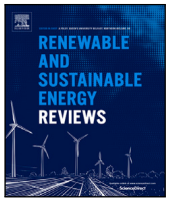
Renewable and Sustainable Energy Reviews 191 (2024) 114091

Contents lists available at ScienceDirect

Renewable and Sustainable Energy Reviews

journal homepage: www.elsevier.com/locate/rser



DP^2-NILM: A distributed and privacy-preserving framework for non-intrusive load monitoring

Shuang Dai [a,b], Fanlin Meng [c,*], Qian Wang [d], Xizhong Chen [e]

[a] Department of Mathematical Sciences, University of Essex, Colchester, UK
[b] Department of Engineering, University of Exeter, Exeter, UK
[c] Alliance Manchester Business School, University of Manchester, Manchester, UK
[d] Department of Computer Science, Durham University, Durham, UK
[e] Department of Chemical Engineering, School of Chemistry and Chemical Engineering, Shanghai Jiao Tong University, Shanghai, China

ARTICLE INFO

Keywords:
Deep neural network
Federated learning
Global differential privacy
Local differential privacy
Non-intrusive load monitoring
Privacy-preserving
Utility optimization

ABSTRACT

Non-intrusive load monitoring (NILM), which usually utilizes machine learning methods and is effective in disaggregating smart meter readings from the household level into appliance-level consumption, can help analyze the electricity consumption behaviors of users and enable practical smart energy and smart grid applications. Recent studies have proposed many novel non-intrusive load monitoring frameworks based on federated deep learning. However, there is a lack of comprehensive research exploring the utility optimization schemes and the privacy-preserving schemes in different federated learning-based NILM application scenarios. In this study, a distributed and privacy-preserving non-intrusive load monitoring (DP^2-NILM) framework was developed to make the first attempt to conduct federated learning-based NILM focusing on both utility optimization and privacy-preserving. Specifically, two alternative federated learning strategies are examined in the utility optimization schemes, i.e., the FedAvg and the FedProx. Moreover, different levels of privacy guarantees, i.e., the local differential privacy federated learning and the global differential privacy federated learning are provided in the DP^2-NILM. Extensive comparison experiments are conducted on three real-world datasets to evaluate the proposed framework.

1. Introduction

Modern urbanization, lifestyles, and technological advancements have increased the energy demand. The energy supply generates greenhouse gas emissions that accelerate climate change, which poses a significant threat to the security and prosperity of the global community. Legal obligations regarding climate change, such as those enacted in the UK, are placing increased strain on traditional centralized power grids. In response, the concept of smart grids has emerged. Smart grids promise a more reliable and intelligent power grid network utilizing information systems, which can significantly contribute to the decarbonization of the energy system and promote the use of renewable energy sources.

As a key part of a smart grid, smart meters allow non-intrusive appliance load monitoring (NILM) [1] to help smart meter clients reduce energy consumption by scheduling appliance usage hours and monitoring abnormal electricity usage patterns. The NILM is a growing trend in utilizing machine learning methods to monitor events (ON/OFF) or energy consumption of individual appliances using the aggregated smart meter reading of the whole building [1]. NILM provides real-time feedback on the energy consumption of smart meter clients, and research findings indicate that the appliance-level feedback can save from 3% to 18% annual energy consumption for the entire house [2].

Deep learning-based models have presented new opportunities for the electrical utility industry, and are the most representative structures applied to NILM [3–6], which have been proven to be more effective than other traditional models. However, most deep learning-based NILM models are centralized [7], which may not be feasible in the era of big data due to concerns about data privacy and excessive communication overhead from smart meters.

To address these challenges, studies have used federated learning (FL), an emerging paradigm for training models that can be tailored to individuals without relying on centralized data [8]. FL-based NILM models benefit from the collaboration of multiple data sources and

* Corresponding author.
E-mail addresses: sd19628@essex.ac.uk (S. Dai), fanlin.meng@manchester.ac.uk (F. Meng).

https://doi.org/10.1016/j.rser.2023.114091
Received 30 June 2022; Received in revised form 6 November 2023; Accepted 9 November 2023
Available online 22 November 2023

Nomenclature

Abbreviations

DNN	Deep Neural Network
FL	Federated Learning
GDPFL	Global Differential Privacy Federated Learning
IID	Independent and Identically Distributed
LDPFL	Local Differential Privacy Federated Learning
NILM	Non-Intrusive Load Monitoring
PSPNet	Pyramid Scene Parsing Network
SGD	Stochastic Gradient Descent

Notations/Symbols

$\Delta\mathcal{F}$	Maximum $\mathcal{L}_1$ distance
δ	Privacy relaxation term
ϵ	Privacy budget/loss
η	Learning rate
γ_t	Residual/unmonitored load at time t
$\hat{l}_t^i$	Predicted load of ith appliance at time t
$\hat{s}_t^i$	Predicted ON/OFF state of ith appliance at time t
λ^i	ON power threshold
$\mathcal{F}$	Query function
$\mathcal{L}_g$	Global model loss
$\mathcal{L}_j$	Loss of jth sample
$\mathcal{L}_n$	Loss of nth household
$\mathcal{M}$	Random algorithm
$\mathcal{N}$	Gaussian noise
μ	Proximal parameter
$\boldsymbol{L}$	Predicted load sequence
$\boldsymbol{S}$	Predicted ON/OFF state sequence
B_L	Local batch size
d_0	Minimum OFF duration
d_1	Minimum ON duration
$F(\cdot)$	Approximation function
I	Number of target appliances
L_t	Aggregated load at time t
l_t^i	Load of ith appliance at time t
N	Number of households
R	Global rounds
s_t^i	ON/OFF state of ith appliance at time t
T	Total monitoring period
w	Model parameters

Units

s	Second
W	Watt

privacy guarantees, making them more efficient than models trained solely on individual households. Even though the FL paradigm has obvious advantages for NILM, there are still challenges in real-world applications.

Firstly, when model parameters are exchanged between the central server and local clients, FL has been identified as vulnerable to privacy invasions [8,9]. The incorporation of differential privacy into deep learning [10] has been suggested as a potential method to provide privacy guarantees for FL-based NILM. While different clients may require different levels of privacy assurance, a single privacy-preserving scheme cannot accommodate the diverse needs of all smart meter clients. Secondly, different households use energy in different ways, which makes current FL-based NILM models ineffective for dealing with the heterogeneity of smart meter clients [11]. Consequently, different optimization schemes are necessary to deal with different types of datasets.

Existing FL-based NILM models have limited scope, either adopting naive FL without privacy guarantees [12,13] or addressing challenges from a single perspective [14,15]. However, in practical real-world NILM scenarios, different smart meter clients may have different requirements, making it challenging for FL-based models to meet all of these demands. This work innovatively presents the first DP^2-NILM framework to jointly optimize utility and preserve privacy for varied FL-NILM scenarios. The main contributions of the study are summarized as follows.

- This study proposes DP^2-NILM, the first framework to systematically explore both utility optimization and privacy preservation schemes for practical FL-NILM applications.
- DP^2-NILM addresses heterogeneity by examining relationships between two utility optimization schemes - FedAvg and FedProx - and NILM accuracy, and it achieves satisfactory performance in both the homogeneous and heterogeneous data environments.
- DP^2-NILM applies both central and local differential privacy to ensure data privacy. This is the first work that accommodates diverse privacy requirements based on different levels of differential privacy in real-world NILM.
- This study examines DP^2-NILM on three real-world datasets, providing new insights to satisfy heterogeneous client requirements and enabling broader adoption of FL for real-world NILM applications.

The remainder of this study is structured as follows. Section 2 reviews literature related to the proposed DP^2-NILM framework. Section 3 provides background knowledge and briefs the preliminaries used in DP^2-NILM. Section 4 overviews the three-tier workflow of the proposed DP^2-NILM. The utility optimization schemes and the privacy-preserving schemes of DP^2-NILM are detailed in Sections 5 and 6, respectively. The performance evaluations on real-world datasets are conducted in Section 7. The conclusion and possible future extensions are given in Section 8.

2. Review of FL-based NILM with enhancing mechanisms

2.1. State-of-the-art NILM

NILM was first proposed by Hart [16], which utilized heuristic methods based on combinatorial optimization to perform load disaggregation. After this, diverse models have been proposed to improve inference accuracy, which can be mainly divided into unsupervised learning and supervised learning approaches.

Unsupervised learning methods such as notable variations of hidden Markov model [17–19] and clustering analysis [20] have been commonly used in NILM studies. For instance, [19] developed a new infinite factorial hidden Markov model for NILM constrained on contextual features, which utilizes the usage information on the appliance-level to improve the disaggregation accuracy. [3] compared unsupervised learning (combinatorial optimization, factorial hidden Markov model) with supervised learning (DNN) for NILM, where the DNN-based model achieved the best performance. While hidden Markov model variations have shown effectiveness, they are computationally expensive and their performance declines as the number of appliances increases [21,22]. This scalability issue hinders their practical application in the NILM.

Alternative methods have been explored to address these limitations. For example, fuzzy C-means was employed [21] to cluster

appliance states, and dynamic time warping was utilized to extract appliance electricity consumption. This method demonstrated lower computational costs compared to the hidden Markov model. However, its performance heavily depends on the initial state and the complexity of the appliance usage patterns. Another study [22] combined the support vector machine with k-means clustering and found it outperformed the hidden Markov model as the number of appliances increased. However, this approach introduces additional complexity in model training and parameter tuning.

Supervised learning models, particularly deep neural network (DNN) based models [23–25] have been widely used in NILM, which provide new opportunities for the electrical utility industry [26]. However, DNN-based NILM models require diverse and substantial training data, posing a challenge in real-world scenarios where datasets are often isolated. Furthermore, integrating smart meter readings into a centralized database is difficult due to communication bandwidth limitations and data privacy legislation.

The emergence of federated learning [27] not only provides privacy guarantees for smart meter data but also solves the challenge of data isolation, which brings considerable benefits to DNN-based NILM models. Despite the potential advantages, applying FL to NILM has only begun to be explored in recent years [13,28–30]. For instance, [28] proposed a FederatedNILM framework to enable the NILM task in the FL paradigm at the residential level. [13] utilized the FL paradigm to improve the model performance for NILM in both residential and industrial scenarios. [29] proposed a FedNILM framework utilizing model compression to reduce the computation overhead while retaining satisfying performance for NILM. [30] adopted DP into the FL for NILM to provide stronger privacy protection, and a membership attack was included to evaluate the privacy guarantee level of the framework.

Despite the novelty of the above discussed state-of-the-art NILM, most NILM studies have focused only on the centralized training environment and relatively few have focused on FL-based NILM. It should also be noted that the majority of FL-based NILM studies focus on the basic framework rather than the comprehensive application functions required for practical NILM applications in a decentralized environment. To fill this research gap, this study evaluates several FL-based NILM scenarios based on two practical enhancement schemes for FL-based NILM to accommodate various real-world requirements from smart meter clients.

