A Robust and Privacy-Aware Federated Learning Framework for Non-Intrusive Load Monitoring

Vidushi Agarwal, Omid Ardakanian, and Sujata Pal

***Abstract*—With the rollout of smart meters, a vast amount of energy time-series became available from homes, enabling applications such as non-intrusive load monitoring (NILM). The inconspicuous collection of this data, however, poses a risk to the privacy of customers. Federated Learning (FL) eliminates the problem of sharing raw data with a cloud service provider by allowing machine learning models to be trained in a collaborative fashion on decentralized data. Although several NILM techniques that rely on FL to train a deep neural network for identifying the energy consumption of individual appliances have been proposed in recent years, the robustness of these techniques to malicious users and their ability to fully protect the user privacy remain unexplored. In this paper, we present a robust and privacy-preserving FL-based framework to train a bidirectional transformer architecture for NILM. This framework takes advantage of a meta-learning algorithm to handle the data heterogeneity prevalent in real-world settings. The efficacy of the proposed framework is corroborated through comparative experiments using two real-world NILM datasets. The results show that this framework can attain an accuracy that is on par with a centrally-trained energy disaggregation model, while preserving user privacy.**

***Index Terms*—energy disaggregation, transformers, federated learning, privacy, poisoning attack.**

I. Introduction

ELECTRIC utilities were historically grappling with the challenge of increasing efficiency and detecting outages, because they had limited visibility into their networks, especially in the last mile of distribution. With the advent of smart meters and advanced metering infrastructure [1], they were finally able to gather high-resolution, aggregate data from their customers for billing purposes in addition to generating useful insights, from monitoring aberrant power usage patterns and developing virtual energy audits to demand response. Some of these applications rely on the power consumption profile of home appliances, which is not readily available and must be inferred from the aggregate data.

Non-Intrusive Load Monitoring (NILM) [2] is the problem of disaggregating the total household energy use, measured by a smart meter, into the load of individual appliances. It helps reduce the energy consumption and electricity bill of homeowners by providing real-time feedback on appliance-level power consumption [3], improving the load forecasting accuracy, and suggesting load shifting strategies during peak hours [4]. While there are several promising techniques for NILM, including signal processing and probabilistic graphical models, deep learning, and in particular the attention mechanism, has been shown to be more effective [5]. One such deep learning model is based on the notion of sequence-to-sequence (seq2seq) translation, in which a sequence of words, e.g. in one language, is mapped to a sequence in another language. By analogy, in energy disaggregation, the seq2seq translation can be used to map the sequence of the aggregate household demand into the power consumption of individual home appliances. However, traditional seq2seq models often struggle with long input sequences because they use recurrent neural networks (RNNs), which process the input one token at a time. This processing can become computationally expensive for long sequences and make it difficult for the model to learn dependencies between tokens that are far apart in the input sequence [6]. Transformers, however, handle long input sequences more efficiently by processing all tokens in parallel [7]. They use a self-attention mechanism to weigh the importance of each token in the input sequence for each output token, allowing the model to capture long-range dependencies more effectively. This makes transformers a more suitable choice for NILM, where the input sequences can be long and complex, and the model needs to learn dependencies between different parts of the sequence to accurately identify individual loads [8].

Regardless of which machine learning model is used for NILM, sending the smart meter data along with groundtruth power profiles of home appliances to a remote server that trains or runs this model could raise privacy concerns. This is because a passive adversary or an intruder can obtain appliance-level information and use this private information to learn the user's habits and lifestyle, such as when they come home, their preferred temperature setting and activities of daily living [9], [10]. To address these privacy concerns and enable training the NILM model on decentralized data that belong to many clients, possibly with unique appliances and usage patterns, Federated Learning (FL) has been adopted in recent work [10]–[13]. In this framework, the NILM model can be trained by the clients in a collaborative fashion, without requiring them to share their data with a central server [14]. The global model is built by aggregating the model updates performed independently by the clients on their local data.

While FL addresses the concern related to sharing raw data with a central server, there are still several challenges that limit its real-world application. Firstly, FL algorithms are prone to privacy attacks during the exchange of parameters between

Vidushi Agarwal is with Department of Computer Science and Engineering, Indian Institute of Technology Ropar, India & also with Department of Computing Science, University of Alberta, Canada. E-mails: vidushi.19csz0010@iitrpr.ac.in; vagarwa1@ualberta.ca.

Omid Ardakanian is with Department of Computing Science, University of Alberta, Canada. E-mail: oardakan@ualberta.ca.

Sujata Pal is with Department of Computer Science and Engineering, Indian Institute of Technology Ropar, India. E-mail: sujata@iitrpr.ac.in.

clients and the aggregation server [15], [16]. The server might be Honest-But-Curious (HBC), i.e., a passive adversary that follows the aggregation protocol but takes a peek at the clients' updates to extract additional (private) information. Different techniques have been proposed to mitigate this attack, from differential privacy to homomorphic encryption and secure multiparty computation [17]. Secondly, malicious clients may attack the FL model by sending incorrect updates to the server during the training process. This client-side attack has multiple types, namely the poisoning attack, model inversion attack, membership inference attack, and backdoor attack [18]. However, existing defense mechanisms against malicious clients in FL are inadequate and more research must be done on developing robust defense mechanisms against adversarial attacks, particularly those that are difficult to detect or prevent [16]. Moreover, the current FL-based frameworks are not fully effective in the presence of heterogeneity, e.g. when homes contain different appliances or exhibit dissimilar feature distributions [19], [20].

Recent advances in FL motivated researchers to apply this paradigm to NILM [10], [12], [13], but the related work builds on the standard FL framework, failing to address the specific challenges of FL in NILM, namely dishonest clients and heterogeneity. We address this gap in the literature by introducing a robust and privacy-aware FL-based NILM framework. The main contributions of this paper are summarized below.

- We propose a robust NILM framework based on FL. We use the bidirectional transformer architecture that follows the pattern of sequence-to-sequence learning for energy disaggregation to achieve better performance in terms of the diverse smart meter clients. To address the challenge of data heterogeneity, we use Model-Agnostic Meta-Learning (MAML) which allows fast and dynamic adaptation of the model according to the client updates.
- To make the model robust against dishonest clients, we devise a reputation scheme for the selective sampling of clients in every round of FL based on their gradient updates during the global model training.
- Through extensive experiments, we corroborate the effectiveness of the proposed FL-based NILM framework by comparing it with the centrally-trained model.

To the best of our knowledge, this is the first work to study and mitigate the model poisoning attack using real-world NILM datasets that exhibit a high degree of heterogeneity.

II. Related Work

A. NILM Techniques

In the early 1980s, Hart [2] introduced NILM for disaggregating electricity usage measurements. With the growing number of smart meters, more attention has been drawn to this problem, leading to the development of a wide range of algorithms for NILM. This includes probabilistic techniques based on a Hidden Markov Model (HMM) [21], as well as a diverse set of supervised and unsupervised learning techniques, from deep neural networks (DNN) [22]–[25] to clustering analysis [26], transfer learning [27] and factorial HMM [28], [29]. Kelly and Knottenbelt [30] show that DNN yields superior performance compared to combinatorial optimization and factorial HMM. Recent work in this area casts NILM as an instance of sequence-to-point, sequence-to-sequence, or sequence-to-subsequence problem for energy usage prediction. This has led to the use of a bidirectional transformer (BERT) for NILM – a model that is based on the self-attention mechanism and transformer architecture [31], and was originally introduced for natural language processing. Yue et al. [5] proposed addressing NILM using a BERT model that utilizes the self-attention and sequence-to-sequence (seq2seq) mechanism.

