Poisoning Attack Mitigation for Privacy-Preserving Federated Learning-based Energy Theft Detection

Mahmoud Srewa[1], Michaela F. Winfree[2], Mohamed I. Ibrahem[3,4], Mahmoud Nabil[5], Rongxing Lu[6], and Ahmad Alsharif[1,4]

[1]Department of Computer Science, University of Alabama, AL 35487, USA
[2]Department of Computer Science, University of Kentucky, KY 40506, USA
[3]School of Computer and Cyber Sciences, Augusta University, Augusta, GA 30912, USA
[4]Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Egypt
[5]Electrical and Computer Engineering, University of North Carolina A&T, NC 27411, USA
[6]Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada

ICC 2024 - IEEE International Conference on Communications | 978-1-7281-9054-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICC51166.2024.10622191

***Abstract*—In federated learning (FL) based electricity theft detection, detection nodes (DNs) locally train deep learning models on consumers' data and share only the local model parameters with an aggregation server (AS) to generate a global model shared by all nodes for better detection accuracy. However, several privacy concerns should be addressed including membership and inference attacks. To mitigate these attacks, several privacy-preserving aggregation schemes have been introduced. Nevertheless, existing FL detectors often overlook the threat of poisoning attacks, in which certain DNs hold maliciously labeled, i.e., poisoned, data during the training. This manipulated data can subsequently be exploited to introduce backdoors into the global model after its deployment. This paper introduces a novel approach that enhances privacy and resilience against poisoning attacks in FL-based electricity theft detection within smart grids. Our approach enables encrypting local parameters before sending them to the AS, thus safeguarding consumers' privacy. Additionally, it utilizes a cosine similarity test over encrypted data to detect and mitigate poisoning attacks by filtering out malicious local gradients from being considered in the global model computation. Through extensive evaluations, we demonstrate the effectiveness of our FL-based detector in substantially reducing the poisoning attack success rate even when 50% of DNs train their local models with malicious targeted power consumption data, all while preserving consumers' privacy.**

***Index Terms*—Federated Learning, Poisoning Attack, Privacy Preservation.**

I. Introduction

Electricity theft is one of the major concerns in the current power grids due to its negative financial impact. For instance, recent reports indicate that the annual financial loss due to electricity theft is about $6B, $173M, and $100M in the United States, the United Kingdom, and Canada respectively [1]. Electricity theft not only causes economic losses but also results in a disrupted and unstable grid operation that may result in power outages [2]. Therefore, there is a necessity to make the power grid smarter and immune to these attacks.

The Smart Grid (SG) is a modernized power grid that utilizes cutting-edge technologies, equipment, and controls that offer two-way communication between various grid entities to ensure efficient and reliable grid operation and energy management [3]. A fundamental component of the smart grid is Advanced Metering Infrastructure (AMI) networks in which Smart Meters (SMs) are installed at consumer's premises to provide utility companies with extensive and high-frequency electricity consumption data. Such data empower electric utilities to analyze and process real-time energy consumption data as well as providing consumers with a substantial degree of convenience in managing their energy consumption [4].

Because SMs are embedded systems running software programs, they may expose the SG to cyber-attacks. Specifically, malicious consumers may hack into their SMs to steal electricity by manipulating their consumption data and hence reduce their electricity bills. To overcome these cyber-attacks, Deep Learning (DL)-based electricity theft detection has emerged as the most effective approach for detecting electricity theft [5]. This is primarily attributed to the ability of DL models to learn and exploit correlations within consumption readings. In a DL-based approach, a Detection Node (DN) trains a DL model over its consumers' energy consumption profiles.

Moreover, to develop a more accurate and robust energy theft detector, several DNs, which are typically owned/operated by different utility companies, use Federated Learning (FL) to collaboratively train a global model over a more diverse and larger energy consumption datasets without sharing their local consumers' data to preserve consumers' privacy [6]. In this scenario, an Aggregation Server (AS) collects the local model's parameters from each DN to compute the global model parameters and send them back to DNs.

