFL models that utilize different DNN algorithms (benchmark models (4)-(5)). The forecasting results of all models are shown in Table II, nMAE, nRMSE, and MAPE are used to measure the accuracy of the prediction results, and the communication cost, as well as computation cost, are recorded. The privacy budget ε range from 1 to 8, and the client number K ranges from 5 to 50. To visualize the performance of the proposed scheme and benchmark models, 30-minute forecasting results of random three houses are presented in Fig. 6 (under the condition ε=8, δ=10^{-5}). In each communication round, only 30% of clients (e.g., 15 clients when $K = 50$) are selected to participate in the training process. Unlike feeder-level load forecasting, which has a regular peak load every day, household-level load forecasting is more challenging as the load profile in different days vary a lot. From the figure, DPFL-ATT-BLSTM performs best among all algorithms, and it is observed that the load curve predicted by the proposed DPFL-ATT-BLSTM model (solid red curve) tracks the ground truth load curve (solid blue curve) in most cases, both the peak part and the off-peak part are predicted with high accuracy. Considering the evaluation metrics, the proposed model has the lowest MAPE, nRMSE, and nMAE values in the same comparison group. Referring to the results shown in Table II, when ε=8 and δ=10^{-5}, the nRMSE and nMAE value of the proposed model reduces 31.95% and 11.22% compared to the DPFL-BLSTM.

Meanwhile, DPFL-MLP (light green solid curve) has the worst performance in most cases. Without the memory cell, it has very limited predictability in forecasting time-series data. From Fig. 6, DPFL-MLP neither track the peak load nor the off-peak load. However, there is also an advantage of this method: the computation cost is the least among all algorithms. In the situation when the computation ability of the edge devices is limited, this method could be the priority choice. DPFL-LSTM (solid orange curve) and DPFL-BLSTM (solid pink curve) models have similar prediction performances, while the DPFL-BLSTM model is slightly better. When ε=8 and δ=10^{-5}, the nRMSE values of DPFL-LSTM and DPFL-BLSTM are 9.94% and 10.17%, respectively.

These results demonstrate that the ATT-BLSTM is more efficient in process time-series data, especially the data is nonstationary and nonlinear. The reasons for the ATT-BLSTM's superior predicting performance can be summarized as follows: (1) The bidirectional structure enables the model to extract features from both forward and backward directions; (2) The attention mechanism helps the model find the most essential hidden state to the current output.

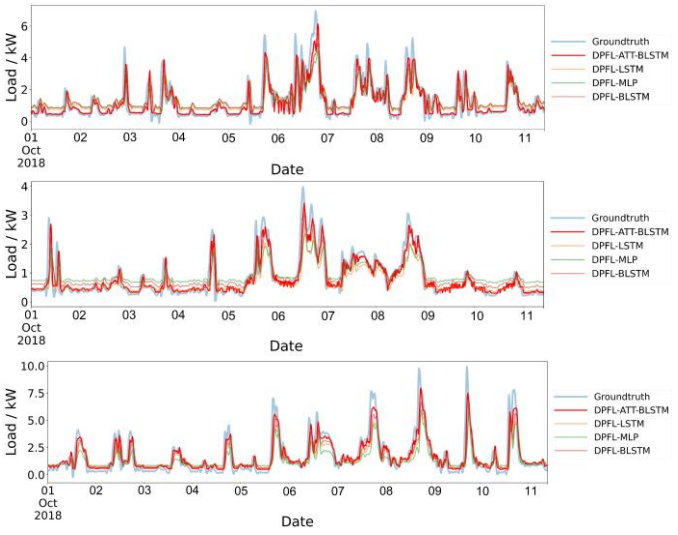

Fig. 6. Short-term load forecasting results of three houses predicted by four differential private federated learning models (ε=8, δ=10^{-5}).

F. *Influence of client number K*

Another vital parameter that influences the performance of the proposed DPFL scheme is the client number K. Table II presents the model performance for $K \in \{5, 10, 50\}$. We also record the privacy metrics, CRs, CC, and CPC for each case. Referring to [57], The choice of DP parameter δ is influenced by K and should obeys the following constrain:

$$\delta \ll \frac{1}{K} \tag{25}$$

This condition is to avoid protecting the majority of consumers' privacy by revealing a few consumers' [57]. According to this requirement, we set the threshold of δ, Q as 1×10^{-5}. From the Table II, it is found that under the DPFL scheme, more clients achieve higher model accuracy: When $K = 5$, the prediction error is considerable high, and when K increases to 50, the accuracy of the model almost reaches the same accuracy as non-DP schemes. This is because during the secure aggregation process, more clients will reduce the standard deviation of the additive noise. Based on the above simulation results, the conclusion is made that under the same privacy budget, more clients can efficiently reduce the accuracy cost.