Despite their higher accuracy, abundant and diverse data is usually required for training DNN-based NILM models. Thus, applying them to a real-world scenario becomes challenging because every user may not possess enough data, and bandwidth and privacy concerns might dissuade users from sending their data to a central server responsible for model training. These issues can be largely addressed, should the NILM model be trained in a federated manner.

B. Privacy Attacks against FL

An ML model can be attacked in both training and deployment phases. The training-time attacks are poisoning attacks and the inference-time attacks are evasion attacks [32]. In a poisoning attack, the attacker inserts malicious data into the training dataset with the goal of introducing bias into the model so as to make it more likely to misclassify data in a specific way. In an evasion attack, the attacker crafts adversarial examples that can fool the trained model into making incorrect predictions. The attacker's goal is to make the model misclassify a legitimate data point by adding small perturbations to it. Evasion attacks launched against an ML model can be either white-box or black-box. In a white-box attack, the attacker is assumed to have full knowledge of the ML model and its parameters, whereas a black-box attack [33] assumes no knowledge of the model and its parameters but the attacker would have access to its output/prediction (i.e., query access to the model).

Depending on the objective of the attacker, poisoning attacks can be further classified into either targeted or untargeted [34]. In the first category, the adversary compromises the integrity of the model by corrupting certain targeted subtasks while maintaining a good accuracy for other tasks. For instance, an attacker may target an image classifier such that it assigns a wrong label to images with some specific features, while other images are classified correctly. Turning to untargeted attacks, the aim is to poison the whole global model so as to reduce its overall accuracy. The poisoned updates can be either generated during the local data collection (data poisoning attacks) or while training the local model (model poisoning attacks). Nevertheless, poisoning attacks attempt to change the behaviour of the global model in some unwanted manner. Since the model updates are typically based on the updates collected from a large number of clients, the impact of this attack is relatively higher in FL. Intuitively, the easiest way to prevent poisoning attacks would be to evaluate and rate the updates sent by the clients. However, since the client's raw

data is not accessible to the server, this evaluation cannot be done reliably.

The attacks listed above are well-studied in the FL literature [15], [35]. In this paper, we mainly focus on mitigating targeted poisoning attacks that can be either backdoor or label-flipping attacks [36] when user data are non-IID. In label-flipping attacks, the attacker aims to modify the labels of honest training examples of one class to another class, while keeping the features of the data unchanged. Backdoor attacks, on the other hand, involve an attacker injecting a secret pattern into the training data of the targeted class. The pattern acts as a trigger that can be exploited by the attacker to control the model's output when it is presented with input containing that pattern.

C. FL-based NILM Techniques

Despite the extensive research on FL, it has been only recently applied to NILM [10]–[13]. Liu et al. [37] proposed FedMeta, a decentralized and task-adaptive learning scheme, combining FL and meta-learning, to create task-specific models collaboratively. Dai et al. [38] introduced DP^2-NILM, a framework for NILM, focusing on utility optimization and privacy preservation. It evaluates two FL strategies, FedAvg and FedProx, to address data heterogeneity and applies local and global differential privacy for diverse privacy needs. MTFed-NILM [39] is a multi-task FL algorithm for NILM, focusing on disaggregating main readings into appliance-level energy consumption while preserving user privacy. Zhou et al. [40] proposed a household load forecasting method utilizing federated deep learning (FedDL) and NILM. The method disaggregates integrated power into individual appliance consumption, using a federated bidirectional LSTM-Attention model for accurate prediction while ensuring the accuracy of the forecast. Although our NILM framework is also based on FL, we mainly assess the NILM performance when user data are not independent and identically distributed (i.e., the non-IID case). This is important because customers have different kinds of appliances and usage patterns in the real world. To tackle this problem and enhance the performance of the NILM model in real-world settings, we apply meta learning to federated learning. Thus, our novelty lies in the implementation of this NILM technique and incorporating a reputation model for the detection of poisoning attacks as described in the next section.

III. Preliminaries

A. Non-Intrusive Load Monitoring (NILM)

The aim of NILM is to identify the electricity consumption of individual appliances using aggregate data, e.g. from a smart meter. Since it requires only a single point of measurement and no extra equipment needs to be installed in the house, this identification technique is deemed non-intrusive. The aggregated power load for a home is time-series data denoted by $P = [p_1, p_2, \cdots, p_T]$ where p_t represents the smart meter reading at time t. This value is the sum total of the energy usage of all appliances that were not in the OFF state. Suppose there are I appliances in a given home, the total power consumption at t is given by

$$p_t = \alpha_t + \sum_{i=1}^{I} e_t^i, \tag{1}$$

where e_t^i is the energy usage of i^{th} appliance and α_t represents the measurement error, which is assumed to be small. The disaggregation problem concerns recovering $E^i = [e_1^i, \cdots, e_T^i]$ for every appliance i from the aggregate measurement $P = [p_1 \cdots, p_T]$. Thus, NILM algorithms approximate a function G that performs the following mapping for every time slot:

$$G(p_t) = [e_t^1, \cdots, e_t^i, \cdots, e_t^I]. \tag{2}$$

For the disaggregation task, each target appliance must have a threshold value so that we can determine if it is in the ON or OFF state. Hence, the state for each appliance depends on the specified minimum on and off duration, maximum power, and the on-power threshold. In our experiments, the ON/OFF states are determined by a simple comparison with the on-status thresholds, but the status changes are considered valid only when they last longer than the minimum ON and minimum OFF duration. The state s_t^i of an appliance i at time t is determined as

$$s_t^i = \begin{cases} 1, & e_t^i \geq \lambda^i \\ -1, & otherwise \end{cases} \tag{3}$$

where λ^i represents the threshold value of the i^{th} appliance, and -1 and 1 correspond to the OFF and ON states, respectively.

We note that disaggregation is a regression task in which the energy consumption of individual appliances is estimated. Once the estimates are obtained, they are utilized to determine the state of each appliance. Therefore, NILM is simultaneously a regression task and a status classification task. This inspired the choice of the loss function as we discuss in Section V-C.

Ethical Considerations: In our FL framework for NILM, ethical concerns extend beyond data privacy and include fairness. The risk of data breaches and unauthorized access to sensitive information poses significant challenges to the NILM community. Using standard and widely used and recognized datasets, we ensure the adherence to privacy standards. These datasets have undergone necessary ethical reviews and clearances, particularly with respect to data collection, and are anonymized to protect individual privacy, aligning with standards for research involving human subjects. Furthermore, our use of two representative NILM datasets, which are collected in two countries from households that may contain different types of each appliance, is a step towards addressing issues of fair and equitable representation.

B. Federated Learning (FL) and Non-IID Data

The idea of FL is to take advantage of decentralized data and compute power of client devices to train a machine learning model that achieves high accuracy on the dataset of every individual client. One of the popular algorithms used to aggregate model updates from multiple clients is federated

averaging (FedAvg) [14]. Concretely, in FedAvg, a server calculates the average of all updates received from clients in every round to obtain a global model. The global model is sent to the clients that participate in the next round so they can further update it according to their own data distribution. Given a loss function *f*, the FL's objective can be written as:

$$\min_{w} \ f(w), \quad \text{where } f(w) = \frac{1}{n}\sum_{i=1}^{n} f(X_i, Y_i; w), \qquad (4)$$

where $f(X_i, Y_i; w)$ is the prediction loss on samples of client i, denoted $\{X_i, Y_i\}$, with w being the vector of model parameters and n being the number of participating clients in one round of FL. We assume that clients' datasets have the following characteristics:

- Non-IID: Every client's data may be from a different distribution, such that the data points and labels available locally do not match the global distribution.
- Unbalanced: The number of training samples in the dataset of each client may vary drastically depending on the amount of data they hold.