2.2. Enhancing mechanisms of federated learning

There are two main streams of approaches for enhancing the FL framework, i.e., the utility optimization schemes and the privacy-preserving schemes.

In recent years, many advanced utility optimization schemes have been proposed [31–36]. For example, [35] used FL for multi-task network anomaly detection, which improved the training efficiency compared with multiple single-task models. Later, [36] combined FL with the deep neural network to solve the similar problem. Moreover, transfer learning was adopted in this study to reconstruct the model for improving the anomaly detection performance. Among them, the most commonly used mechanism is the federated averaging (FedAvg) [31], which averages the updated gradients from the client models to optimize the global model. However, the FedAvg has been demonstrated to diverge empirically in scenarios where the data is non-independent and identically distributed (non-IID) across clients [31]. FedProx [37], which uses proximal terms to stabilize model updating, was then proposed as a solution to heterogeneity in federated networks.

Although many utility optimization mechanisms have been proposed, FL offers limited privacy guarantees. Prior studies have proposed differential private FL (DPFL) to provide clients with stronger privacy guarantees, which has been used as the basis for many privacy-preserving FL-based schemes [12,14,30,38]. Privacy in FL can be divided into global differential privacy FL (GDPFL) [39] and local differential privacy FL (LDPFL) [14] based on different noise adding mechanisms. In GDPFL, the trusted server applies the noise during the parameter aggregation, whereas in LDPFL, each participant adds noise to the model parameters before uploading them to the server.

Most existing studies only provided privacy guarantees for FL-based NILM at a fixed level. There has been limited exploration into privacy preservation schemes for FL at varying scales, such as global and local levels. Moreover, while the data heterogeneity in FL-based NILM scenarios is a practical and important characteristic due to different users having varied lifestyles and, accordingly, different electricity usage patterns, there is no existing research that has tackled this challenge from the utility optimization perspective. Therefore, this study makes the first attempt to explore FL-based NILM focusing on the utility optimization schemes and the privacy-preserving schemes by developing the DP^2-NILM framework and conducting extensive and comparative experiments on practical NILM scenarios based on real-world smart meter datasets.

3. Preliminaries

This section introduces several essential concepts related to the proposed DP^2-NILM framework.

3.1. Non-intrusive load monitoring

Given the aggregated load L_t at time t:

$$L_t = \sum_{i=1}^{I} l_t^i + \gamma_t, \tag{1}$$

the goal of NILM is to recover the status of I target electrical appliances. l_t^i and γ_t denote the load consumption for the ith appliance and the residual/unmonitored load respectively at time t, $1 \le t \le T$. NILM can be formulated as either a classification task or a regression task depending on the status variables of individual electrical appliances.

For the regression task, the NILM model aims to find the approximation, denoted as F, of the true relationship between the aggregated household-level consumption (L_t) and the appliance-level consumption

$$\boldsymbol{L} = [\hat{l}_t^1, \hat{l}_t^2, \dots, \hat{l}_t^i, \dots, \hat{l}_t^I] = F(L_t), \tag{2}$$

where $\boldsymbol{L}$ is the predicted load consumption sequence of I target electrical appliances at time t.

For the classification task, thresholds need to be set for the NILM model to determine the states (e.g., ON/OFF) of each target appliance. A commonly used threshold method is the activation-time thresholding proposed by [3], which could filter out false activation of the abnormal spikes by the minimum ON/OFF duration during the OFF state to better improve the inference accuracy [3]. For the sake of simplicity, this study assumes that there are two typical states (ON/OFF) for the target appliances, and the state s_t^i for ith appliance at time t is related to its threshold λ^i. The activation-time thresholding can be then described in Algorithm 1, and the classification task for NILM can be defined as

$$\boldsymbol{S} = [\hat{s}_t^1, \hat{s}_t^2, \dots, \hat{s}_t^i, \dots, \hat{s}_t^I] = F_s(L_t), \tag{3}$$

where $\hat{s}_t^i$ is a binary variable indicating the predicted ON/OFF state of ith electrical appliance at time t.

3.2. Federated deep learning

When data owners intend to combine their local data to train a common utility model, the traditional centralized approach is to pool their own private data at a central server, during which the data uploading and integration process are often restricted by data privacy legislation. To address this challenge, FL was brought up [27], which only requires the exchange of updated model parameters rather than the raw data between clients and the central server, and therefore is deemed to be the state-of-the-art approach for distributed data privacy protection.

Algorithm 1: Activation-time thresholding [3]

Input: ON power threshold λ^i of i-th appliance, minimum ON state duration d_1, minimum OFF state duration d_0
Output: s_t^i: the state of i-th appliance at time t, 0: OFF, 1: ON

```
Procedure:
if l_t^i < λ^i then
    if t < d_0 then
        s_t^i = 1
    else
        s_t^i = 0
    end
else
    if t < d_1 then
        s_t^i = 0
    else
        s_t^i = 1
    end
end
return s_t^i
```

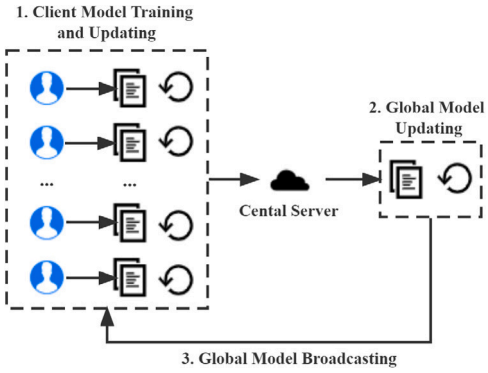


Fig. 1. Training process of the federated deep learning framework.

FL is a machine learning strategy aimed at training a high-quality global model while the raw private datasets are distributed locally in each client without the need to transfer them to a central server. Fig. 1 shows the training process of the federated deep learning framework, which can be described in three steps. Firstly, each client trains their local model and updates model parameters during each training round. Then, each client passes the updated parameters to a central server. Then, the global model aggregates the updated parameters from all local clients and updates its parameters accordingly in the central server. Finally, the updated global model parameters are then broadcast to each local client, and these three steps are iterated for multiple rounds until the convergence is reached.

3.3. Differential privacy

DP introduces noise into the raw dataset so that it provides statistical guarantees against the information a malicious adversary may infer from the output of a randomized algorithm [40].

Definition 1 (*Differential Privacy [10]*). A random algorithm $\mathcal{M}$ is compliant with (ϵ, δ)-DP if for any two neighboring input datasets L, L' and for any subset of outputs/events $\boldsymbol{S} \subseteq Rang(\mathcal{M})$,

$$\Pr[\mathcal{M}(L) \in \boldsymbol{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(L') \in \boldsymbol{S}] + \delta. \tag{4}$$

In Eq. (4), ϵ is the privacy budget/loss, which is inversely proportional to the privacy level. δ is the probability that the upper privacy bound is broken, i.e., the occurrence of a bad event. It is a plain ϵ-DP when δ equals 0.

In practical applications, the (ϵ, δ)-DP is enforced by a Laplacian or Gaussian mechanism that relies on the ϵ to characterize the sensitivity of $\mathcal{F}$. For a real-valued query function $\mathcal{F}$, a common exemplification is to calibrate an additive zero-mean Laplacian or Gaussian noise mechanism to the sensitivity of $\mathcal{F}$, which can be denoted as

$$\Delta\mathcal{F} = \max_{L,L'} \left\| \mathcal{F}(L) - \mathcal{F}(L') \right\|_1, \tag{5}$$

where $\Delta\mathcal{F}$ measures the maximum $\mathcal{L}_1$ distance between the results of $\mathcal{F}$ over the neighboring datasets L and L'.

The Gaussian mechanism adds Gaussian noises to $\mathcal{F}$ to satisfy (ϵ, δ)-DP: $\forall \delta \in (0, 1)$, the noise is denoted by $\mathcal{N}(0, \Delta\mathcal{F}^2 \cdot \sigma^2)$, resulting in

$$\mathcal{M}(L) = \mathcal{F}(L) + \mathcal{N}(0, \Delta\mathcal{F}^2 \cdot \sigma^2), \tag{6}$$

where $\Delta\mathcal{F} \cdot \sigma$ is the standard deviation, and $\sigma \geq \frac{\sqrt{2\ln(1.25/\delta)}}{\epsilon}$.

4. DP^2-NILM framework

4.1. Overview of DP^2-NILM

The DP^2-NILM framework aims to train different federated learning models based on utility optimization and privacy-preserving schemes according to different real-world NILM application scenarios. It is also important to note that the DP^2-NILM framework is easily extensible to incorporate various state-of-the-art DNN models and datasets. Fig. 2 presents the whole workflow of the DP^2-NILM framework, which contains three tiers.

- **Client Model Training Tier.** In this tier, smart meter readings from the client side are preprocessed into standard formats for the federated pipeline. The client can either specify their privacy-preserving or the data heterogeneity optimization requirements. After preprocessing, each client trains their data based on a state-of-the-art DNN model, which will be introduced in Section 4.2, and then uploads their parameters through the DP^2-NILM paradigm.
- **Federated Model Training Tier.** This tier is the key part of the DP^2-NILM framework. Based on the special requirement from the client model training tier, the DP^2-NILM assigns different federated learning mechanisms to each client. For example, a client-side requires a strict privacy-preserving mechanism to protect its sensitive data. After receiving this request, DP^2-NILM will deliver a high-level privacy-preserving paradigm, the local differential privacy federated learning (Section 6.2), to train the FL model based on the typical FL training steps.
  During FL training, the objective for N clients can be described as an optimization problem:

$$\begin{aligned} \min_{w_g} \mathcal{L}_g(w_g) &= \frac{1}{|L|} \sum_{n=1}^{N} |L^n| \cdot \mathcal{L}_n(w_n) \\ \text{where} \quad \mathcal{L}_n(w_n) &= \frac{1}{|L^n|} \sum_{j \in L^n} \mathcal{L}_j(w_j), \\ & \forall L^n \in L, n \in \{1, 2, \ldots, N\} \end{aligned} \tag{7}$$

  where $\mathcal{L}_g(w_g)$ is the loss of the global model, $\mathcal{L}_n(w_n)$ is the loss of the nth local client model, and $\mathcal{L}_j(w_j)$ is the loss of a single smart meter reading sample. $|L|$ is the sample length of the whole training set L, and $|L^n|$ is the sample length of the smart meter readings from the local household n. Each household $n \in \{1, 2, \ldots, N\}$, and generates its private smart meter readings $L^n = \{(L_1^n, s_1^n), \ldots, (L_t^n, s_t^n), \ldots, (L_T^n, s_T^n)\}$, where L_t^n is the aggregated load consumption of the target appliances for the nth local household client at time step t, and s_t^n is the corresponding states (ON/OFF) set of these appliances.
  The most commonly used optimization algorithm for FL is the FedAvg [31]. Based on the FedAvg, two subsequent research streams for enhancing the FL paradigm have been proposed, i.e., the utility optimization schemes and the privacy-preserving schemes. Following this development, the DP^2-NILM framework uses the FedAvg as the baseline to include the enhancing schemes.