G. *Influence of privacy budget ε*

In the DPFL scheme, the most important parameters to make the trade-off between privacy and accuracy are the two DP parameters ε and δ. Recall Algorithm 1, during the secure aggregation process in each communication round, given ε and GM parameter σ; the central server accountant evaluates δ [20]. The central server will continue the communication rounds until δ reaches the threshold Q, then the whole training process will be stopped, and the server sends the well-trained model to all clients. As defined in Section V, Q is selected as 1×10^{-5}. In this case study, the influence of different values of ε (7 values are chosen ranging from 1 to 12) on the model performance is investigated. From Fig. 7 (b), the DPFL-ATT-BLSTM scheme with small ε (1, 2, 4) reaches the threshold Q quickly within just a few communication rounds. However, the model accuracy is undesirable as nRMSE maintains a high level, even higher than the Localized scheme, the benchmark model with the worst performance. At this privacy level, although the privacy of the consumers is protected perfect, the functionality is sacrificed ultimately. In contrast, when ε is large enough (such as 10 or 12 in our case), it takes more communication rounds until σ reaches the threshold. More communication rounds allow the central model gets fully trained with frequent updates of its model parameters. Consequently, the model accuracy increases as ε become larger (As shown in Fig. 7 (a)). However, large ε allows less similarity of the outputs from different clients, and the adversary can distinguish different clients more effortless, and the model provides less privacy consequently. Hence, when ε between 6 and 8, the proposed scheme can efficiently make accurate load forecasting and provide a good level of privacy protection at the same time.

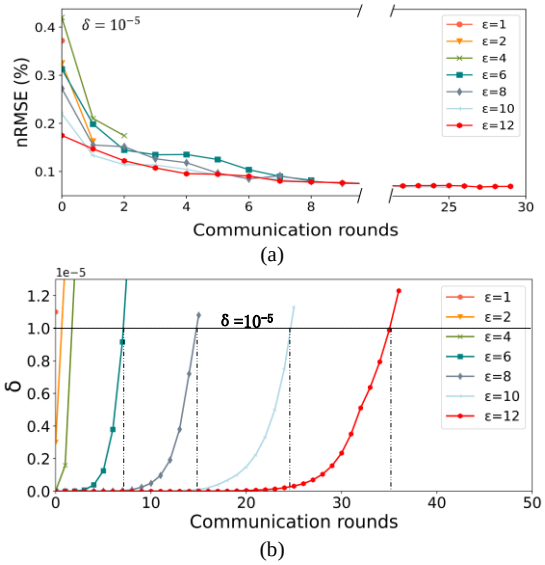

Fig. 7. (a) model performance of the differential private federated learning scheme with different levels of privacy budget; (b) accumulation of total δ with increasing communication rounds under different privacy budgets.

VI. Conclusion

In this paper, we design a novel privacy-preserving TPS platform by considering both privacy, security, and data ethics requirements. The platform is constructed based on the DPFL framework and utilizes state-of-the-art ATT-BLSTM as the training algorithm to train the local model and the central model. Instead of sending all private data to the cloud server, the consumers under the scheme will train the local model with their personal devices and only shared the model parameters to the central server. From security concerns, the proposed model introduced a pretraining process with K-Means clustering to filter out the malicious clients. To better protect privacy, the central server will add random Gaussian noise during the aggregation process to hide the clients' identities during the training. Therefore, the proposed scheme maintains client-level DP with low computation and communication costs. In the case study of household-level STLF, we evaluate the proposed scheme with benchmark models (Centralized model, Localized model, FL model, DPFL-LSTM model, DPFL-BLSTM model, DPFL-MLP model). After simulation, the following conclusions are made:

(1) Although the proposed DPFL-ATT-BLSTM scheme cannot achieve equal model accuracy as non-DP schemes (such as Centralized scheme and FL scheme), it can also track the ground truth load curve precisely. Most importantly, the proposed scheme overperforms much better than the Localized model, while the latter achieves complete privacy protection but minimal local data for model training.

(2) With the attention mechanism that can find the most critical hidden state to the current output, the ATT-BLSTM algorithm has superior performance in processing time-series load data rather than other benchmark algorithms.

The privacy budget ε, which influence the model performance, are also thoroughly investigated. From the simulation result, it is found that although smaller ε provide a

stronger privacy guarantee but sacrifices the model accuracy as the price; bigger ε provides better model performance but insufficient privacy protection. To make a trade-off between accuracy and privacy, an ε with a value between 6 and 8 is the best choice.

REFERENCES

[1] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges," *IEEE Transactions on Smart Grid,* vol. 10, no. 3, pp. 3125-3148, 2019.

[2] T. Morey, T. Forbath, and A. Schoop, "Customer data: Designing for transparency and trust," *Harvard Business Review,* vol. 93, no. 5, pp. 96-105, 2015.

[3] X.-Y. Zhang, C. Watkins, C. C. Took, and S. Kuenzel, "Privacy boundary determination of smart meter data using an artificial intelligence adversary," *International Transactions on Electrical Energy Systems,* vol. n/a, no. n/a.