In the NILM application, clients will likely have different type and number of appliances, e.g., some customers may not have a microwave at all while others have microwaves of different makes and models. Hence, it is reasonable to assume that clients' datasets are independent and not identically distributed.

Some of the existing works use FedProx [41] for handling non-iid data, which introduces an additional regularization term, known as the proximal term, to the standard federated averaging process. However, FedProx relies on hyperparameter tuning that can significantly impact its performance. Selecting appropriate values for hyperparameters, such as the proximal term weight and learning rate, can be nontrivial. Model-Agnostic Meta-Learning (MAML) [42], on the other hand, aims to learn a general initialization that can be quickly fine-tuned to new clients or tasks. This enables MAML to handle variations and imbalances in non-IID data more effectively compared to FedProx. This motivates the use of MAML for handling heterogeneity in this work.

C. Threat Model

In this paper we assume the aggregation server is honest, but clients can be dishonest (malicious). The dishonest clients manipulate their own data, but they cannot observe or manipulate the data of other clients. Multiple dishonest clients may collude and form a group of *sybils* to perform coordinated attacks in federated learning. Non-colluding adversaries can control multiple sets of sybils to carry out poisoning attacks concurrently. We assume that every class of data required in the global model is included in the dataset of at least one honest client. This assumption is necessary because, in the absence of any honest client, the model would not be able to learn anything about the correct classes in the first place.

We primarily focus on targeted poisoning attacks where the attacker sends malicious updates to manipulate the parameters of the global model. The goal of the attacker is to increase the chance of one class being classified incorrectly without changing the probabilities of other classes. This can be done using the label-flipping strategy [18].

D. Motivations for the Poisoning Attack

The rationale behind poisoning attacks on NILM-based systems is either to obtain economic advantages or avoid regulatory compliance issues. One prominent motivation is the exploitation of virtual energy auditing [43]. Here, NILM applications, which analyze appliance-level energy consumption, could be manipulated by attackers possibly affiliated with appliance manufacturers. Attackers could deceive homeowners into purchasing new, seemingly more efficient models by tampering with data to show false inefficiencies in appliances like air conditioners. This reveals a strong economic reason for engaging in poisoning attacks. Moreover, attackers could manipulate NILM data to show reduced energy consumption to falsely claim incentives offered for buying energy-efficient appliances. In commercial contexts, rival companies might engage in poisoning attacks to impair the NILM systems of competitors, thereby gaining an unfair market advantage. Furthermore, attackers could alter energy consumption patterns to either mask aberrant characteristics or create fictitious profiles of a household's energy usage to satisfy regulatory requirements.

IV. The Proposed Solution: Reputation-based Aggregation and Selection

Suppose there are several homes that have different kinds of appliances, record their electricity consumption using a smart meter, and are interested in training an accurate NILM model in a collaborative fashion. These homes are the clients in FL. There is also an aggregation server, presumably owned by the NILM service provider, that receives all the updates from the clients. We outline steps of the proposed FL framework below:

1) The model parameters are initialized and the first round of training begins. At the central FL server, parameters of the global model are initialized randomly and sent to participating clients (households) as illustrated in Figure 1.
2) The households train the received global model on their own data (locally). They follow the BERT deep learning model for NILM as discussed in the next section. As shown in Step 2 of Figure 1, the local model updates are then sent to the global server for model aggregation.
3) The FL server uses the FedAvg algorithm to update the global model. For robust and fault-tolerant aggregation of the updates, we adopt a reputation model and a client sampling technique that takes into account clients' reputation (Steps 3 and 4 of Figure 1) as described in Section IV-B2.
4) The new aggregated optimal model (Step 5 of Figure 1) is then broadcast to the clients and this process repeats until a stopping criterion is met (e.g. maximum number of training rounds reached).

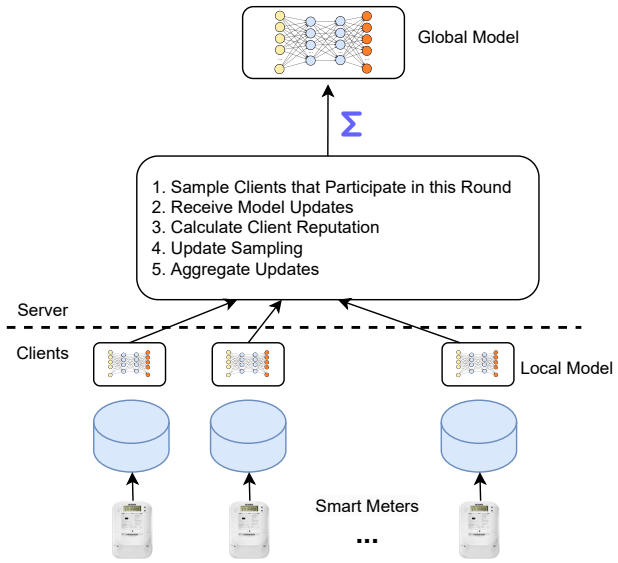


Figure 1: Illustration of the proposed mechanism.

A. The BERT Model for NILM

The model we choose for NILM is the Bidirectional Encoder Representations from Transformers (BERT) model with sequence-to-sequence learning. This model is considered the state-of-the-art and outperforms the other NILM techniques, according to different metrics [5]. The BERT model, with its foundation in transformer architectures, presents a significant advancement over traditional neural network models such as RNNs. Transformers employ the self-attention mechanism, allowing the model to process entire sequences of data simultaneously. This contrasts with older sequence processing methods that handle one data point at a time. The attention mechanism in transformers enable dynamic focus on different parts of the input sequence, which is essential for understanding context and relationships within data. BERT builds upon this by analyzing data bidirectionally, considering both prior and subsequent information in the sequence. This bidirectional processing is particularly effective in NILM, where understanding the sequential context of energy usage is crucial for accurate predictions.

The basis of the model is BERT [31] which consists of an embedding module, transformer layers, and an output multilayer perceptron (MLP). It is shown in [31] that the bidirectional model attains a deeper understanding of the context compared to unidirectional models. The model takes fixed-length sequential data as input and predicts the energy usage of individual appliances, an output of the same shape. The on-power thresholds can then be used to determine the state of each appliance.

The transduction model has an encoder-decoder architecture, where the encoder maps an input sequence to a continuous sequence of symbols. The decoder then uses the encoding to generate an output sequence of symbols by spitting out one element at a time. In the BERT model, the input data is first mapped to a convolutional output by extracting relevant features from the one-dimensional input sequence and increasing the dimensionality of the hidden representation sequence. By employing this convolutional layer, the network is able to capture important patterns and enhance the latent representation of the input data. This layer is then pooled and added to a positional embedding matrix that makes up the sequence positional encoding. The formed embedding matrix is then sent to a bidirectional transformer consisting of several layers of transformers and attention heads within each layer. After the multi-head attention operation in every transformer layer, the previous matrix is further passed through a position-wise feed-forward network (PFFN). By incorporating the PFFN into the architecture, the model captures intricate relationships and patterns within the matrix, leading to improved learning and representation capabilities. Finally, the output MLP consists of a deconvolutional layer followed by two linear layers. The various layers of the BERT model are shown in Figure 2. The training process of BERT involves masking a portion of the input sequence with a special token. This random masking, where a fraction of input elements are masked, allows the model to learn from the surrounding context and predict the masked items. By focusing on the output results from the masked positions, the model is compelled to capture significant patterns from the whole input sequence. Energy prediction is done by the output values multiplied by the maximal device power while corresponding on-power thresholds are used to obtain the appliance status.

B. Reputation-based Aggregation for FL

We propose a reputation-based mechanism for the detection of dishonest clients at the aggregation server. Let $c_{j,t}$ be the gradient received at the server from client j in iteration t.