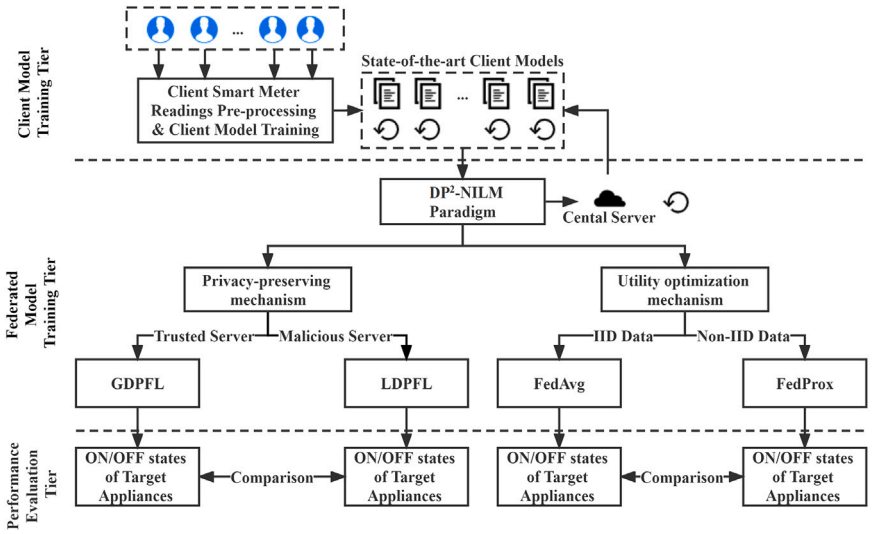


Fig. 2. The workflow of proposed DP^2-NILM framework.

Specifically, in the practical application scenarios, datasets may come from different types of households, and training models need to be considered in both homogeneous and heterogeneous environments. In [37], FedProx was proposed to solve the heterogeneity problem in federation learning by adding a proximal term in the training process and providing greater robustness to the federated learning framework. Therefore, the DP^2-NILM adopts the FedAvg and the FedProx to optimize the model utility for FL-based NILM in both homogeneous and heterogeneous environments. Furthermore, studies [39,41] have established robust theoretical foundations for GDPFL and LDPFL based on the FedAvg, whereas discussing different levels of privacy-preserving in the FL-based NILM framework has received limited attention. To fill this gap, the DP^2-NILM aims to satisfy different client-side requirements and provide privacy-preserving FL-based NILM at different levels by developing GDPFL and LDPFL in privacy-preserving schemes based on FedAvg.

- **Performance Evaluation Tier.** Different model training paradigms are designed for different NILM application scenarios based on three real-world smart meter datasets, and the models of each scenario are evaluated and validated in this tier.

In summary, the DP^2-NILM framework is a three-tiered approach that integrates federated learning with utility optimization and differential privacy to provide decentralized, privacy-preserving, and efficient solutions for NILM. The framework utilizes state-of-the-art DNN models for training on client devices and leverages the FedAvg and FedProx algorithms for optimization. Additionally, privacy preservation enhancements considering both global and local differential privacy are incorporated into the framework. The effectiveness of the framework is validated through performance evaluation based on real-world client requirements.

4.2. State-of-the-art NILM client model

This study introduces a state-of-the-art deep learning architecture, i.e., the pyramid scene parsing network (PSPNet) [42], to enhance the performance of DP^2-NILM in both the local client model training and the central server global model training, which was originally used for image semantic segmentation. The selection of this particular architecture is motivated by its potentially promising performance in learning the inherent signatures of appliances as demonstrated in [43]. The PSPNet model was further adjusted for the NILM task, and Fig. 3 shows the training structure of the adjusted PSPNet.

The rest of the subsection describes the adjusted PSPNet model, which consists of three modules: the encoder, the temporal pooling module, and the decoder.

- **Encoder.** The input of the encoder is the household aggregated load consumption of the target appliances over a 1-h interval (the consumption datasets were resampled to 30 s). The encoder is made up of four modules, each of which is alternated by a max pool layer except for the last block. The encoder increases the output features from a single aggregation value to 256, while paying the price of decreasing the time signal resolution by 10 times.
- **Temporal Pooling.** The temporal pooling consists of four average pooling modules, filter sizes of which are decreased from the whole size of the input signal to one-sixth of it. After going through a convolutional layer, the feature dimension of the input is reduced to a quarter of its original size, and the acquired feature maps are upsampled to the size of the input time signals. Then the upsampled feature maps (shallow features) are concatenated with the original input signal (deep features) from the temporal pooling to get the final feature maps. The fusion of the deep and shallow features of the temporal pool could enable this block to get contextual information fed into the decoder.
- **Decoder.** The decoder receives the output from the temporal pooling block and passes it to a convolutional layer to recover the time signal resolution. Then the output is fed into the final convolutional layer to produce the final appliance-level load disaggregation.

5. Utility optimization of DP^2-NILM

Recall the FL optimization objective (Eq. (7)), the DP^2-NILM framework considers two utility optimization schemes, the FedAvg-NILM and the FedProx-NILM, to achieve this goal.

5.1. FedAvg-NILM

Algorithm 2 depicts the steps of FedAvg-NILM. The FedAvg [31] allows the smart meter clients to train their local DNN models iteratively using the same learning rate and the number of epochs before uploading the updated model weights to the central server.

For each global round (line 4), every smart meter client receives a copy of the global model and trains its local DNN models with its own private smart meter readings for multiple epochs using $w^n \leftarrow$

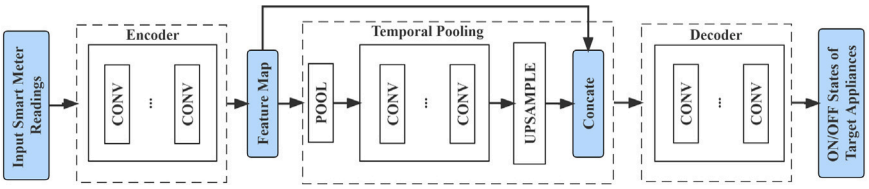


Fig. 3. The overall layout of the deep learning model for NILM.

Algorithm 2: FedAvg-NILM

1 **Input:** Aggregated load consumption of target appliances from all N houses $\{L^n | n \in N\}$, the number of global communication rounds R, the local batch B_L, the number of local epochs E.
Output: The optimal global deep learning model parameters.
2 **Central Server Execution:**
3 Initialize the global model parameters w_g
4 **for** *each global round* $r \leq R$ **do**
5 **for** *each client* $n \in \{1, 2, \ldots, N\}$ ***in parallel*** **do**
6 $w^n_{r+1} \leftarrow HouseholdsUpdate(w^n_r)$
7 **end**
8 $w_{r+1} \leftarrow \frac{\sum_{n=1}^{N} w^n_r}{N}$
9 **end**
10 Broadcast the global model to all clients
11 **Smart Meter Client Execution:**
12 **procedure** $HouseholdsUpdate(w^n_r)$:
13 Split L_n into batches of size B_L;
14 **for** *each local client epoch* $e \leq E$ **do**
15 **for** *each batch of* L_n **do**
16 $w^n \leftarrow w^n - \eta \nabla \mathcal{L}(w^n)$
17 **end**
18 **end**
19 Upload w^n to the central server

$w^n - \eta \nabla \mathcal{L}(w^n)$ (line 14–18), where η is the learning rate. After this, the local clients upload their updated local model weights w^n to the central server (line 19). Then, the central server updates the global model by averaging the uploaded weights from the smart meter clients (line 8) and broadcasts the updated global model to all clients (line 10).

An advantage of FedAvg-NILM is that a well-trained FedAvg-NILM model can outperform a single local NILM model while maintaining data privacy. Moreover, FedAvg has been proven to be efficient in reducing the communication overhead between the local clients and the global server [31].

Nevertheless, the FedAvg only performs effectively under the premise that all the local clients utilize a similar initialization, and it has been shown that heterogeneity of data impedes the convergence of FedAvg [37]. In the real-world NILM tasks, smart meter clients often exhibit diverse appliance usage patterns, making the local client models easily deviate from the global model, thereby reducing the overall performance.

5.2. FedProx-NILM

It is likely that data from smart meters are heterogeneous since they are collected under various contexts (e.g., across different countries) and are affected by diverse client behaviors leading to heterogeneous load usage distributions. Our DP^2-NILM framework is efficient for guaranteeing the convergence of the FL model in heterogeneity settings, i.e., the non-IID data settings, by incorporating FedProx [37] as an extension of the utility optimization scheme.

Algorithm 3 depicts the steps of FedProx-NILM. The central server executes the same steps as in the FedAvg-NILM. However, a proximal term $\mu(w^n - w^n_r)$ is added to update the local model of smart meter clients (line 10), which keeps local updates from deviating too much from the initial global model. When $\mu = 0$, the FedProx-NILM will produce the same results as the FedAvg-NILM.

Algorithm 3: FedProx$_{mod}$-NILM

1 **Input:** Aggregated load consumption of target appliances from all N houses $\{L^n | n \in N\}$, the number of global communication rounds R, the local batch B_L, the number of local epochs E.
Output: The optimal global deep learning model parameters.
2 **Central Server Execution:**
3 (// Same central server execution steps as the FedAvg-NILM)
4 Broadcast the global model to all clients
5 **Smart Meter Client Execution:**
6 **procedure** $HouseholdsUpdate(w^n_r)$:
7 Split L_n into batches of size B_L
8 **for** *each local client epoch* $e \leq E$ **do**
9 **for** *each batch of* L_n **do**
10 $\nabla \mathcal{L}_{prox}(w^n) \leftarrow \nabla \mathcal{L}(w^n) + \mu(w^n - w^n_r)$
11 $w^n \leftarrow w^n - \eta \nabla \mathcal{L}_{prox}(w^n)$
12 **end**
13 **end**
14 Upload w^n to the central server

Specifically, in the typical FedProx training paradigm, there is an inexact minimizer adjusting the local epoch of each client to reduce the negative impact of the system heterogeneity, which is defined as follows.

Definition 2 (γ-*Inexact Solution [37]*)**.** The w^* is a γ-inexact minimizer solution for the optimization objective in Eq. (7) if $\left\|w^* - w^n_r\right\| \leq \gamma \left\|w^n_r - w^n_{r-1}\right\|$, where $\gamma \in [0, 1)$.

The γ-inexact minimizer solution considers adjusting the local computation and the global communication overhead based on the number of local model epochs performed by the clients. Our framework hypothesizes that most smart meter clients are available and capable of completing a certain number of local epochs whereas for the very few stragglers, their destabilized training environment may produce models that contribute little to the FL global model. Therefore, FedProx was adjusted to be more efficient in the DP^2-NILM framework by utilizing the proximal term $\mu(w^n - w^n_r)$ with the exact minimizer solution w^n_r rather than the inexact one.