[4] P. Voigt, and A. Von dem Bussche, "The eu general data protection regulation (gdpr): A Practical Guide" *A Practical Guide, 1st Ed., Cham: Springer International Publishing,* vol. 10, pp. 3152676, 2017.

[5] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials,* vol. 22, no. 3, pp. 2031-2063, 2020.

[6] R. t. t. E. Commission, *Essential regulatory requirements and recommendations for data handling, data safety, and consumer protection,* Technical Report version 1.0, 2011. https://ec. europa. eu/energy/sites/ener ….

[7] Y. Sun, L. Lampe, and V. Wong, "Smart meter privacy: Exploiting the potential of household energy storage units," *IEEE Internet of Things Journal,* vol. 5, no. 1, pp. 69-78, 2017.

[8] P. Barbosa, A. Brito, H. Almeida, and S. Clauß, "Lightweight privacy for smart metering data by adding noise," in Proceedings of the 29th Annual ACM Symposium on Applied Computing, 2014, pp. 531-538.

[9] C. Efthymiou, and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in 2010 first IEEE international conference on smart grid communications, 2010, pp. 238-243.

[10] M. R. Asghar, G. Dán, D. Miorandi, and I. Chlamtac, "Smart meter data privacy: A survey," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 4, pp. 2820-2835, 2017.

[11] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine,* vol. 37, no. 3, pp. 50-60, 2020.

[12] X. Zhang, F. Fang, and J. Wang, "Probabilistic solar irradiation forecasting based on variational Bayesian inference with secure federated learning," *IEEE Transactions on Industrial Informatics,* 2020.

[13] Y. Wang, I. L. Bennani, X. Liu, M. Sun, and Y. Zhou, "Electricity Consumer Characteristics Identification: A Federated Learning Approach," *IEEE Transactions on Smart Grid,* 2021.

[14] S. Lee, and D.-H. Choi, "Federated reinforcement learning for energy management of multiple smart homes with distributed energy resources," *IEEE Transactions on Industrial Informatics,* 2020.

[15] X. Liao, P. Srinivasan, D. Formby, and R. A. Beyah, "Di-PriDA: Differentially private distributed load balancing control for the smart grid," *IEEE Transactions on Dependable and Secure Computing,* vol. 16, no. 6, pp. 1026-1039, 2017.

[16] C. Rottondi, and G. Verticale, "Privacy-friendly appliance load scheduling in smart grids," in 2013 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2013, pp. 420-425.

[17] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences,* vol. 118, no. 17, 2021.

[18] Q. Zhang, C. Xin, and H. Wu, "Privacy Preserving Deep Learning based on Multi-Party Secure Computation: A Survey," *IEEE Internet of Things Journal,* 2021.

[19] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," *Advances in neural information processing systems,* vol. 27, pp. 2879-2887, 2014.

[20] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 3-18.

[21] 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, 2015, pp. 1322-1333.

[22] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in International conference on machine learning, 2013, pp. 1310-1318.

[23] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[24] M. Schuster, and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing,* vol. 45, no. 11, pp. 2673-2681, 1997.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473,* 2014.

[26] C. Raffel, and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *arXiv preprint arXiv:1512.08756,* 2015.

[27] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao, "Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks," in Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 1903-1911.

[28] V. Piccialli, and A. M. Sudoso, "Improving non-intrusive load disaggregation through an attention-based deep neural network," *Energies,* vol. 14, no. 4, pp. 847, 2021.

[29] C. Dwork, "Differential privacy: A survey of results," in International conference on theory and applications of models of computation, 2008, pp. 1-19.

[30] T. Ha, T. K. Dang, T. T. Dang, T. A. Truong, and M. T. Nguyen, "Differential privacy in deep learning: an overview." pp. 97-102.

[31] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: a survey," *IEEE Communications Surveys & Tutorials,* 2019.

[32] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 308-318.

[33] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in 28th Security Symposium (Security 19), 2019, pp. 267-284.

[34] N. Agarwal, A. T. Suresh, F. Yu, S. Kumar, and H. B. McMahan, "cpSGD: communication-efficient and differentially-private distributed SGD." pp. 7575-7586.

[35] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557,* 2017.

[36] O. Parson, G. Fisher, A. Hersey, N. Batra, J. Kelly, A. Singh, W. Knottenbelt, and A. Rogers, "Dataport and NILMTK: A building data set designed for non-intrusive load monitoring," in 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2015, pp. 210-214.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems,* vol. 32, pp. 8026-8037, 2019.

[38] C. Radebaugh, and U. Erlingsson, "Introducing tensorflow privacy: learning with differential privacy for training data," *Medium. com (accessed 2020-01-27). https://medium. com/tensorflow/introducing-tensorflowprivacy-learning-with-differential-privacy-for-trainingdata-b143c5e801b6,* 2019.