FL-based approaches are vulnerable to model inversion, membership inference, and model poisoning attacks. For instance, revealing local models' parameters facilitates launching model inversion and membership inference attacks and hence leaks sensitive information about consumers' consumption profiles [7]. To address this critical issue, various privacy-preserving FL approaches have been proposed to allow the AS to train a global model using *encrypted* local parameters while preventing adversaries from launching model inversion and membership inference attacks [8], [9]. Nevertheless, the

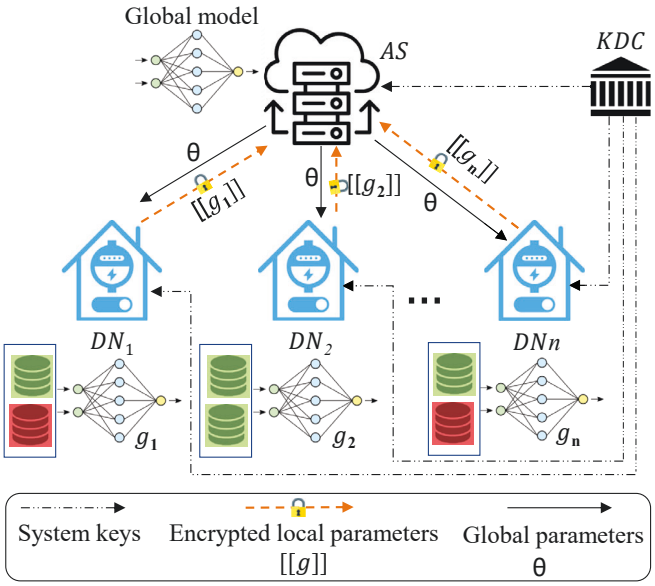

Figure 1: System model

aforementioned privacy-preserving FL-based approaches not only remain vulnerable to model poisoning attacks but also make it difficult for the AS to detect local model poisoning since the local model parameters are encrypted [10], [11].

A. Related Works and Limitations

In [6], the authors introduced an FL-based electricity theft detection framework with majority voting classifiers. However, during the learning phase, DN sends raw weights to the AS, posing a privacy risk as the training data may be inferred using the local model's weights. The authors in [8] addressed this privacy concern by introducing the FedDetect framework, where homomorphic encryption is used to encrypt the local models' parameters to preserve the customers' privacy while allowing the AS to build the global model securely. However, FedDetect requires the existence of two non-colluding servers to cooperate during the FL training process to achieve the privacy preservation goals. In [9], the authors developed a decentralized functional encryption scheme to mitigate membership and inference attacks in an FL-based electricity theft detection.

However, none of these research works had addressed targeted model poisoning attacks in the FL-based electricity theft detection in which poisoned power consumption profiles used during the FL training can create a backdoor for malicious consumers to steal electricity while being undetected. Although other research works [10], [11] address mitigating model poisoning attacks in other settings and applications, they rely on the impractical two non-colluding server assumption that is not suitable in several FL scenarios.

B. Our Contributions

The research problem we address is how to enable DNs to build a robust electricity theft detection model that is not only immune to model inversion and membership inference attacks but also mitigates the threats of model poisoning attacks without learning the consumers' training data or the local models' parameters to preserve consumers' privacy.

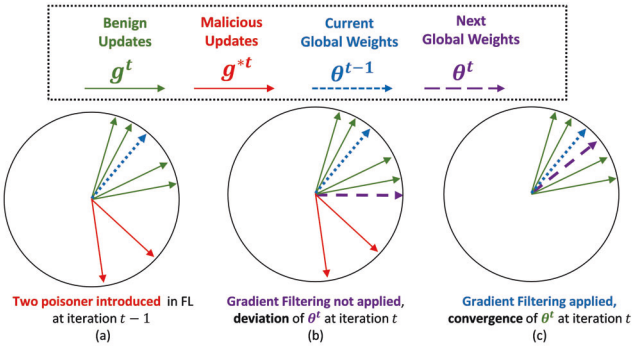

Figure 2: Impact of Poisoning attack on FL convergence

The main contributions of this work are as follows:

- We propose novel secure and privacy-preserving schemes that mitigate model poisoning attacks while preventing model inversion and membership inference attacks. In specific, our scheme allows an honest-but-curious aggregation server to securely filter out the poisoned local gradient updates submitted by the detection nodes by computing the cosine similarity between the encrypted local gradient updates and current global parameter without revealing the local gradient updates to preserve the consumers' privacy during the FL training.
- Using a real energy consumption dataset, we conducted extensive experiments and our results validate that (1) a high poisoning attack success rate can be achieved in the existing privacy-preserving FL-based electricity theft detection solutions, and (2) our scheme can mitigate model poisoning attacks by reducing the attack success rate while achieving high overall model accuracy.

The remainder of this paper is organized as follows. Section II describes the system model and design goals. The proposed scheme is presented in Section III. The security analysis and performance evaluation are discussed in Section IV. Finally, Section V concludes our work.

II. System Models and Design Goals

A. Network Model

- **Aggregation Server (AS)**: In the FL training's initialization phase, the AS sends to all the DNs an initial model with default parameters. In each FL training iteration, it receives encrypted local gradients from DNs, filters out malicious gradients using a privacy-preserving cosine similarity test, and computes new global model parameters from the remaining encrypted local gradients. Figure 2 shows the concept of filtering malicious gradients from being included in the global parameters' computation.
- **Detection Node (DN)**: A set of DNs $\mathbb{DN} = \{DN_i, 1 \leq i \leq n\}$ collaborates in a FL environment to construct a robust electricity theft DL model. Each DN initially

receives the detection model from the AS. In each FL training iteration, a DN trains the model with its dataset to generate new local model gradients. It then employs the proposed masking and encryption technique to send encrypted local training parameters to the AS.

- **Key Distribution Center (KDC)**: KDC is a trusted and independent entity that distributes and manages all the public and private keys in the initialization setup phase. The KDC will not be involved in any further process.

B. Threat Model and Design Goals

The DNs and AS are considered honest-but-curious; they will follow the proper operation of the scheme without interruption. However, they are curious to learn about other DN's power consumption profiles through model inversion and membership inference attacks. Moreover, each DN cannot ensure the trustworthiness of the power consumption profiles used by other DNs to train their local models. If some of these power consumption profiles contain a backdoor, i.e., power consumption profiles of electricity theft that are mislabeled as benign profiles, then model poisoning attacks would become successful and this backdoor would be exploited by malicious consumers to steal electricity. Therefore, our design goals are:

1) **Poisoning Attack Resistance**: The proposed scheme should mitigate model poisoning attacks by eliminating the encrypted local gradient updates that are honestly calculated using the local poisoned consumer profiles.
2) **Privacy Preservation**: The proposed scheme should resist model inversion and membership inference attacks, i.e., consumers' power consumption data should not be leaked to eavesdroppers, AS, or any other DN.

III. The Proposed Scheme

This section presents our scheme that is developed based on, but not limited to, secure inner product (SIP) techniques and ID-based cryptography (IBC). We constructed a variant of our proposal at [12] that integrates lightweight one-time masks generated in a non-interactive manner using IBC. This would allow the AS to measure the cosine similarity between the encrypted local gradient updates and the current global model parameters such that the encrypted poisoned gradient updates are filtered out in the ciphertext domain. Furthermore, our scheme ensures privacy-preserving computation of global parameters' updates from the non-poisoned local updates without revealing the individual local gradients to prevent model inversion and membership inference attacks.