6. Privacy-preserving of DP^2-NILM

The DP^2-NILM considers privacy-preserving mechanisms at two different levels to suit various privacy requirements from smart meter clients, i.e., the global differential privacy federated learning and the local differential privacy federated learning. In the practical privacy-preserving mechanisms of the DP^2-NILM, recall the (ϵ, δ)-DP defined in Eqs. (4) to (6), the query function $\mathcal{F}$ represents the aggregated main readings of a household, and $\Delta\mathcal{F}$ denotes the maximum electricity consumption of any household. The random algorithm $\mathcal{M}$ injects the noise $\mathcal{N}(0, \Delta\mathcal{F}^2 \cdot \sigma^2)$ into the aggregated weights of all the N households to provide global differential privacy to the federated learning NILM or into the updated weights of each local household to provide local differential privacy to the federated learning NILM.

6.1. Global differential privacy federated learning NILM

In the DP^2-NILM paradigm, if a client sends out the privacy requirement and meanwhile trusts the central server, the GDPFL-NILM will be

utilized for this client. The GDPFL-NILM is designed to accommodate the needs of smart meter clients who are not concerned about their data, but are concerned about identity leakage. Although there must be a certain degree of trust in the central server, this presumption is significantly less stringent than granting the server access to the data. Algorithm 4 details the GDPFL-NILM scheme in the DP^2-NILM.

Algorithm 4: GDPFL-NILM

1 **Input:** Aggregated load consumption of target appliances from all N houses $\{L^n | n \in N\}$, the number of global communication rounds R, the local batch B_L, the number of local epochs E, privacy budget ϵ, privacy relaxation term δ.
Output: The optimal global deep learning model parameters with GDP protection.
2 **Central Server Execution:**
3 Initialize the global model parameters w_g
4 **for** *each global round* $r \leq R$ **do**
5 Compute privacy cost: $\hat{\epsilon}_r \leftarrow PrivacyAccount(\delta, \sigma)$;
6 **if** $\hat{\epsilon}_r > \epsilon_r$ **then**
7 **return** w_r
8 **end**
9 **else**
10 **for** *each client* $n \in \{1, 2, \ldots, N\}$ ***in parallel*** **do**
11 $w^n_{r+1} \leftarrow HouseholdsUpdate(w^n_r)$
12 **end**
13 $w_{r+1} \leftarrow \frac{\sum_{n=1}^{N} w^n_r}{N} + \mathcal{N}(0, \Delta F^2 \cdot \sigma^2)$
14 **end**
15 **end**
16 Broadcast the global model to all clients
17 **Smart Meter Client Execution:**
18 (// Same smart meter client execution steps as the FedAvg-NILM)
19 Upload w^n to the central server

In GDPFL-NILM, the smart meter client execution steps are the same as in FedAvg-NILM. The central server guarantees participant-level privacy by perturbing the model weights aggregation, i.e., adding Gaussian noise $\mathcal{N}(0, \Delta F^2 \cdot \sigma^2)$ to the aggregated results (line 13). Moreover, to ensure the (ϵ, δ)-GDP, after each global round, the algorithm $PrivacyAccount()$ calculates the accumulated privacy budget (line 5), and if it exceeds the overall budget ϵ, the global training iteration will be stopped (line 7). In particular, the privacy cost is associated with the Gaussian noise added to the updated weights, which can be calculated by numerical integration as described in [10]. The global training generally involves gradients at multiple layers, and the accountant accumulates the privacy cost associated with each of them.

6.2. Local differential privacy federated learning NILM

In the LDPFL [10], smart meter clients apply noise on the updated local model weights before uploading them to the central server. The LDPFL-NILM is designed to cater to the needs of smart meter clients who are concerned about local data leakage. The LDPFL-NILM scheme in DP^2-NILM is presented in Algorithm 5.

The central server updating process in the LDPFL-NILM is the same as in FedAvg-NILM. However, the smart meter clients guarantee their own privacy by perturbing the updated local model weights, i.e., adding Gaussian noise $\mathcal{N}(0, \frac{\Delta F^2 \cdot \sigma^2}{N})$ to the updated model weights (line 10). The LDPFL-NILM provides better privacy protection than the GDPFL-NILM, and therefore it is suitable for clients who require strict data privacy-preserving discipline.

7. Performance evaluation

In this section, real-world smart meter datasets are used to evaluate the proposed DP^2-NILM framework. The datasets and the evaluation criteria are first introduced. Then, the performance of the FL setting in DP^2-NILM is compared with the Local-NILM models trained on individual household datasets and the Centralized-NILM model trained on aggregated household datasets. After this, the utility optimization schemes are examined in the DP^2-NILM paradigm. Finally, based on the FedAvg, two privacy-preserving schemes, i.e., the GDPFL-NILM and the LDPFL-NILM, are compared in terms of the trade-off between model utility and privacy.

Algorithm 5: LDPFL-NILM

1 **Input:** Aggregated load consumption of target appliances from all N houses $\{L^n | n \in N\}$, the number of global communication rounds R, the local batch B_L, the number of local epochs E, privacy budget ϵ, privacy relaxation term δ.
Output: The optimal global deep learning model parameters with LDP protection.
2 **Central Server Execution:**
3 (// Same central server execution steps as the FedAvg-NILM)
4 Broadcast the global model to all clients
5 **Smart Meter Client Execution:**
6 **procedure** $HouseholdsUpdate(w^n_r)$:
7 Split L_n into batches of size B_L
8 **for** *each local client epoch* $e \leq E$ **do**
9 **for** *each batch of* L_n **do**
10 $\nabla \mathcal{L}_{ldp}(w^n) \leftarrow \nabla \mathcal{L}(w^n) + \mathcal{N}(0, \frac{\Delta F^2 \cdot \sigma^2}{N})$
11 $w^n \leftarrow w^n - \eta \nabla \mathcal{L}_{ldp}(w^n)$
12 **end**
13 **end**
14 Upload w^n to the central server

7.1. Experimental settings

This study used three real-world smart meter datasets to evaluate the DP^2-NILM framework, including UKDALE [44], REDD [45] and REFIT [46].

- UKDALE: The U.K. domestic appliance level electricity (UKDALE) dataset contains five buildings in the U.K. between 2013 and 2015 with a 1 s sampling period for mains and a 6 s sampling period for appliances.
- REDD: The reference energy disaggregation dataset (REDD) consists of six buildings in the U.S. spanning from 3 to 19 days, with a 1 s sampling period for mains and a 6 s sampling period for appliances.
- REFIT: The REFIT dataset contains 20 buildings in the U.K. from 2013 to 2015, sampled at 8 s for both mains and appliances.

Three appliances (fridge, dishwasher, washing machine) are selected as our target appliances for comparison purposes, which are the most common appliances possessed by most households among the three datasets and represent both two-state and multi-state appliances. Labeled data for all three appliances are available in UKDALE houses 1, 2, and 5, REDD houses 1, 2, and 3, and REFIT houses 2, 5, and 9, therefore only these households were considered.

Fig. 4 gives an example distribution of the three chosen appliances for house 1 of the REDD dataset. It can be seen from the distribution that the multi-state appliances including the dishwasher and washing machine have a high power consumption spike when turned on, and the two-state appliance fridge has relatively lower power consumption during the whole monitoring period. This may partly reflect that the demand of the smart meter clients for the fridge is largely constant, while both the dishwasher and washing machine are only used occasionally. Moreover, there are many small power spikes for the fridge, but relatively few for the dishwasher and washing machine. This distribution occurs primarily because the fridge undergoes more frequent ON-OFF cycles than the dishwasher and washing machine during the day.

As the three datasets are sourced from different data repositories, they have data heterogeneity in terms of sample periods and data magnitude. The original sample frequency for the appliances is 6 or 8 s, which contains more spikes of appliance consumption, it undermines

Table 1
Distribution of the selected datasets.

Dataset	Building	Total period	Training (80%)	Validation (10%)	Testing (10%)
UKDALE	1	2013-04-12 to 2017-04-25	2013-04-12 to 2016-07-04	2016-07-05 to 2016-11-29	2016-11-29 to 2017-04-25
	2	2013-05-22 to 2013-10-30	2013-05-22 to 2013-09-28	2013-09-28 to 2013-10-14	2013-10-14 to 2013-10-30
	5	2014-06-29 to 2014-09-01	2014-06-29 to 2014-08-19	2014-08-19 to 2014-08-25	2014-08-25 to 2014-09-01
REDD	1	2011-04-19 to 2011-05-19	2011-04-19 to 2011-05-12	2011-05-13 to 2011-05-15	2011-05-16 to 2011-05-19
	2	2011-04-18 to 2011-05-21	2011-04-18 to 2011-05-14	2011-05-14 to 2011-05-17	2011-05-17 to 2011-05-21
	3	2011-04-17 to 2011-05-30	2011-04-17 to 2011-05-21	2011-05-21 to 2011-05-25	2011-05-25 to 2011-05-30
REFIT	2	2013-09-18 to 2015-05-27	2013-09-18 to 2015-01-23	2015-01-23 to 2015-03-26	2015-03-26 to 2015-05-27
	5	2013-09-27 to 2015-07-05	2013-09-27 to 2015-02-25	2015-02-25 to 2015-05-01	2015-05-01 to 2015-07-05
	9	2013-12-18 to 2015-07-07	2013-12-18 to 2015-03-15	2015-03-15 to 2015-05-11	2015-05-11 to 2015-07-07

Table 2
Relevant threshold information.

	Fridge	Dishwasher	Washing Machine
Max power (W)	300	2500	2500
Power threshold λ^i (W)	50	20	20
Min. ON duration d_1 (s)	1	60	60
Min. OFF duration d_0 (s)	0	60	5

the training accuracy and makes the process more computationally intensive.

On the other hand, sub-sampling smart meter data is a common processing technique that has been adopted in many previous studies for NILM [43,47–50], since it is useful for improving computational and memory efficiency. Besides, by setting an appropriate sampling frequency, the smart meter can efficiently capture essential data without requiring extra hardware or energy-consuming devices for additional metering [48]. This allows for optimal data collection and analysis, enabling better energy management and insights without unnecessary costs or complexity.

It is essential to experiment with different granularity of the smart meter data to facilitate an informed decision. Therefore, models for different sample frequencies, 8 s and 30 s, are considered in the following Section 7.2 to assist the choice of the data granularity by considering the trade-off between temporal resolution and computational efficiency.

This study splits the 8 s and 30 s resolution data points into training, validation, and testing datasets while preserving consecutive intervals. The dataset at 8 s intervals contains 3.75 times more data points than the dataset at 30 s intervals, potentially leading to a substantial rise in computational costs. The distributions of the selected buildings are listed in Table 1. Specifically, 80% records from each smart meter client were selected as the training set, followed by 10% for validation, and 10% for testing. The training-testing segment setting is applied to all the schemes in the proposed DP^2-NILM. Then, to overcome the heterogeneity of data magnitude, all the data were normalized in the same data range before training.