1) Detecting malicious clients: Sybils in FL provide updates that lead to the poisoning of the global model towards a common objective. In non-IID case, one key insight is that the diversity of the gradient updates can be used to separate honest clients from sybils. Specifically, since the data distribution of each client is different, the updates from sybils will be more similar to each other compared to the ones from honest clients [36]. Using this insight, we assign a reputation value to clients that help in the aggregation process by reducing the chances of suspicious clients being selected. As described in Algorithm 1, cosine similarity is used to calculate the angular distance between client updates. We choose cosine similarity over Euclidean distance because the magnitude of a gradient can be manipulated by sybils to achieve dissimilarity. But they cannot easily change its direction without reducing the effectiveness of the attack. We maintain a history for each client and update the history vector at every iteration into a single aggregated gradient. Thus, instead of using just the update from the current iteration, cosine similarity is calculated using the aggregated historical updates. Finally, the maximum cosine similarity a client has with some other client is compared to the average cosine similarity to calculate the reputation of that client.

2) Reputation calculation and client selection: The selection of clients for an iteration is based on their calculated reputation for high accuracy and robust model training. For

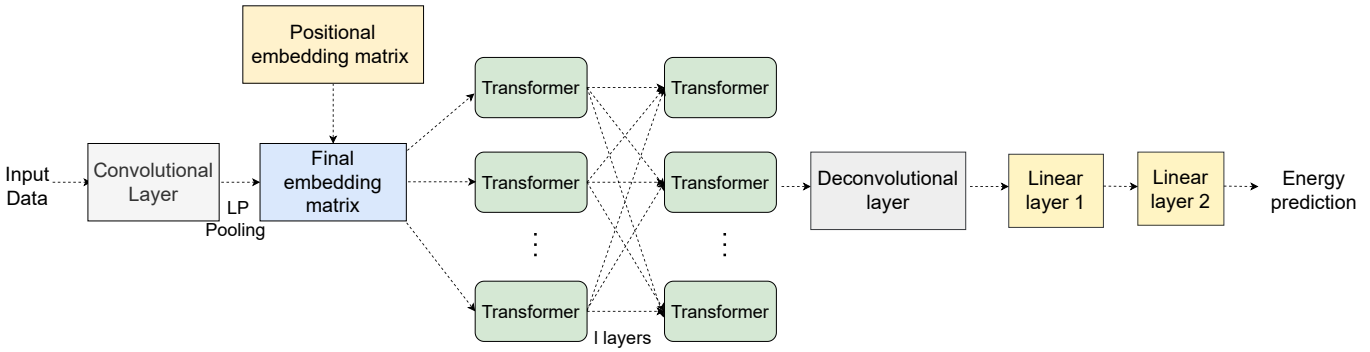


Figure 2: Architecture of the BERT model used in this work.

non-IID data, the reputation is calculated on the basis of cosine similarity with other clients. The value of reputation, r_j is compared to the average of the maximum cosine similarity of all clients and computed according to Steps 11 and 13 of Algorithm 1. In this algorithm, δ represents a hyperparameter that can be tuned to adjust the rate at which reputations are updated. The choice of $t \log t$ as the reputation update function in the algorithm allows for a gradual and balanced growth of the reputation score that reflects the long-term contributions and reliability of each client. It grows faster than linear but slower than quadratic or exponential functions, making it a better choice for updating the reputation. Since the model should converge to the optimal point, changes in reputation become more substantial as the number of iterations increases.

If the reputation score of a client *j* becomes greater than or equal to the threshold β (determined experimentally), it means this client can be chosen for aggregation, as it is considered reputable (Step 14 of the algorithm). Finally, we select *K* number of clients with the highest reputation scores for aggregation in this iteration. By iteratively updating the reputation scores based on the similarity between client histories and selecting the most reputable clients for aggregation, the algorithm aims to identify and choose reliable clients, while detecting potentially malicious or unreliable ones. The reputation scores serve as a measure of trustworthiness, and the threshold β determines the minimum reputation required for a client to affect the global model.

3) Adapting to time-varying clients' behavior: An assumption made in our work is that clients do not change their type, which dictates their behavior, during model training; e.g. an honest client will not start behaving maliciously and vice versa. This can be justified given the short duration of the training phase, which makes type changes unlikely to occur. But this assumption can be relaxed, and clients' reputation can be updated differently to address the following conditions: a) A client initially deemed trustworthy could begin to exhibit malicious behavior; b) A client initially categorized as malicious could later demonstrate normal behavior.

To enhance the adaptability of our framework to behavior changes, our reputation mechanism can be modified to place greater emphasis on the most recent contributions from clients, rather than a cumulative analysis of their entire historical data, enabling a more agile and current reflection of a client's reliability. This can be achieved by considering a sliding window for calculating each user's reputation or implementing exponential smoothing as a forgiveness mechanism.

C. Model Agnostic Meta-Learning for FL

In meta learning, which is a "learning to learn" approach, the goal is to train a model (i.e., the main task) by learning from multiple related subtasks. This model can adapt quickly to new tasks by making use of only a few training iterations and data points. We use the MAML algorithm of [42] as it is compatible with task definition in FL and allows us to address the data heterogeneity challenge. Concretely, in our NILM framework, each subtask involves training the described BERT model on a client's electricity data. The global NILM model is optimized towards the direction that could quickly adapt to all subtasks, each associated with a user's local data. This enables better generalization to heterogeneous data.

The benefits of meta-learning in the NILM framework are manifold. First, it allows the global NILM model to exploit the knowledge gained from training on multiple client datasets, enabling better generalization to heterogeneous data. This is particularly advantageous in FL, where clients may have different types of appliances, energy usage patterns, or household characteristics. Furthermore, meta-learning can enhance the efficiency of the overall NILM process. Since the global model is pre-trained on a diverse set of subtasks, it can quickly adapt to new client data with only a few iterations and data points. Additionally, meta-learning enables knowledge transfer across clients. The global NILM model can capture common patterns and dependencies across different households, appliances, or energy usage scenarios. Overall, meta-learning in the NILM framework enhances adaptability and generalization capabilities of the global model.

The model parameters are trained by minimizing the meta-loss, which measures the performance of the adapted model $f_{\theta_0 i}$ with respect to the model parameters θ across a set of tasks sampled from the distribution $p(T)$. This meta-loss can be expressed as [42]:

$$\min_{\theta} \sum_{T_i \sim p(T)} \ell_{T_i}(f_{\theta_0 i}) = \sum_{T_i \sim p(T)} \ell_{T_i}(f_{\theta - \alpha \nabla_\theta L_{T_i}(f_\theta)}) \quad (5)$$

Here, ℓ_{T_i} represents the loss function for each task T_i (given in Eq. (9)), f_θ denotes the model with parameters θ, $\theta_0 i$ repre-

sents the updated model parameters obtained through gradient descent updates, and α is the step size or learning rate. In summary, our objective is to find the optimal model parameters θ that lead to the best result when updated using one step of gradient descent on subtasks. This process facilitates the global model's ability to adapt swiftly to individual client tasks with minimal fine-tuning.

Algorithm 1 Detecting malicious clients at server

Input: Initial history $h_j = c_{j,1}$, initial reputation $r_j = 1$.
for *iteration* t **do**
  for *every client* j **do**
    Receive $c_{j,t}$ for this round and append it to h_j
    while $i \neq j$ **do**
      $s_{j,i} \leftarrow h_j \cdot h_i / (\|h_j\| \|h_i\|)$
    $w_j \leftarrow max_i(s_j)$ //max cosine similarity of client j
  $\tau_t \leftarrow$ average of w for all clients
  for *every client* j **do**
    if $w_j > \tau_t$ **then**
      $r_j \leftarrow r_j - \delta \cdot t \log t$
    else
      $r_j \leftarrow r_j + \delta \cdot t \log t$
    if $r_j \geq \beta$ **then**
      Client j can be chosen for aggregation
  Select the most reputable K clients for aggregation

Table I: Overview of different appliance values.