A. System Setup

The KDC generates the keys required for the SIP computation as follows. It generates a master key set $\mathcal{MK} = \{M_1, M_2, N_1, N_2, N_3, N_4\}$, such that all elements in $\mathcal{MK}$ are random invertible matrices of size $(p+1) \times (p+1)$ where, p is the size of the DL model parameters and sets the AS's key as $\mathcal{ASK} = \{M_1N_1^{-1}, M_1N_2^{-1}, M_2N_3^{-1}, M_2N_4^{-1}\}$. In addition, for every node DN_i, the KDC generates a unique key as $\mathcal{DNK}_i = \{N_1A_i, N_2B_i, N_3C_i, N_4D_i\}$ such that $A_i + B_i = M_1^{-1}$ and $C_i + D_i = M_2^{-1}$ where A_i, B_i, C_i, D_i of size $(p+1) \times (p+1)$. Finally, the KDC generates a splitting binary vector sv of size $(p+1)$.

Table I: Main notations

Notation	Description
$\mathbb{DN}$	Set of Detection Nodes $\mathbb{DN} = \{D_i, 1 \leq i \leq n\}$
p	Size of deep learning model parameters
$\mathcal{MK}$	Master key set for SIP
	$\mathcal{MK} = \{M_1, M_2, N_1, N_2, N_3, N_4\}$
$\mathcal{DNK}_i$	DN_i key $\mathcal{DNK}_i = \{N_1A_i, N_2B_i, N_3C_i, N_4D_i\}$
$\mathcal{ASK}$	Aggregation server key $\mathcal{ASK} = \{M_1N_1^{-1}, M_1N_2^{-1}, M_2N_3^{-1}, M_2N_4^{-1}\}$
sv	Splitting vector used during encryption
$\{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, q, Q, H\}$	ID-based cryptography public parameters
Q_i, X_i	ID-based public/private key pair of DN_i
$\mathcal{H}(K, m)$	Keyed hash function
K_{ij}	Shared key between DN_i and DN_j
g_i^t	Local gradients of DN_i at FL-iteration t
mv_i^t	Masking vector of DN_i at FL-iteration t
$g_i^{(m)t}$	Masked local gradients of DN_i at FL-iteration t
$[\![g_i^t]\!]$	Encrypted local gradients of DN_i at FL-iteration t
w^t	Global model at FL-iteration t
w_i^t	DN_i Local model at FL-iteration t
θ^t	Global aggregated gradients at FL-iteration t
cs^t	Cosine similarity vector of al DNs at FL-iteration t
T	Elimination threshold
$\mathbb{DN}^*$	Set of DNs with poisoned gradients updates

In order to empower all the DNs with the ability to establish pairwise one-time masks, the KDC generates the ID-based system parameters by choosing bilinear pairing-based parameters$\{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, q\}$, a master secret $s \in \mathbb{Z}_q^*$ and computes the corresponding public key as $Q = sP \in \mathbb{G}_1$. It also chooses a cryptographic hash function H defined as $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ and a keyed hash function $\mathcal{H}(K, m)$ where K is the key used to calculate the hash of an input m. Finally, it sets the system public parameters to $\{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, q, Q, H, \mathcal{H}\}$. For each DN_i with an identity ID_i, the KDC generates the DN's ID-based secret key X_i, public key Q_i as follow, $X_i = sQ_i$ where the $Q_i = H(ID_i)$.

At the end of this phase, the AS receives its key $\mathcal{ASK}$ and each DN_i receives its key $\mathcal{DNK}_i$ and the ID-based public/private key pair Q_i/X_i. Finally, the AS constructs an initial DL model, generates the initial model parameters weight W^1, and distributes this parameter to all $\mathbb{DN}$.

After each DN receives its keys and the public parameters, each DN_i can compute a pairwise symmetric key K_{il} to be shared with every other $DN_l \in \mathbb{DN}$ in the system in a non-interactive manner. DN_i computes the key as $K_{il} = \hat{e}(X_i, Q_l) = \hat{e}(Q_i, Q_l)^s$, whereas DN_l computes the same key as $K_{il} = \hat{e}(Q_i, X_l) = \hat{e}(Q_i, Q_l)^s$. This key will be used to derive unique one-time masks used during every FL iteration

as will be shown in the next subsection.