Table 2 gives the relevant thresholds used in data preprocessing based on empirical analysis of appliance behavior in [3], which are commonly used in many studies [43,51,52]. The abnormal load consumption was firstly filtered out by the max power [3], and then the load consumption of all the nine households was down-sampled from 6 s to 30 s through averaging. After this, the resampled data were normalized by subtracting the mean and dividing a constant load value 2000 W following [3]. Then the state series of each target appliance was derived from activation-time thresholding described in Algorithm 1 as the input to feed into the DP^2-NILM.

Table 3
Parameters used in the DP^2-NILM framework.

Parameters	Value
Batch size	32
Global rounds	10
Local epochs	8
Number of clients	9
Proximal parameter μ	0.01
Privacy budget ϵ	[4, 8, 12]
Privacy relaxation term δ	10^{-5}
Gradient clipping threshold	4
Learning rate η	10^{-4}
Activation function	ReLU
Dropout probability	0.1
Momentum	0.5
Optimizer	SGD

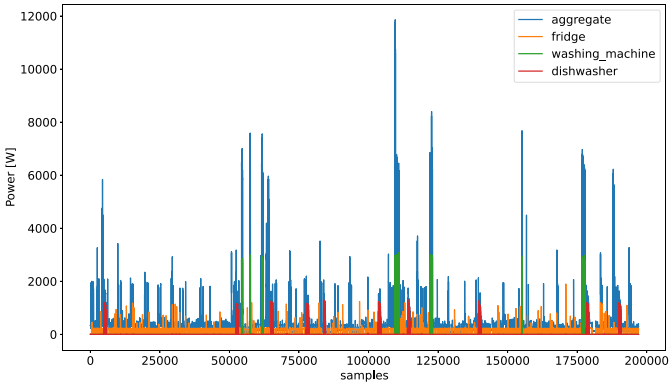


Fig. 4. The distribution of the three appliances during the monitoring period for house 1 of the REDD dataset.

Furthermore, parameters used in the DP^2-NILM framework are listed in Table 3. TensorFlow is used to train the DP^2-NILM framework , and it is particularly challenging to reproduce exact results due to the inherent variance of deep learning algorithms. The experiments were repeated five times on various datasets for replicability, and the final average scores of each model were reported. Furthermore, the standard deviation of the evaluation scores for the model performance from five runs is provided as a means to gain insights into the stability and consistency of each model [53].

To keep the comparison fair, all the models in our experiment use the same DNN architecture described in Fig. 3. Appropriate selection of the local training epochs has been proven to be effective in accelerating

the model convergence [31]. To simulate real-world environmental conditions, the local epochs for each client can also be set differently depending on the network/hardware environment, and the same local epoch setting is used here for the convenience of comparison. For all the FL models in DP^2-NILM, each global training round consists of eight local epochs, allowing the clients to take reasonable learning steps before central server aggregation. For the privacy-preserving scheme, the study varies the privacy budget ϵ between 4 and 12 while keeping $\delta = 10^{-5}$, and report the performance and attack success risk, i.e., the Accuracy and the ASR. The choice of $\delta = 10^{-5}$ satisfies the requirement that δ should be smaller than the inverse of the training data size [10]. To bound the sensitivity ΔF^2 of the gradients, clipping is required, which is a computationally efficient and common practice in deep learning. With the TensorFlow Privacy framework, the batch clipping was implemented with a threshold of 4. Further, for the listed parameters, it should be noted that an additional parameter tuning step may improve the final model performance, however at the cost of massive computational resources.

Four evaluation metrics are used to assess the model performance of the DP^2-NILM framework. Denote true positive as TP, true negative as TN, false positive as FP, and false negative as FN, the evaluation metrics can be defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

where the precision represents the proportion of TPs to all the data sequences classified to the ON state. Recall denotes the ratio of TPs to all data sequences that are actually in the ON state. Accuracy reflects the ratio of all correctly identified samples to all the smart meter data sequences. F_1 is defined as a weight average representation for the precision and the recall within the range of $[0, 1]$. An F_1 close to 1 indicates that the classification results for the target appliances are better.

Moreover, to measure the privacy risk for the privacy-preserving DP^2-NILM models, the member inference attack metric defined in [54] was utilized. To test the membership of an input record, this attack mechanism evaluates the loss of the uploaded local model parameters and then classifies it as a member if the loss is smaller than the average training loss. The attack success risk can be calculated as

$$ASR = TPR - FPR, \tag{12}$$

where $TPR = \frac{TP}{TP+FN}$ denotes the TP rate, and $FPR = \frac{FP}{FP+TN}$ represents the FP rate.

7.2. Evaluation on the baseline model of DP^2-NILM

This subsection evaluates the FL setting of DP^2-NILM. There are three different model settings in this subsection: (1) Local-NILM models: The Local-NILM models are trained on nine household datasets separately. This setting eliminates the need for data sharing with the central server, but at the expense of having to update all of the nine models separately; (2) Centralized-NILM model: The Centralized-NILM model is trained on aggregated datasets from all the nine households, which requires raw data sharing from the smart meter clients; (3) FL-setting of DP^2-NILM (FedAvg-NILM): The FL-setting of DP^2-NILM utilizes FedAvg as the optimization method and trained on all the nine households without any exchange of the raw smart meter data. The FL model trained based on FedAvg in DP^2-NILM will be used later as the baseline for evaluating two schemes in the proposed framework. Moreover, the thorough investigation of different granularity levels within the smart meter data is of utmost significance to facilitate an informed decision in the proposed DP^2-NILM. Accordingly, this subsection also examines models designed for distinct sample frequencies—specifically, 8 s and 30 s. The aim is to support the selection of the most suitable data granularity by assessing the compromise between temporal resolution and computational efficiency.

For the Local-NILM models and the Centralized-NILM model, the epochs are set to 80 to achieve the final convergence. For comparison purposes, Table 4 lists the mean and standard deviation of performance scores of the Local-NILM models, the Centralized-NILM model, and the FL-setting of DP^2-NILM (FedAvg-NILM) for the nine households based on 8 s and 30 s temporal resolutions over five runs. Moreover, Centralized-NILM and FedAvg-NILM aim to train a central or global model that captures appliance patterns from all nine clients. The variable std was introduced to denote the standard deviation of scores acquired from five individual runs during the testing period, capturing the variation of scores across these runs. Concurrently, the Local-NILM aims to train models on individual devices, with its std of scores calculated from five runs based on the averaged scores across the nine households. To quantify the variation in scores among the different households in Local-NILM, std_H was incorporated to represent the standard deviation of scores derived from these nine households. In std_H, scores are averaged across five runs within each household before calculating the standard deviation.

A comparison of FedAvg-NILM with Local-NILM models examines the performance of federated learning strategies in capturing diversities among clients. Moreover, comparing FedAvg-NILM to the Centralized-NILM model evaluates the overall performance of the common utility FL model. The results showed that for each appliance, all models achieved satisfactory results on the dishwasher and washing machine, and reasonable results on the fridge. Note that the FedAvg-NILM achieved the same accuracy score and higher F_1, precision, and recall scores on dishwashers and washing machines compared with the centralized-NILM model. For the fridge, as it consumes relatively low power compared with other appliances, it is likely to be learned with less evident signature during model training and such consumption can easily be omitted as unidentified load noise in the FL paradigm. Furthermore, the standard deviation std of each score over five runs for the three models in Table 4 is very small compared to the scale of the mean value. Overall, it can be concluded that the FedAvg-NILM model in the DP^2-NILM framework works well, and its performance can serve as the baseline for further evaluations.

Moreover, it is worth noting that the std_H of the Local-NILM models are relatively higher compared with the std. The reason for this may be that the performance scores of the models are the averaged scores of all nine households for the five runs, which may smooth the negative impact of outliers in the individual households, thus leading to a smaller standard deviation std than the std_H. On the other hand, the larger value of std_H implies that there is more variability in scores among different households. In essence, the higher standard deviations observed in the scores from nine households, where each score is the average of five runs for models, provide evidence of variations among different households.

It is also observed that with more global rounds, the FedAvg-NILM may achieve more satisfying performances. For example, the study has set 100 global rounds for the FedAvg-NILM, and the final obtained average accuracy for the fridge, dishwasher, and washing machine were 0.89, 0.99, and 0.99, respectively, which are even better than the centralized NILM model. However, parameter tuning in FL remains a challenge as a result of the distributed environment and the associated computational overhead [8]. It is believed that fixed global and local training rounds enable efficient, fair, and comparable evaluations in our DP^2-NILM framework.

Overall, models trained with 8 s resolution data exhibited improved performance across most evaluation metrics for the fridge. This enhancement in performance can be ascribed to the inherent

Table 4
Performance scores (mean and standard deviation) of the Local-NILM models, the Centralized-NILM model, and the FL-setting of DP^2-NILM for nine households based on different temporal resolutions over five runs. *std*: standard deviation of scores across five runs on the testing period. std_H: standard deviation of scores among the nine different households.

			Fridge				Dishwasher				Washing Machine			
			Accuracy	F_1	Precision	Recall	Accuracy	F_1	Precision	Recall	Accuracy	F_1	Precision	Recall
Local-NILM	8 s	Mean	0.84	0.81	0.81	0.80	0.97	0.66	0.81	0.65	0.98	0.64	0.81	0.55
		$std_R(\times10^{-3})$	5.45	4.44	4.59	3.21	6.11	3.76	4.63	5.90	5.98	5.73	4.43	3.44
		$std_H(\times10^{-2})$	2.60	2.62	2.83	3.38	3.09	3.95	2.49	3.40	3.99	2.60	3.39	3.10
	30 s	Mean	0.83	0.79	0.82	0.74	0.98	0.88	0.86	0.83	0.99	0.78	0.89	0.70
		$std_R(\times10^{-3})$	6.37	9.17	8.72	9.32	6.13	5.68	9.84	5.47	7.55	5.81	8.30	7.73
		$std_H(\times10^{-2})$	2.08	2.25	2.69	2.53	2.20	2.52	4.06	3.22	2.28	4.20	2.69	3.75
Centralized-NILM	8 s	Mean	0.90	0.82	0.81	0.80	0.97	0.69	0.85	0.57	0.96	0.62	0.68	0.58
		$std_R(\times10^{-3})$	1.94	3.32	6.06	5.01	5.41	4.49	4.69	3.10	4.61	1.76	3.34	2.79
	30 s	Mean	0.86	0.80	0.79	0.81	0.97	0.70	0.87	0.59	0.97	0.66	0.71	0.62
		$std_R(\times10^{-3})$	4.73	5.38	9.45	4.38	6.51	4.14	3.48	4.25	9.25	2.99	4.47	5.68
FedAvg-NILM	8 s	Mean	0.87	0.83	0.84	0.81	0.95	0.79	0.77	0.83	0.97	0.51	0.83	0.40
		$std_R(\times10^{-3})$	6.05	5.22	3.74	4.72	3.75	4.53	3.26	3.20	4.36	3.73	6.20	2.97
	30 s	Mean	0.65	0.63	0.50	0.85	0.97	0.75	0.92	0.64	0.98	0.71	0.83	0.62
		$std_R(\times10^{-3})$	2.80	3.86	3.02	3.80	1.78	2.71	4.22	3.25	5.38	3.37	3.94	3.64

characteristics of fridges, which tend to exhibit a relatively stable and consistent power consumption pattern over a short duration. By employing higher-resolution data, the models become capable of capturing more finely detailed fluctuations in the power consumption of the fridge. Consequently, this enables the model to more effectively differentiate the unique patterns associated with the fridge from those of other household appliances. However, for dishwashers and washing machines, the utilization of 8 s resolution data leads to compromised performance compared to 30 s resolution data. This counter-intuitive outcome may be attributed to the dynamic power consumption patterns and rapid fluctuations exhibited by these appliances during their operational cycles, which result in increased noise in the 8 s resolution data, challenging the accurate separation of appliance-specific patterns from noise, thus leading to inferior results.