Dataset	Appliance	Max Power (W)	On-power threshold (W)	Minimum on Duration (s)	Minimum off Duration (s)
REDD	Microwave	1800	200	12	30
	Dishwasher	1200	10	1800	1800
	Washer	500	20	1800	160
	Fridge	400	50	60	12
UK-DALE	Microwave	3000	200	12	30
	Dishwasher	2500	10	1800	1800
	Washer	2500	20	1800	160
	Fridge	300	50	60	12
	Kettle	3100	2000	12	0

Table II: Dataset details for the appliances in different houses.

House No.	1	2	3	4	5	6
# Appliances	18	9	20	18	24	15
Microwave	✓	✓	✓		✓	
Dishwasher	✓	✓	✓	✓	✓	✓
Washer	✓	✓	✓	✓	✓	✓
Fridge	✓	✓	✓		✓	✓

(a) REDD Dataset

House No.	1	2	3	4	5
# Appliances	52	19	4	11	24
Microwave	✓	✓		✓	✓
Dishwasher	✓	✓			✓
Washer	✓	✓		✓	✓
Fridge	✓	✓		✓	✓
Kettle	✓	✓	✓	✓	✓

(b) UK-DALE Dataset

V. Performance Evaluation

A. Dataset and Preprocessing

We used two real-world energy datasets to evaluate the performance of our framework:

- REDD [44]: The Reference Energy Disaggregation Dataset consists of the electricity consumption data for 6 real houses in the U.S. over several months, with the sampling period of 1s for mains and 6s for appliances.
- UK-DALE [45]: The UK-Domestic Appliance-Level Electricity consists of data from 5 houses in the UK with a sampling period of 1s for mains and 6s for appliances.

These datasets were selected due to their status as standard benchmarks in the field of NILM, offering comprehensive insights from diverse environments. These datasets include both individual appliance and aggregated consumption data, making them appropriate for use in training and test phases [46]. Moreover, they encompass a broad range of appliances, providing a more extensive and varied dataset compared to others that may lack complete appliance coverage or proper labeling. This diversity not only adds reliability to our evaluation but also allows us to effectively test our framework in non-IID scenarios, which is critical for assessing performance in real-world settings.

For the REDD dataset, we choose four specific appliances for training our model: microwave, dishwasher, washer and dryer, and refrigerator. For the UK-DALE dataset, we also include the kettle along with these four appliances. Similar to the preprocessing of BERT4NILM [5], the raw data is resampled and clamped to specify the minimum on- and off-duration, on-threshold, and maximum power of each appliance as given in Table I. The ON/OFF status of each appliance is determined by a comparison between the received data and the on-power thresholds, with the status changes being valid if they last longer than the minimum on/off duration. In the case of REDD, house number 1 is used for testing and other houses are used for training, whereas in UK-DALE, house number 2 is used for testing and other houses are used for training. Different appliances present in the given houses of these datasets are listed in Table II.

B. Baselines

To assess the performance of our proposed approach, we compare with the disaggregation results of the following learning schemes while keeping the neural network architecture unchanged:

- **Centrally-trained model:** The centrally-trained model is the primitive form of ML training where raw data from all the households is aggregated and processed at a central location. Only one central model is trained and tested for all the clients.
- **FL-based NILM model (using FedAvg):** Each client uses its raw dataset to train a local model and sends model updates to the aggregation server. The server then aggregates the updates via FedAvg and trains the final NILM model over several iterations (100 to 150) between the clients and server. Hence, the clients do not need to share their actual data with the server. This baseline is used for evaluating the following two schemes in the proposed framework for non-iid datasets:

– FL-based NILM in the presence of malicious clients: An adversary can impersonate the clients participating in the aggregation process leading to a targeted attack, i.e., the label flipping attack.
– Robust FL-based NILM: The clients can be adversarial aiming to poison the model. We use the aggregation technique outlined in Algorithm 1.

C. Federated Experiments

We evaluate our framework using the sequence-to-sequence (seq2seq) benchmark evaluation and train the BERT model for NILM. In this paper, we have used the PyTorch implementation of BERT4NILM as our base model [1]. To train our model, we split the dataset differently for IID and non-IID settings to simulate a higher number of clients for accurate training as follows:

- IID Scenario: The data from a single house in our dataset is split day-wise and distributed proportionally amongst *n* clients. This is because data from a house will have the same probability distribution and is therefore suitable for the IID scenario.
- Non-IID Scenario: We know that the same appliances from different houses can differ in many aspects, such as voltage and power profiles, energy efficiency, etc. Therefore, for the data to be split in a true non-IID fashion, we assign the data of one appliance from every house to each client.

We set the default sampling period to 6s with a learning rate of 10^{-4} for training the model. We use Adam as the optimization function as it performs better by faster convergence and requires lesser parameters for tuning. The loss function used for training (in all learning schemes) is described next.

1) Loss Function: Following [5], the loss function we use for training the BERT model has multiple terms, each described below.
The first one is the Mean Square Error (MSE) loss for observed and predicted power usage values, which is given by:

$$\ell_{\text{mse}} = \frac{1}{T}\sum_{t=1}^{T}(\hat{e}_t^i - e_t^i)^2, \tag{6}$$

where T is the disaggregation length, i.e., the total number of time steps, $\hat{e}_t^i$ is the predicted energy usage of appliance i at time t, and e_t^i is the corresponding observation as described in Section III. The energy usage values $\hat{e}_t^i, e_t^i$ are normalized between 0 and 1 by dividing them by the maximum power limit. This ensures that the power usage sequences are comparable and consistent across different appliances, regardless of their individual power profile.

To minimize the relative entropy between the predicted and observed power usage, we also incorporate the Kullback–Leibler (KL) divergence in the loss function. The tempered softmax operation applies a temperature parameter to the softmax function that converts a vector of real numbers into a probability distribution. Since electrical appliances are frequently in an off state, we have chosen a temperature parameter of 0.1 to account for the distinction between on-loads and off-loads. This adjustment aims to enhance the performance of the model on error metrics, especially for rarely utilized appliances such as the kettle. A hyper-parameter η is introduced to fine-tune the temperature for our designed loss function. It can be given mathematically as:

$$\ell_{\text{kl}} = D_{\text{KL}}(softmax(\frac{\hat{e}_t^i}{\eta})||softmax(\frac{e_t^i}{\eta})) \tag{7}$$

Finally, to reduce the effect of misclassification and penalize inconsistent predictions, we consider a soft-margin loss and an L1 term which are given by:

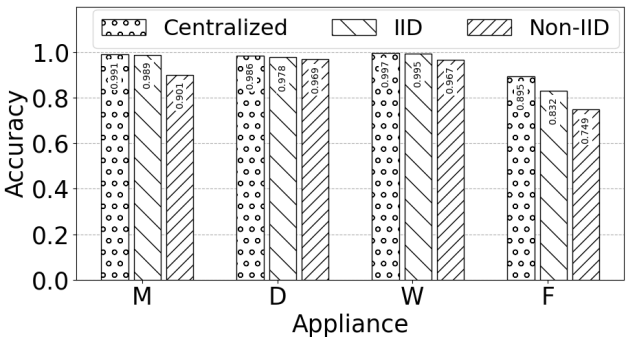
$$\ell_{\text{sm}} = \frac{1}{T}\sum_{t=1}^{T} log(1 + exp(-s_t^i \hat{s}_t^i)) \tag{8}$$

such that s_t^i is the state label of an appliance as described in Section III and $\hat{s}_t^i$ is the corresponding prediction.