B. Local gradient encryption

At an FL-iteration t, each DN_i learns its local gradient vector g_i^t using stochastic gradient descent over its local electricity consumption profiles. Then it applies the proposed scheme to generate an encrypted gradient $[\![g_i^t]\!]$ to be sent to the AS. The detailed process is as follows.

Local Gradients Masking: At iteration t, each DN_i generates the t-th iteration p-dimensional masking vector mv_i^t. In specific, the z-th element in mv_i^t is calculated as follow:

$$mv_i^t(z) = \sum_{\substack{l=1 \\ 1\le l\le|\mathbb{DN}|}}^{l<i} \mathcal{H}(K_{il}, t||z) \; - \sum_{\substack{l>i \\ 1\le l\le|\mathbb{DN}|}}^{|\mathbb{DN}|} \mathcal{H}(K_{il}, t||z)$$

Then, DN_i normalizes g_i^t as follow: $c = \frac{c}{||g_i^t||} \;\forall\; c \in g_i^t$ and masks the normalized g_i^t to generate a masked gradient vector $g_i^{(m)t} = g_i^t + mv_i^t$.

To ensure the correctness of the cosine similarity computation over encrypted data, DN_i appends to the masked vector an additional mask cancelation scalar value mc_i^t that is computed as $mc_i^t = \langle mv_i^t, w^{t-1}\rangle$, where w^{t-1} is the global weight at $t-1$ iteration.

Local Gradients Encryption: In this phase, DN_i utilizes its key $\mathcal{DNK}_i$ to encrypt $g_i^{(m)t}$ as follows.

1) DN_i uses the split vector sv to split the masked gradient vector $g_i^{(m)t}$ into two vectors $g_i^{t'}$ and $g_i^{t''}$ such that, the $z-$th element in both $g_i^{t'}$ and $g_i^{t''}$ is calculated as follow:

$$g_i^{t'}(z) = g_i^{t''}(z) = g_i^{(m)t}(z) \quad \text{if} \quad sv(z) = 1$$
$$g_i^{t'}(z) = r_z,\; g_i^{t''} = g_i^{(m)t}(z) - r_z \quad \text{if} \quad sv(z) = 0$$

, where r_z is a random number.

2) DN_i uses its encryption key $\mathcal{DNK}_i$ to generate its encrypted local gradients components $[\![g_i^t]\!]$ as

$$[\![g_i^t]\!] = \left[\, N_1 A_i g_i^{t'} \;\;,\;\; N_2 B_i g_i^{t'} \;\;,\;\; N_3 C_i g_i^{t''} \;\;,\;\; N_4 D_i g_i^{t''} \right]$$

Finally, DN_i sends the encrypted local gradient vector $[\![g_i^t]\!]$ to the AS.

C. Poisoned Gradients' Filtration

Upon receiving the encrypted local gradients $[\![g^t]\!]$ from all DNs, AS runs the poisoned gradients' filtration phase.

1) AS constructs an n-dimensional vector cs that holds the cosine similarity values where the i-th element in the vector is calculated by finding the cosine similarity between the local gradient g_i^t and the global weight w^{t-1} as

$$cs(i) = \frac{\langle w^{t-1}, g_i^t\rangle}{||w^{t-1}||.||g_i^t||}$$

Given that the received gradients are encrypted, an additional set of operations is performed by the AS to calculate the SIP between w^{t-1} and $[\![g_i^t]\!]$ and hence compute the cosine similarity between w^{t-1} and g^t without revealing the content of g^t. To calculate $\langle w^{t-1}, g_i^t\rangle$ using $[\![g_i^t]\!]$, the following steps are performed by the AS:

a) AS appends a value of (-1) to the vector w^{t-1} to ensure proper mask cancelation. Then it uses the split vector sv to split the updated w^{t-1} into two vectors $w^{t-1'}$ and $w^{t-1''}$ such that, the z-th element in both $w_i^{t-1'}$ and $w_i^{t-1''}$ is calculated as follow:

$$w^{t-1'}(z) = w^{t-1''}(z) = w^{t-1}(z) \;\; \text{if } sv(z) = 0$$
$$w^{t-1'}(z) = y_z,\; w^{t-1''}(z) = w^{t-1}(z) - y_z \;\; \text{if } sv(z) = 1$$