In the context of DP^2-NILM, it is important to consider the practical trade-off between data resolution and computational efficiency. While higher-frequency data can offer more detailed insights, it comes at the cost of increased resource requirements for processing and model training. This study strikes a balance between these conflicting considerations by adopting a methodology akin to that employed in prior studies [43,47–50]. The datasets were sub-sampled to 30 s intervals, ensuring both feasibility across a wide range of smart meters and consistency with the experimental design scenarios pursued in our investigation.

7.3. Evaluations on utility optimization of DP^2-NILM

The performance of FedAvg-NILM regarding the nine smart meter clients is satisfactory for the dishwasher and washing machine. However, its performance on the fridge is worse compared to both the Local-NILM and Centralize-NILM models. The study further conjecture that the load consumption distribution of the fridge for the clients may be heterogeneous because they are collected geographically, i.e., the REDD dataset is from the U.S., whereas the other two datasets are from the U.K., and the size of the smart meter records from REDD are smaller than the other two datasets. Fig. 5 shows the load consumption distribution of the fridge for all nine clients.

It can be seen that different clients have different fridge usage patterns, with UKDALE house 1, REDD house 3, and REFIT houses 5 and 9 having more operation spikes, while UKDALE house 5, REDD house 2, and REFIT house 2 have more steady consumption patterns.

It is hypothesized that using optimization algorithms that can accommodate statistical heterogeneity may be useful for improving the performance of FL models. This subsection will explore the relationship between data heterogeneity and the different types of FL utility optimization models. By comparing the FedProx-NILM to the FedAvg-NILM, the ability of the two strategies to learn from heterogeneous data in the DP^2-NILM framework is evaluated. Table 5 lists the average performance scores of the FedAvg-NILM and the FedProx-NILM, and highlights the improvement in blue and the downgrade in red of the FedProx-NILM corresponding to the FedAvg-NILM.

The standard deviation of each score over five runs in Table 5 for the FedProx-NILM is remarkably small compared to the scale of the mean value. It can be observed that FedProx-NILM significantly outperforms FedAvg for most scores on fridge and dishwasher, especially on the fridge with improved accuracy by 20% and an increased precision by 32%. The fridge has an extremely short activation cycle when compared to the other two appliances, and it consumes less electricity than dishwashers and washing machines, which makes the activation cycle easily obscured by unobserved noise.

Moreover, the use of a fridge in a household does not follow a daily routine like dishwashers and washing machines, its activation cycle is more irregular (e.g., UKDALE house 1, REDD house 3, and REFIT houses 5 and 9), and the usage patterns from multiple households may contain higher levels of randomness. Therefore, training the FedAvg-NILM model with the simple averaging algorithm may result in a degradation of the global learning model, thus leading to relatively worse performance scores. Besides, FedProx-NILM achieved a satisfactory precision for the fridge, which means 32% fewer cases of the fridge status being falsely identified as ON than FedAvg-NILM.

In the practical application scenarios, the precision implies that the appliance status was falsely identified as ON, while the recall implies that the appliance status was falsely identified as OFF before it was actually turned off. It is worth noting that the FedProx-NILM achieved better recall and a worse precision score for the dishwasher. This is because the FedProx-NILM system is specifically designed to handle the heterogeneity from different households by introducing the proximal term, and thus to keep the model parameters updated by the local client not deviating too much from the global model parameters, it can be inferred that the FedProx-NILM can detect longer operation duration with lower electricity consumption such as the draining of the dishwasher and the typical standby state of the fridge.

There are a slightly drop (1%) on accuracy and a significant drop (22%) on recall of the washing machine. It is inferred that, as the signatures of the washing machines are more complex than the other two appliances [11], the proximal term in FedProx-NILM reduces the

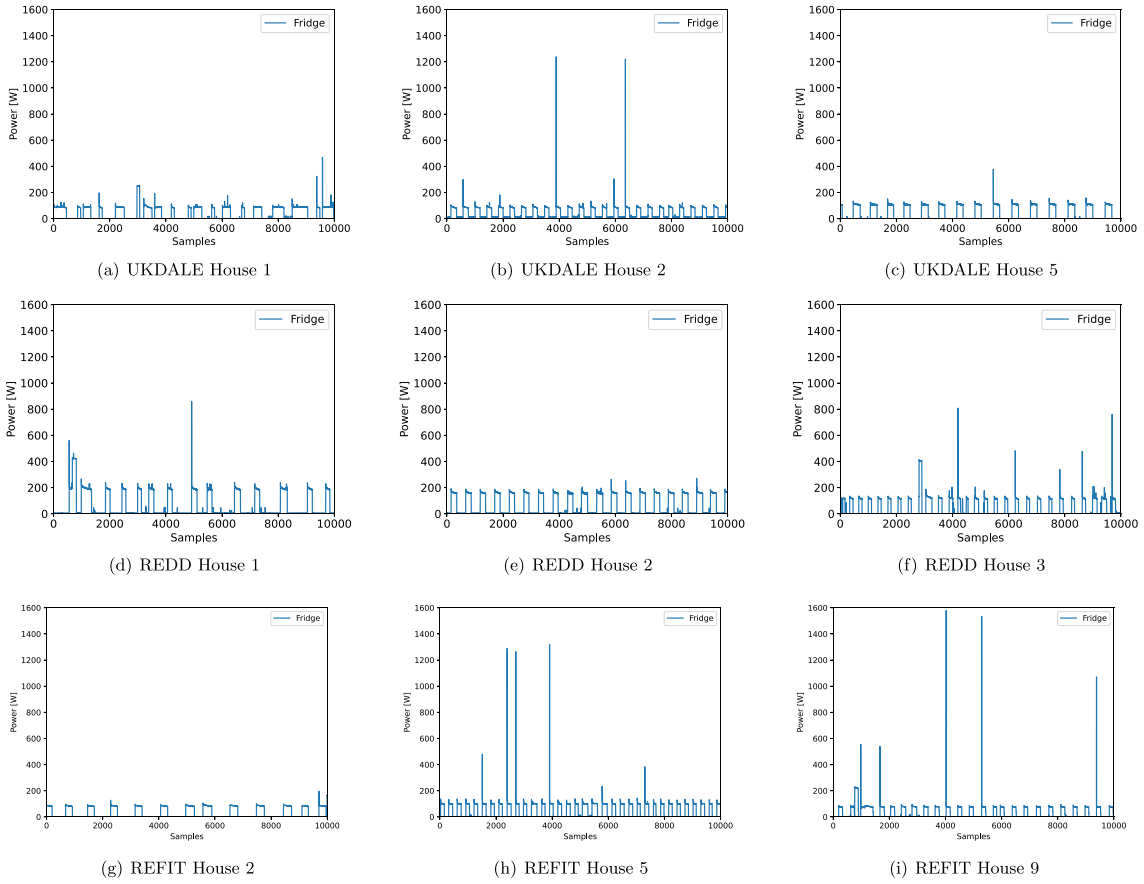


Fig. 5. Example fridge usage distributions for UKDALE, REDD, and REFIT.

Table 5
Performance scores (mean and standard deviation) of FedAvg-NILM and FedProx-NILM schemes for nine households over five runs.

		Fridge				Dishwasher				Washing Machine			
		Accuracy	F_1	Precision	Recall	Accuracy	F_1	Precision	Recall	Accuracy	F_1	Precision	Recall
FedAvg-NILM	Mean	0.65	0.63	0.50	0.85	0.97	0.75	0.92	0.64	0.98	0.71	0.83	0.62
	$std(\times 10^{-3})$	2.80	3.86	3.02	3.80	1.78	2.71	4.22	3.25	5.38	3.37	3.94	3.64
FedProx-NILM	Mean	0.85	0.81	0.82	0.81	0.98	0.80	0.78	0.82	0.97	0.54	0.83	0.40
	$std(\times 10^{-3})$	2.58	5.71	2.65	3.64	4.34	3.50	3.42	3.41	7.49	3.25	2.77	1.63
Evaluation		(↑ 20%)	(↑ 18%)	(↑ 32%)	(↓ 4%)	(↑ 1%)	(↑ 5%)	(↓ 19%)	(↑ 18%)	(↓ 1%)	(↓ 17%)	(–)	(↓ 22%)

difference in weight updates for individual models, which may undermine the learning of significant features of washing machines by the client models. Therefore, regarding the washing machine results, FedAvg-NILM is more efficient in identifying high-energy appliances similar to washing machines.

In terms of the F_1 score, the FedProx-NILM outperformed the FedAvg-NILM for fridges and dishwashers but had a worse F_1 score for washing machines than the FedAvg-NILM. To conclude, the results further confirm our assumptions regarding the utility optimization based on FedProx-NILM for handling heterogeneous smart meter appliances, and it can also be concluded that FedAvg-NILM performs better in detecting intensive operation duration with higher electricity consumption.

The insights gained from analyzing the performance of FedProx-NILM and FedAvg-NILM models can provide guidance for the development of more efficient and reliable appliance detection systems. The observed variations in model performance for different appliances underscore the importance of implementing appliance-specific policies or standards. This, in turn, can drive the development of more tailored energy efficiency strategies to optimize appliance-level energy consumption.

7.4. Evaluations on privacy-preserving of DP^2-NILM

FedAvg-NILM and FedProx-NILM presented unique advantages for devices with different signatures and datasets with different degrees of consistency. However, studies are suggesting that potential risks still exist in the training communication process even though the transmitted objects are the updated parameters instead of the original data [55]. Therefore, it is necessary to provide stronger privacy guarantees to the FL-based NILM.

This subsection evaluates two privacy-preserving schemes of DP^2-NILM, i.e., the GDPFL-NILM and the LDPFL-NILM. When clients decide whether to participate in the DP^2-NILM paradigm for smart meter data analysis, our framework serves as a reference for quantifying the potential privacy loss based on the privacy budget ϵ. By comparing the benefits of participating in the framework, clients can make an informed decision on whether to join.

Table 6
Performance scores (mean and standard deviation) of the GDPFL-NILM and the LDPFL-NILM schemes for nine households over five runs.