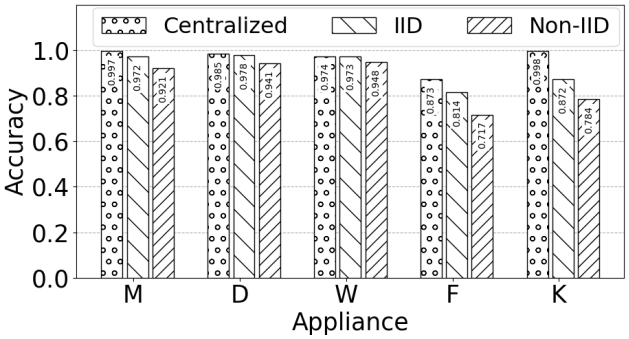
Putting these together, the loss function used to train the BERT model for NILM is as follows:

$$\ell_{total} = \ell_{\text{mse}} + \ell_{\text{kl}} + \ell_{\text{sm}} \tag{9}$$

This loss is specifically designed to encourage the model to make more accurate and consistent predictions, reducing the impact of misclassification. Note that the above loss function is specific to a single appliance, and during training, the losses for all appliances are summed up to compute the total loss.



(a) REDD Dataset



(b) UK-DALE Dataset

Figure 3: Disaggregation accuracy at 150 epochs.

2) Evaluation criteria: For the evaluation of our framework, we adopt three widely used metrics in NILM research: accuracy, F1 score, and mean absolute error (MAE). Accuracy measures the overall correctness of appliance state predictions.

[1]https://github.com/Yueeeeeeee/BERT4NILM

Table III: Average performance scores for REDD

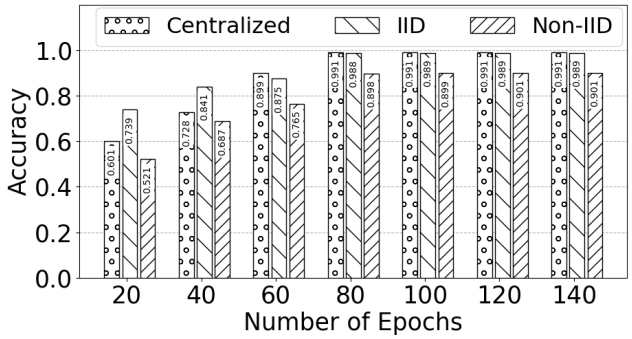
	Microwave			Dishwasher			Washer			Fridge		
	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE
Centrally-trained model	0.991	0.476	17.58	0.986	0.523	20.49	0.997	0.559	34.96	0.895	0.756	32.35
Proposed (IID)	0.988	0.421	17.21	0.955	0.413	22.13	0.989	0.547	35.13	0.736	0.621	36.91
Proposed (Non-IID)	0.896	0.413	18.408	0.964	0.510	21.56	0.951	0.516	35.87	0.656	0.543	38.34

Table IV: Average performance scores for UK-DALE

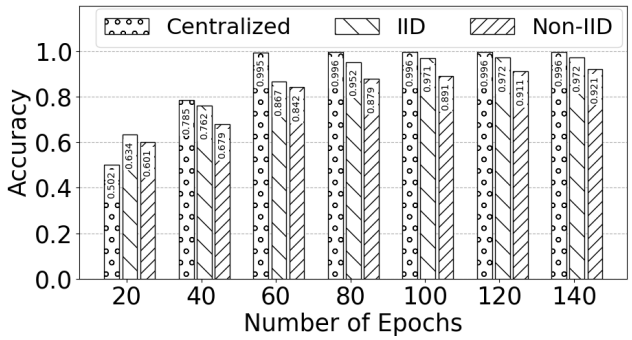
	Microwave			Dishwasher			Washer			Fridge			Kettle		
	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE	Accuracy	F1	MAE
Centrally-trained model	0.997	0.014	6.57	0.985	0.667	16.18	0.974	0.325	6.98	0.873	0.766	25.49	0.998	0.907	6.82
Proposed (IID)	0.951	0.121	5.45	0.963	0.533	17.02	0.956	0.312	6.99	0.687	0.567	31.26	0.785	0.601	15.71
Proposed (Non-IID)	0.876	0.012	4.95	0.919	0.512	21.45	0.898	0.297	8.96	0.632	0.498	32.43	0.701	0.479	19.21

Table V: Parameters used in our experiments

Parameter	**Value/Description**
Datasets	REDD and UK-DALE
Sampling Period	1s for mains, 6s for appliances
Learning Rate	10^{-4}
Optimization Function	Adam
Number of Epochs	150
Batch Size	128
Number of Clients	20 to 100
Model Architecture	BERT4NILM
Software Environment	PyTorch 1.7
Loss Function	MSE + KL divergence + soft-margin loss
Aggregation Algorithm	FedAvg



(a) REDD Dataset.



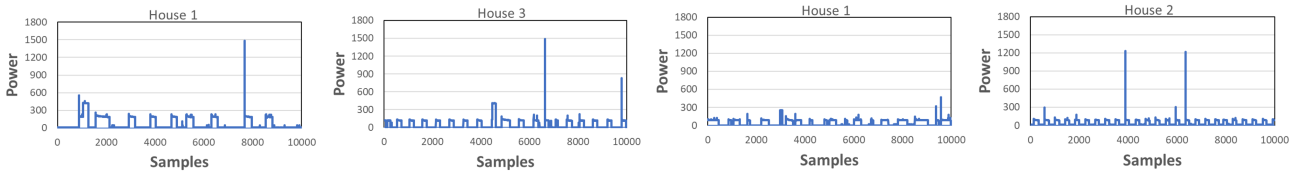
(b) UK-DALE Dataset.

Figure 4: Changes in accuracy when using more epochs

It is calculated as the ratio of correctly predicted states (both ON and OFF) to the total number of appliance state predictions. F1 score, on the other hand, combines precision and recall into a single metric and is particularly useful when the dataset is imbalanced, which is often the case in NILM. Lastly, MAE is calculated by taking the average of the absolute differences between the predicted and true power consumption values per appliance. A summary of the simulation parameters used in our experiments is given in Table V.

VI. Results and Analysis

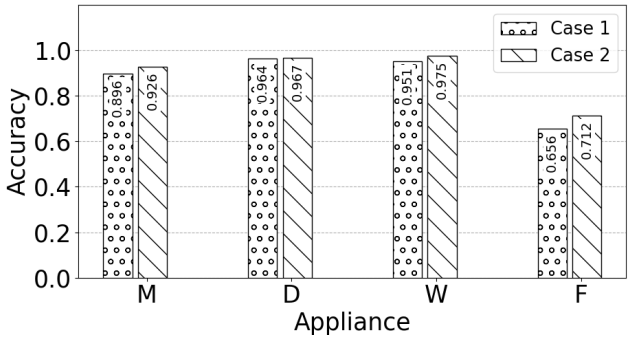
For the centrally trained model and our model trained using FL (in the case of IID and non-IID data), we initially set the number of epochs to 70. Table III and IV show the average performance of these models for different home appliances. It can be seen that the models trained using FL yield satisfactory performance for most of the appliances compared to the traditional models. For fridge, which has a less evident signature due to its relatively low power consumption, the FL-trained models cannot compete with the centrally trained model. Moreover, for less often used appliances such as kettle, an improved masking strategy and more training data could help improve the scores, given the BERT model's complex training mechanism. We have found that by fine-tuning model parameters and increasing the number of global rounds, the FL-trained model achieves better performance. Specifically, when we set the number of global rounds to 150, the performance scores improve drastically as can be seen in Figure 3a and 3b. Moreover, Figure 4 shows that the accuracy increases with the increase in the number of epochs. In these figures



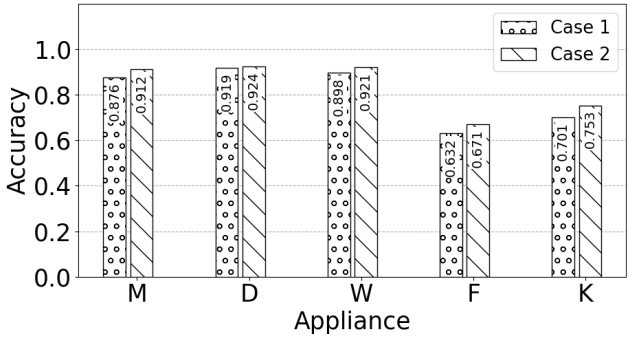
(a) REDD (b) REDD (c) UK-DALE (d) UK-DALE

Figure 5: Example power profile of a fridge (samples are taken every 6 seconds).

and the subsequent ones where appliances are represented on x-axis, we abbreviated the appliance name. Thus, microwave, dishwasher, washer, fridge, and kettle are denoted as M, D, W, F, and K, respectively. Overall, we conclude that federated learning makes possible high performance in the NILM task.