, where y_z is a random number.

b) AS uses its key $\mathcal{ASK}$ to generate its encrypted weight components $[\![w^{t-1}]\!]$ as

$$[\![w^{t-1}]\!] = \begin{bmatrix} w^{t-1'} M_1 N_1^{-1} & , & w^{t-1'} M_1 N_2^{-1} \\ w^{t-1''} M_2 N_3^{-1} & , & w^{t-1''} M_2 N_4^{-1} \end{bmatrix}^T$$

c) AS calculates $cs(i)$ as

$$cs(i) = \frac{[\![w^{t-1}]\!] \cdot [\![g_i^t]\!]}{||w^{t-1}||}$$

Note that g_i^t is normalized and hence $||g_i^t||$=1.

2) After AS calculate the cs vector, AS performs a min-max normalization to cs and then computes the poisoned gradient filtration threshold T as $T = mean(cs)$. Subsequently, each element in cs is compared with T. Specifically, for each DN_i, if $cs_i > T$, then DN_i will be classified as a node with a poisoned model update and its identifier will be added to $\mathbb{DN}^*$, the set of DNs with poisoned gradient updates.
At the end of this step, if the set $\mathbb{DN}^*$ is empty, then all the received gradients are considered to be benign, and AS moves forward to the aggregation phase. Otherwise, AS sends the list $\mathbb{DN}^*$ to all $\mathbb{DN}$ such that, subsection III-B will be executed again by the benign $\mathbb{DN}$ to update the masking vectors without including the pairwise masks shared members of $\mathbb{DN}^*$. This step is necessary so that the AS can retest updates sent by the set of benign nodes and ensure mask cancelation during the aggregation phase.

D. Secure Aggregation

After the poisoned gradient filtration phase is successfully executed with $|\mathbb{DN}^*| = 0$. AS will calculate the new global gradient as $\theta^t = \frac{1}{n}\sum_{i=1}^n g_i^t$, θ^t is calculated through the following steps:

1) AS calculates the aggregated encrypted local gradients $[\![g_{agg}^t]\!]$ as follow:

$$[\![g_{agg}^t]\!] = \sum_{i=1}^n [\![g_i^t]\!] = \begin{bmatrix} \sum_{i=1}^n N_1 A_i g_i^{t'} & , & \sum_{i=1}^n N_2 B_i g_i^{t'} \\ \sum_{i=1}^n N_3 C_i g_i^{t''} & , & \sum_{i=1}^n N_4 D_i g_i^{t''} \end{bmatrix}$$

Such that, g_{agg}^t consists of four components $\{[\![g_{agg,1}^t]\!],\; [\![g_{agg,2}^t]\!],\; [\![g_{agg,3}^t]\!],\; [\![g_{agg,4}^t]\!]\}$ where each components is a row vector of size p.

2) the AS uses its $\mathcal{ASK}$ to recover $g_{agg}^{t'}$ and $g_{agg}^{t''}$ as follows.

$$g_{agg}^{t'} = M_1 N_1^{-1}[\![g_{agg,1}^t]\!] + M_1 N_2^{-1}[\![g_{agg,2]}^t] = \sum_{i=1}^{n} g_i^{t'}$$

$$g_{agg}^{t''} = M_2 N_3^{-1}[\![g_{agg,3}^t]\!] + M_2 N_4^{-1}[\![g_{agg,4}^t]\!] = \sum_{i=1}^{n} g_i^{t''}$$