	Privacy budget ϵ		Fridge				Dish Washer				Washing Machine				Privacy guarantee	Trusted server
			Accuracy	F_1	Precision	Recall	Accuracy	F_1	Precision	Recall	Accuracy	F_1	Precision	Recall		
FedAvg-NILM	\	Mean	0.65	0.63	0.50	0.85	0.97	0.75	0.92	0.64	0.98	0.71	0.83	0.62	Basic	Yes
		$std(\times 10^{-3})$	2.80	3.86	3.02	3.80	1.78	2.71	4.22	3.25	5.38	3.37	3.94	3.64		
GDPFL-NILM	4	Mean	0.54	0.53	0.37	0.95	0.90	0.14	0.16	0.72	0.95	0.68	0.40	0.92	Moderate	Yes
		$std(\times 10^{-3})$	3.53	2.49	2.61	4.13	3.19	0.45	1.08	3.39	3.52	1.74	2.76	6.99		
	8	Mean	0.63	0.61	0.49	0.84	0.97	0.69	0.93	0.56	0.98	0.68	0.79	0.60		
		$std(\times 10^{-3})$	2.17	2.70	1.83	2.65	4.55	5.42	5.07	1.69	7.44	2.43	5.91	2.92		
	12	Mean	0.66	0.82	0.81	0.83	0.99	0.85	0.86	0.85	0.98	0.74	0.80	0.63		
		$std(\times 10^{-3})$	2.06	2.69	4.57	2.16	3.75	4.65	7.73	5.05	5.72	2.68	6.63	1.84		
LDPFL-NILM	4	Mean	0.58	0.40	0.40	0.38	0.93	0.11	0.21	0.39	0.94	0.10	0.11	0.34	Strong	No
		$std(\times 10^{-3})$	3.78	2.24	3.27	1.94	4.99	0.74	1.16	2.79	6.30	0.64	0.43	2.04		
	8	Mean	0.58	0.42	0.41	0.44	0.94	0.20	0.30	0.40	0.96	0.20	0.40	0.47		
		$std(\times 10^{-3})$	2.51	1.57	1.34	1.81	4.51	0.76	2.01	1.82	2.01	0.62	1.71	3.04		
	12	Mean	0.65	0.42	0.36	0.50	0.94	0.13	0.26	0.48	0.96	0.43	0.40	0.50		
		$std(\times 10^{-3})$	3.19	1.59	1.12	3.10	5.78	1.12	2.04	2.72	7.60	1.86	2.76	3.14		

Table 6 compares the GDPFL-NILM and the LDPFL-NILM trained with varied privacy budget ϵ, in which the FedAvg-NILM is used again as the baseline model. Intuitively, the Gaussian random noise will slow the convergence of both the GDPFL-NILM and the LDPFL-NILM models, while providing stronger privacy guarantees for the local clients, leading to trade-off problems between model utility and privacy.

The standard deviation of each score from five independent runs in Table 6, both for the LDPFL-NILM and GDPFL-NILM, exhibits an apparently small scale compared with the mean value. Overall, the GDPFL-NILM outperforms the LDPFL-NILM over most scores, which proves that more strict privacy-preserving schemes will undermine the disaggregation performance in DP^2-NILM. It can be observed from the results that the accuracy for the GDPFL-NILM and the LDPFL-NILM are comparable to or even better than that of the FedAvg-NILM. However, other scores such as F_1 and precision are much worse than those of the FedAvg-NILM, especially for the dishwasher and washing machine.

Typically, these two appliances, by nature, remain in an inactive state for extended periods, resulting in sparse activation of their ON states. This sparsity affects the balance of positive (ON status) and negative (OFF status) samples in the dataset, creating an imbalance that impacts the reliability of accuracy measurements. Consequently, while the accuracy might appear high due to the correct classification of numerous OFF states, its effectiveness in capturing the infrequent ON states, which are of greater significance, becomes the true challenge.

While accuracy quantifies the overall correctness of load monitoring, in certain specialized scenarios, as previously mentioned, recall and precision take precedence over accuracy. The balance between precision and recall depends on the specific objectives and consequences of errors in the NILM application. In contexts such as NILM-based remote health monitoring [56], where the timely detection of potential health issues or emergencies through the monitoring of daily activities (e.g., the use of electrical medical devices) is crucial for individual well-being, recall assumes paramount importance. Ensuring that all instances of relevant events (e.g., abnormal heart rhythms, seizures) are detected is crucial to providing timely medical intervention and preventing adverse health outcomes. Prioritizing recall helps minimize the risk of missing important events.

Surprisingly, the GDPFL-NILM achieved better recall scores than the FedAvg-NILM for fridges and washing machines with privacy budgets of $\epsilon = 4$, and dishwashers with privacy budgets of $\epsilon = 12$. This indicates that the GDPFL-NILM can detect the low energy consumption standby mode for the appliances, possibly because the noise added to the aggregated weights makes the trained model more robust. However, the recall scores of the LDPFL-NILM are worse in most cases, which indicates that adding noise to updated local weights may disturb the final aggregated weights, thereby affecting the final disaggregation results.

While the recall is important, precision also holds significance. In certain NILM scenarios, such as choosing appliances or customers from a pool for automated demand response in grid balancing service, the prioritization of precision is vital to minimize false positives. For the fridge, the GDPFL-NILM achieved the highest precision with the privacy budget of $\epsilon = 12$, for the dishwasher it achieved the highest precision with the privacy budget of $\epsilon = 8$, and for the washing machine, it achieved the highest precision with the privacy budget of $\epsilon = 12$.

In addition, although the GDPFL-NILM model has exhibited a decline in precision as the privacy budget has decreased, the scores are still comparable to those of the FedAvg-NILM. Moreover, the precision of the GDPFL-NILM model for the dishwasher and washing machine with the lowest privacy budget of $\epsilon = 4$ drops dramatically compared to the FedAvg-NILM. This is likely because they differ from the fridge in terms of features, as dishwashers and washing machines may offer more insight into individual behavior because they are more closely related to the routines of smart meter clients.

This study then compares the performance of the GDPFL-NILM and the LDPFL-NILM in terms of privacy attacks. To determine whether a client has participated in a training session, the attack success risk introduced in Section 7 is used as the evaluation criterion. Fig. 6 illustrates ASRs based on various epsilon budgets for FedAvg-NILM, GDPFL-NILM, and LDPFL-NILM, respectively.

All three models with three privacy budget values (i.e., $\epsilon = 4$, $\epsilon = 8$, and $\epsilon = 12$) with a fixed $\delta = 10^{-5}$ are evaluated. Fig. 6 shows that LDPFL-NILM with the setting $\epsilon = 4$ mitigates the attack success risk better (downgrades the risk to 0.33) with compromises in decreasing model accuracy by 7% for fridge, 4% for dishwasher, and 4% for washing machine. The GDPFL-NILM with $\epsilon = 8$ achieved satisfying performance on all three appliances as well as reduced the attack accuracy to 0.59.

Not surprisingly, the LDPFL-NILM imposes more noise compared to the GDPFL-NILM, which provides stronger privacy guarantees but less utility due to a higher amount of noise. It is worth noting that with a higher privacy budget $\epsilon = 12$, the attack success risk in both the GDPFL-NILM and the LDPFL-NILM is similar to that in FedAvg-NILM whereas the F_1, precision, and recall for the LDPFL-NILM are much worse than the FedAvg-NILM and the GDPFL-NILM. Therefore, it may suggest that utilizing the GDPFL-NILM or the FedAvg-NILM may achieve a better

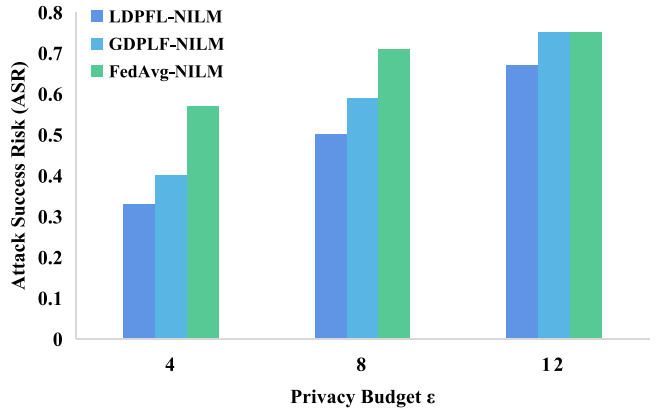


Fig. 6. The ASRs of FedAvg-NILM, the GDPFL-NILM, and the LDPFL-NILM in the DP^2-NILM framework.

trade-off between utility and privacy when there is a higher privacy budget from clients.

The superior performance of GDPFL-NILM compared to LDPFL-NILM, particularly in terms of recall, implies that industries involved in smart appliances and energy management systems can greatly benefit from adopting GDPFL-NILM. By implementing this model, they can achieve better detection of appliance usage patterns, leading to improved energy management and cost reductions. Moreover, the adoption of GDPFL-NILM enables industries to promote energy efficiency and user privacy, aligning with broader policy targets related to energy, environment, and data protection.

8. Conclusion

This research presents the DP^2-NILM framework, which combines federated learning, utility optimization, and differential privacy to provide decentralized and privacy-preserving solutions for NILM. The framework offers two key schemes: a utility optimization scheme and a privacy-preserving scheme.

The utility optimization scheme includes FedAvg-NILM and FedProx-NILM, which effectively handle data heterogeneity in different appliance types and heterogeneous environments. FedAvg-NILM excels at detecting intensive operation duration with higher electricity consumption, while FedProx-NILM is more effective at detecting longer operational duration with lower electricity consumption. The privacy-preserving scheme consists of LDPFL-NILM and GDPFL-NILM, which provide different levels of privacy guarantees based on varying privacy budgets. LDPFL-NILM offers stricter privacy rules, while GDPFL-NILM achieves an optimal trade-off between privacy and utility.

Extensive experiments conducted on real-world smart meter datasets demonstrate the scalability and effectiveness of the DP^2-NILM framework. The results highlight the importance of considering different utility optimization and privacy-preserving algorithms based on appliance types and available privacy budgets.

The relevance of this research extends to various domains. In terms of engineering design, the DP^2-NILM framework enables the development of smart energy services at the local/residential level, contributing to the decarbonization of the energy system. It also addresses privacy concerns, aligning with regulations and policies promoting data privacy and security. Moreover, the framework allows financial institutions and investors to evaluate the environmental impact and sustainability of energy consumption patterns.

Further research can explore the application of the DP^2-NILM framework in other client types, such as commercial and industrial sectors. Nevertheless, the training environment for such client types may be more complex, so it is important to further consider the system heterogeneity to ensure the robustness of the framework. Besides, the experimental results suggest that the FL paradigm can be implemented more time efficiently with reasonable increases in local epochs, while local epoch ranges that are acceptable to participants must still be verified by trial, such as questionnaires. Asynchronous schemes should also be incorporated into the proposed framework to accommodate stragglers and unfinished local iterations. Furthermore, as smart devices enable real-time feedback from smart meter clients, adapting the DP^2-NILM framework to online scenarios will deliver more flexible smart meter data analysis and improve the communication efficiency of the FL paradigm.