(a) REDD Dataset

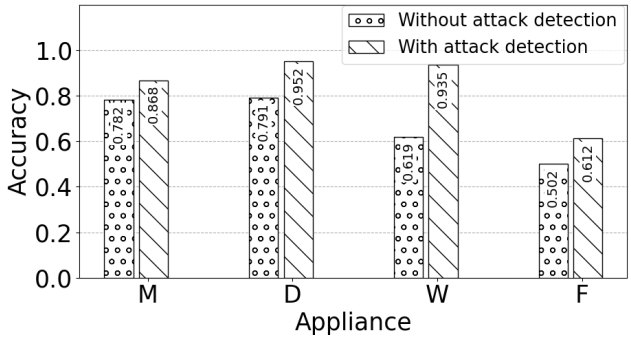


(b) UK-DALE Dataset

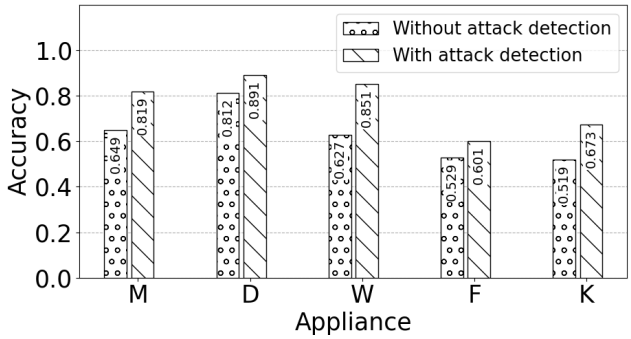
Figure 6: Average accuracy in the non-IID case.

A. *Investigating the Non-IID Case*

We observe through experimentation that by incorporating MAML into our model training, the accuracy for non-IID data is improved by 4% to 6%. Although the scores for the federated model achieved satisfying performance, the values for non-IID cases are still subpar than both the IID and the centralized NILM models. We speculate that the usage patterns and load consumption signatures of the same appliances from different houses may be dissimilar due to several factors. It can be seen from Figure 5 that the load consumption distribution of the appliance fridge for separate houses differs significantly in both datasets. Therefore, we present a comparison of two cases of data heterogeneity that can be taken into account for the non-IID data distribution:

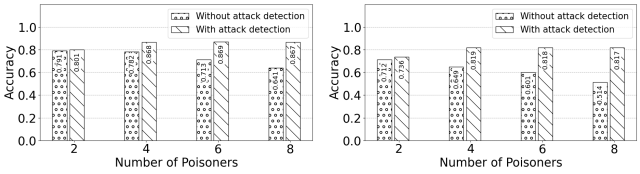


(a) REDD Dataset



(b) UK-DALE Dataset

Figure 7: Performance under model poisoning attack.

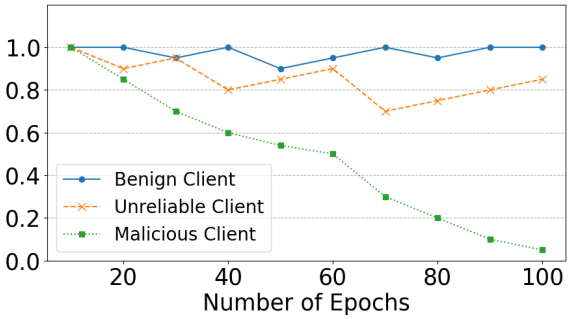


(a) REDD Dataset. (b) UK-DALE Dataset.

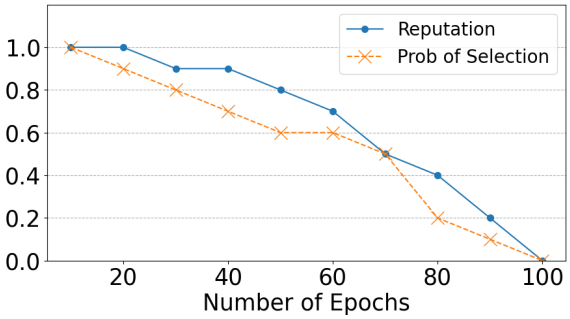
Figure 8: Performance of the attack model for varying number of sybils in case of non-iid data.

- Case 1 (missing classes): Each client owns the data from a particular appliance of a different house.
- Case 2 (heterogeneous data): 80% of the data owned by a client is from a particular appliance of a specific house and 20% of the data can belong to other heterogeneous classes of appliances.

Figure 6a and 6b present the average accuracy for the two cases and highlight the improvement when the degree of heterogeneity is decreased.



(a) Reputation scores for different clients.



(b) Probability of a malicious client being selected.

Figure 9: Visualization of reputation scores.

B. Evaluating the Robust FL Framework

We now evaluate the performance of our reputation-based aggregation scheme in the non-IID case when there are some dishonest clients. Figure 7a and 7b show the performance of our NILM model in REDD and UK-DALE datasets, respectively. It can be seen that without using the proposed robust FL framework the model performance drops drastically under the model-poisoning attack. However, using our framework, we can keep the accuracy high despite the model-poisoning attack. Moreover, the model performance does not decline with the increase in the number of dishonest clients as can be seen from Figure 8a and 8b. We also show that the probability by which a client is selected in a particular round decreases with the decrease in his reputation, thus mitigating the selection of dishonest clients. Figure 9a represents the variation of reputation values for a benign client (honest and accurate), unreliable client (honest but might be inaccurate due to non-iid data), and malicious client (dishonest). As we can see from Figure 9b, if we set the value of δ to 0.01, the probability of selecting a client falls below 50% if its reputation goes below 0.6. In conclusion, our reputation mechanism demonstrates robust capabilities in identifying dishonest clients, safeguarding the model's integrity and contributing to the overall reliability of our framework.

VII. Conclusion

In this paper, we proposed a federated learning-based framework for real-life NILM applications. The proposed framework offers a solution for energy consumption monitoring while preserving user privacy. By using a bidirectional transformer architecture, a meta-learning algorithm to handle data heterogeneity, and a reputation mechanism for the selective sampling of clients, we were able to achieve high accuracy and robust against privacy attacks by malicious clients. The experimental results demonstrate the efficacy of the proposed framework on real-world energy datasets in terms of various parameters. In future work, we aim to study the trade-off between model performance and privacy protection. We will focus on other privacy attacks that could pose a threat to our framework and develop techniques to mitigate them.

Acknowledgment

This work was partially supported through an Overseas Visiting Doctoral Fellowship sponsored by the Science and Engineering Research Board (SERB), India.