CRediT authorship contribution statement

Shuang Dai: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Fanlin Meng:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Qian Wang:** Conceptualization, Investigation, Writing – review & editing. **Xizhong Chen:** Conceptualization, Investigation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used are public dataset. The links to the data sources are provided as reference in the manuscript.

References

[1] Hart GW. Nonintrusive appliance load monitoring. Proc IEEE 1992;80(12):1870–91.
[2] Kelly D. Disaggregation of domestic smart meter energy data (Ph.D. dissertation), Imperial College London; 2016.
[3] Kelly J, Knottenbelt W. Neural nilm: Deep neural networks applied to energy disaggregation. In: Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments. 2015, p. 55–64.
[4] Kim J-G, Lee B. Appliance classification by power signal analysis based on multi-feature combination multi-layer LSTM. Energies 2019;12(14):2804.
[5] Gopinath R, Kumar M, Srinivas K. Feature mapping based deep neural networks for non-intrusive load monitoring of similar appliances in buildings. In: Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation. 2020, p. 262–5.
[6] Kukunuri R, Aglawe A, Chauhan J, Bhagtani K, Patil R, Walia S, Batra N. EdgeNILM: towards NILM on edge devices. In: Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation. 2020, p. 90–9.
[7] Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Shabtai A, Breitenbacher D, Elovici Y. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Comput 2018;17(3):12–22.
[8] Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, Bonawitz K, Charles Z, Cormode G, Cummings R, et al. Advances and open problems in federated learning. Found Trends® Mach Learn 2021;14(1–2):1–210.
[9] Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning. In: International conference on artificial intelligence and statistics. Proceedings of Machine Learning Research (PMLR); 2020, p. 2938–48.
[10] Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. In: Proceedings of the 2016 ACM special interest group on security, audit and control (SIGSAC) conference on computer and communications security. 2016, p. 308–18.
[11] D'Incecco M, Squartini S, Zhong M. Transfer learning for non-intrusive load monitoring. IEEE Trans Smart Grid 2019;11(2):1419–29.
[12] Hudson N, Hossain MJ, Hosseinzadeh M, Khamfroush H, Rahnamay-Naeini M, Ghani N. A framework for edge intelligent smart distribution grids via federated learning. In: 2021 International conference on computer communications and networks (ICCCN). IEEE; 2021, p. 1–9.
[13] Wang H, Si C, Liu G, Zhao J, Wen F, Xue Y. Fed-NILM: A federated learning-based non-intrusive load monitoring method for privacy-protection. Energy Convers Econ 2022;3(2):51–60.
[14] Cao H, Liu S, Zhao R, Xiong X. IFed: A novel federated learning framework for local differential privacy in Power Internet of Things. Int J Distrib Sens Netw 2020;16(5):1550147720919698.

[15] Li B, Wu Y, Song J, Lu R, Li T, Zhao L. DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems. IEEE Trans Ind Inf 2020;17(8):5615–24.

[16] Hart GW. Residential energy monitoring and computerized surveillance via utility power flows. IEEE Technol Soc Mag 1989;8(2):12–6.

[17] Kong W, Dong ZY, Ma J, Hill DJ, Zhao J, Luo F. An extensible approach for non-intrusive load disaggregation with smart meter data. IEEE Trans Smart Grid 2016;9(4):3362–72.

[18] Xia D, Ba S, Ahmadpour A. Non–intrusive load disaggregation of smart home appliances using the IPPO algorithm and FHM model. Sustainable Cities Soc 2021;67:102731.

[19] Salem H, Sayed-Mouchaweh M, Tagina M. Unsupervised Bayesian non parametric approach for non-intrusive load monitoring based on time of usage. Neurocomputing 2021;435:239–52.

[20] Desai S, Alhadad R, Mahmood A, Chilamkurti N, Rho S. Multi-state energy classifier to evaluate the performance of the nilm algorithm. Sensors 2019;19(23):5236.

[21] Cominola A, Giuliani M, Piga D, Castelletti A, Rizzoli AE. A hybrid signature-based iterative disaggregation algorithm for non-intrusive load monitoring. Appl Energy 2017;185:331–44.

[22] Altrabalsi H, Stankovic V, Liao J, Stankovic L. Low-complexity energy disaggregation using appliance load modelling. Aims Energy 2016;4(1):884–905.

[23] Zhang C, Zhong M, Wang Z, Goddard N, Sutton C. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In: Proceedings of the association for the advancement of artificial intelligence (AAAI) conference on artificial intelligence, vol. 32. 2018, no. 1.

[24] Shin C, Joo S, Yim J, Lee H, Moon T, Rhee W. Subtask gated networks for non-intrusive load monitoring. In: Proceedings of the association for the advancement of artificial intelligence (AAAI) conference on artificial intelligence, vol. 33. 2019, p. 1150–7, no. 01.

[25] Bejarano G, DeFazio D, Ramesh A. Deep latent generative models for energy disaggregation. In: Proceedings of the association for the advancement of artificial intelligence (AAAI) conference on artificial intelligence, vol. 33. 2019, p. 850–7, no. 01.

[26] Mishra M, Nayak J, Naik B, Abraham A. Deep learning in electrical utility industry: A comprehensive review of a decade of research. Eng Appl Artif Intell 2020;96:104000.

[27] Li Q, Wen Z, Wu Z, Hu S, Wang N, Li Y, Liu X, He B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Trans Knowl Data Eng 2021.

[28] Dai S, Meng F, Wang Q, Chen X. Federatednilm: A distributed and privacy-preserving framework for non-intrusive load monitoring based on federated deep learning. In: 2023 International joint conference on neural networks (IJCNN). IEEE; 2023, p. 01–8.

[29] Zhang Y, Tang G, Huang Q, Wang Y, Wu K, Yu K, Shao X. Fednilm: Applying federated learning to nilm applications at the edge. IEEE Trans Green Commun Netw 2022.

[30] Pötter H, Lee S, Mossé D. Towards privacy-preserving framework for non-intrusive load monitoring. In: Proceedings of the twelfth ACM international conference on future energy systems. 2021, p. 259–63.

[31] McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. Proceedings of Machine Learning Research (PMLR); 2017, p. 1273–82.

[32] Nguyen HT, Sehwag V, Hosseinalipour S, Brinton CG, Chiang M, Poor HV. Fast-convergent federated learning. IEEE J Sel Areas Commun 2020;39(1):201–18.

[33] Khan LU, Pandey SR, Tran NH, Saad W, Han Z, Nguyen MN, Hong CS. Federated learning for edge networks: Resource optimization and incentive mechanism. IEEE Commun Mag 2020;58(10):88–93.

[34] Kang J, Xiong Z, Niyato D, Yu H, Liang Y-C, Kim DI. Incentive design for efficient federated learning in mobile networks: A contract theory approach. In: 2019 IEEE VTS Asia pacific wireless communications symposium (APWCS). IEEE; 2019, p. 1–5.

[35] Zhao Y, Chen J, Wu D, Teng J, Yu S. Multi-task network anomaly detection using federated learning. In: Proceedings of the tenth international symposium on information and communication technology. 2019, p. 273–9.

[36] Zhao Y, Chen J, Guo Q, Teng J, Wu D. Network anomaly detection using federated learning and transfer learning. In: International conference on security and privacy in digital economy. Springer; 2020, p. 219–31.

[37] Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. Proc Mach Learn Syst 2020;2:429–50.

[38] Wang H, Zhang J, Lu C, Wu C. Privacy preserving in non-intrusive load monitoring: A differential privacy perspective. IEEE Trans Smart Grid 2020;12(3):2529–43.

[39] Wei K, Li J, Ding M, Ma C, Yang HH, Farokhi F, Jin S, Quek TQ, Poor HV. Federated learning with differential privacy: Algorithms and performance analysis. IEEE Trans Inf Forensics Secur 2020;15:3454–69.

[40] Dwork C, Roth A. The algorithmic foundations of differential privacy. Found Trends® Theor Comput Sci 2014;9(3–4):211–407.

[41] Arachchige PCM, Bertok P, Khalil I, Liu D, Camtepe S, Atiquzzaman M. Local differential privacy for deep learning. IEEE Internet Things J 2019;7(7):5827–42.

[42] Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 2881–90.

[43] Massidda L, Marrocu M, Manca S. Non-intrusive load disaggregation by convolutional neural network and multilabel classification. Appl Sci 2020;10(4):1454.

[44] Kelly J, Knottenbelt W. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes.. Sci Data 2015;2(1):150007.

[45] Kolter JZ, Johnson MJ. REDD: A public data set for energy disaggregation research. In: Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA, vol. 25. 2011, p. 59–62, no. Citeseer.

[46] Murray D, Stankovic L, Stankovic V. An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. Sci data 2017;4(1):1–12.

[47] Precioso D, Gómez-Ullate D. NILM as a regression versus classification problem: the importance of thresholding. 2020, arXiv preprint arXiv:2010.16050.

[48] Massidda L, Marrocu M, Manca S. Non-intrusive load disaggregation via a fully convolutional neural network: improving the accuracy on unseen household. In: 2020 2nd IEEE international conference on industrial electronics for sustainable energy systems (IESES), vol. 1. 2020, p. 317–22.

[49] Singh S, Chouzenoux E, Chierchia G, Majumdar A. Multi-label deep convolutional transform learning for non-intrusive load monitoring. ACM Trans Knowl Discov Data (TKDD) 2022;16(5):1–6.

[50] Manca MM, Massidda L. Deep learning based non-intrusive load monitoring with low resolution data from smart meters. Commun Appl Ind Math 2022;13(1):39–56.

[51] Zhao B, Stankovic L, Stankovic V. On a training-less solution for non-intrusive appliance load monitoring using graph signal processing. IEEE Access 2016;4:1784–99.

[52] Mengistu MA, Girmay AA, Camarda C, Acquaviva A, Patti E. A cloud-based online disaggregation algorithm for home appliance loads. IEEE Trans Smart Grid 2018;10(3):3430–9.

[53] Brownlee J. Deep learning with python: Develop deep learning models on theano and tensorflow using keras. Machine Learning Mastery; 2016.

[54] Yeom S, Giacomelli I, Fredrikson M, Jha S. Privacy risk in machine learning: Analyzing the connection to overfitting. In: 2018 IEEE 31st computer security foundations symposium (CSF). IEEE; 2018, p. 268–82.

[55] Carlini N, Liu C, Erlingsson Ú, Kos J, Song D. The secret sharer: evaluating and testing unintended memorization in neural networks. In: Proceedings of the 28th USENIX conference on security symposium. 2019, p. 267–84.

[56] Dai S, Wang Q, Meng F. A telehealth framework for dementia care: an ADLs patterns recognition model for patients based on NILM. In: 2021 International joint conference on neural networks (IJCNN). IEEE; 2021, p. 1–8.