References

[1] O. Ardakanian, *Advances in Distribution System Monitoring*, pp. 13–16. Cham: Springer International Publishing, 2020.
[2] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
[3] C. Fischer, "Feedback on household electricity consumption: a tool for saving energy?," *Energy efficiency*, vol. 1, no. 1, pp. 79–104, 2008.
[4] M. Zeifman and K. Roth, "Nonintrusive appliance load monitoring: Review and outlook," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, 2011.
[5] Z. Yue, C. R. Witzig, D. Jorde, and H.-A. Jacobsen, "BERT4NILM:: A bidirectional transformer model for non-intrusive load monitoring," in *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*, pp. 89–93, ACM, 2020.
[6] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2469–2489, 2019.
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
[8] S. Sykiotis, M. Kaselimi, A. Doulamis, and N. Doulamis, "Electricity: An efficient transformer for non-intrusive load monitoring," *Sensors*, vol. 22, no. 8, p. 2926, 2022.
[9] Y. Himeur, A. Alsalemi, F. Bensaali, A. Amira, and A. Al-Kababji, "Recent trends of smart nonintrusive load monitoring in buildings: A review, open challenges, and future directions," *International Journal of Intelligent Systems*, vol. 37, no. 10, pp. 7124–7179, 2022.
[10] H. Wang, C. Si, G. Liu, J. Zhao, F. Wen, and Y. Xue, "Fed-NILM: A federated learning-based non-intrusive load monitoring method for privacy-protection," *Energy Conversion and Economics*, vol. 3, no. 2, pp. 51–60, 2022.
[11] Y. Zhang, G. Tang, Q. Huang, Y. Wang, K. Wu, K. Yu, and X. Shao, "Fednilm: Applying federated learning to NILM applications at the edge," *IEEE Transactions on Green Communications and Networking*, 2022.
[12] H. Wang, C. Si, and J. Zhao, "A federated learning framework for non-intrusive load monitoring," *arXiv preprint arXiv:2104.01618*, 2021.
[13] H. Pötter, S. Lee, and D. Mossé, "Towards privacy-preserving framework for non-intrusive load monitoring," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, pp. 259–263, 2021.
[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.
[15] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948, PMLR, 2020.
[16] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*, pp. 634–643, PMLR, 2019.
[17] H. Hu, Z. Salcic, L. Sun, G. Dobbie, and X. Zhang, "Source inference attacks in federated learning," in *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 1102–1107, IEEE, 2021.
[18] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[19] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021.
[20] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 965–978, IEEE, 2022.
[21] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
[22] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.
[23] C. Shin, S. Joo, J. Yim, H. Lee, T. Moon, and W. Rhee, "Subtask gated networks for non-intrusive load monitoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1150–1157, 2019.
[24] G. Bejarano, D. DeFazio, and A. Ramesh, "Deep latent generative models for energy disaggregation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 850–857, 2019.
[25] J. Wang, S. El Kababji, C. Graham, and P. Srikantha, "Ensemble-based deep learning model for non-intrusive load monitoring," in *2019 IEEE Electrical Power and Energy Conference (EPEC)*, pp. 1–6, IEEE, 2019.
[26] S. Desai, R. Alhadad, A. Mahmood, N. Chilamkurti, and S. Rho, "Multi-state energy classifier to evaluate the performance of the NILM algorithm," *Sensors*, vol. 19, no. 23, p. 5236, 2019.
[27] X. Chang, W. Li, C. Xia, Q. Yang, J. Ma, T. Yang, and A. Y. Zomaya, "Transferable tree-based ensemble model for non-intrusive load monitoring," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 4, pp. 970–981, 2022.
[28] W. Kong, Z. Y. Dong, J. Ma, D. J. Hill, J. Zhao, and F. Luo, "An extensible approach for non-intrusive load disaggregation with smart meter data," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3362–3372, 2016.
[29] D. Xia, S. Ba, and A. Ahmadpour, "Non–intrusive load disaggregation of smart home appliances using the IPPO algorithm and fhm model," *Sustainable Cities and Society*, vol. 67, p. 102731, 2021.
[30] J. Kelly and W. Knottenbelt, "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*, pp. 55–64, 2015.
[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
[32] Y. Deldjoo, T. D. Noia, and F. A. Merra, "A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.
[33] J. Wang and P. Srikantha, "Stealthy black-box attacks on deep learning non-intrusive load monitoring models," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3479–3492, 2021.
[34] Z. Tian, L. Cui, J. Liang, and S. Yu, "A comprehensive survey on poisoning attacks and countermeasures in machine learning," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–35, 2022.
[35] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1605–1622, 2020.
[36] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.
[37] R. Liu and Y. Chen, "Learning task-aware energy disaggregation: a federated approach," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, pp. 4412–4418, IEEE, 2022.
[38] S. Dai, F. Meng, Q. Wang, and X. Chen, "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)*, pp. 01–08, IEEE, 2023.
[39] X. Wang and W. Li, "Mtfed-nilm: Multi-task federated learning for non-intrusive load monitoring," in *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pp. 1–8, IEEE, 2022.
[40] X. Zhou, J. Feng, J. Wang, and J. Pan, "Privacy-preserving household load forecasting based on non-intrusive load monitoring: A federated deep learning approach," *PeerJ Computer Science*, vol. 8, p. e1049, 2022.
[41] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.
[42] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, PMLR, 06–11 Aug 2017.
[43] Y. Wang, A. Pandharipande, and P. Fuhrmann, "Energy data analytics for nonintrusive lighting asset monitoring and energy disaggregation," *IEEE Sensors Journal*, vol. 18, no. 7, pp. 2934–2943, 2018.
[44] J. Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, vol. 25, pp. 59–62, Citeseer, 2011.
[45] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.
[46] L. Pereira and N. Nunes, "Performance evaluation in non-intrusive load monitoring: datasets, metrics, and tools—a review," *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, vol. 8, no. 6, p. e1265, 2018.

Vidushi Agarwal is currently a visiting research fellow at the University of Alberta, Canada, and a full-time Ph.D. research scholar at the Department of Computer Science and Engineering at the Indian Institute of Technology, Ropar. She is the recipient of the Tata Consultancy Services (TCS) Research Scholar Program Fellowship, by TCS Research India. She was a Gold Medalist in M.Tech, Computer Science and Engineering from the National Institute of Technology, Hamirpur, India in 2019. Her current areas of research interest include the Internet of Things, Wireless Sensor Networks, Data Security, Blockchain, and Federated Learning.

Omid Ardakanian (M'15) is an Associate Professor in the Department of Computing Science at the University of Alberta, Canada. He received his B.Sc. from Sharif University of Technology in 2009, and M.Math. and Ph.D. from the University of Waterloo in 2011 and 2015, respectively. From 2015 to 2017, he was an NSERC Postdoctoral Fellow at the University of California, Berkeley and the University of British Columbia. His research focuses on the design and implementation of networked systems. He served as Guest Editor for a special issue of IEEE Transactions on Smart Grid and is currently serving as Area Editor for ACM SIGENERGY Energy Informatics Review.

Sujata Pal is an Assistant Professor in the Department of CSE at IIT Ropar. She received her Ph.D. degree from IIT Kharagpur. She was a recipient of the TCS Scholarship. She was awarded the prestigious Schlumberger Faculty for the Future Fellowship for 2015 and 2016. She was a Postdoctoral fellow at the University of Waterloo, Canada, before joining IIT Ropar. Her research works have been published in high quality international journals, such as IEEE TCE, TMC, TPDS, TC, TCyb, ACM Computing Surveys, FGCS, International conferences, such as ICC, IEEE MASS, GLOBECOM, MobiCom, SenSys, a book and several book chapters. Her current research interests include IoT, blockchain, WBANs, software defined networks and mobile ad-hoc networks. Visit http://cse.iitrpr.ac.in/dr-sujata-pal for more information.