3) AS utilize the sv to merge $g_{agg}^{t'}$ and $g_{agg}^{t''}$ into g_{agg}^t such that, the z-th element in g_{agg}^t is calculated as follows.

$$g_{agg}^t(z) = g_{agg}^{t'}(z) \quad \text{if} \quad sv(z) = 1$$

$$g_{agg}^t(z) = g_{agg}^{t'}(z) + g_{agg}^{t''}(z) \quad \text{if} \quad sv(z) = 0$$

4) AS calculate new global update vector $\theta^t = \frac{1}{n} g_{agg}^t$, and broadcast θ^t to all $\mathbb{DN}$.

By the end of this phase, $\mathbb{DN}$ will update their local weights using the received global update θ^t as $w_i^{t+1} = w_i^t - \eta^t \theta^t$ where η^t is the learning rate for the t iteration. After the local weight update, another iteration in FL will begin. FL will halt when it finishes a preset number of iterations or the model converges to a predefined accuracy.

IV. DISCUSSION AND EVALUATIONS

A. Privacy Preservation of Consumers' Power Profiles

The local gradient g_i^t should not be accessed to external adversaries, the AS, or any other DN in the system to prevent model inversion and membership inference attacks that leak sensitive information about consumers' power profiles [9].

1) In our scheme, we employ a refined variant of the encryption scheme outlined in our prior work [12]. The security guarantees of this approach have been formally proved within the known ciphertext model [13]. Without access to the master key set $\mathcal{MK}$, external adversaries as well as curious DNs cannot decipher the sensitive information contained in the local encrypted gradients.
2) A curious AS can exploit the SIP technique that enables inner product functionality over encrypted vectors as follows. AS can construct a malicious extraction vector $[1, 0, ..., 0]$, encrypts it using its key $\mathcal{ASK}$, and multiply the encrypted malicious vector by $[\![g_i^t]\!]$. As a result, AS can reconstruct the first element from encrypted gradient $[\![g_i^t]\!]$. To prevent this attack, we introduce the pairwise one-time masking technique based on IBC. This means that the AS can recover the masked element $g_i^t(0) + mv(0)$. For AS to eliminate the mask $mv(0)$, AS needs to collude with all the $n-1$ DN, which is not feasible. In [14], the authors provide a formal security proof demonstrating that the protection of masked power data is upheld when suitable mask sizes are chosen, and when the masks are generated using a pseudo-random function. Consequently, our scheme is shown to be effective in resisting collusion between AS and DN.

B. Validation of Mitigating Model Poisoning Attacks

This section provides an overview of our experiments, including the dataset, attack generation, model architecture, and performance metrics to validate that our scheme can mitigate the impact of model poisoning attacks. We detail our targeted poisoning attack methodology and evaluate the effectiveness of our privacy-preserving aggregation defense mechanism against different attack percentages.

Dataset Description. We utilized an authentic Smart Meter (SM) dataset produced from the Irish Smart Energy Trials [15]. We utilized a subset of 300 users from the dataset, where each SM device recorded electricity consumption readings at 30-minute intervals. To prepare the data for our classifier, we performed dimension reshaping. As a result, we obtained 321,315 sample data, with each sample consisting of a 1-day smart meter reading. All the data readings contained within the dataset represent authentic and legitimate consumer reports. The dataset is randomly split into training, testing, and validation with ratios 60%, 20%, and 20% respectively. In a FL setting, we have 30 DN, with the samples evenly distributed among them. As a result, each DN holds 10,710 training samples.

1) **Malicious record generation**. To address the difficulty of obtaining falsified readings from fraudulent consumers, we employ a reduction function, denoted as f_r, on the power consumption readings of each distribution node DN to generate a malicious dataset. The function $f_r(pw_{i,j}) = \beta[j] pw_{i,j}$ aims at reducing the power consumption reading pw_i by applying dynamically reducing the reading $pw_i[j]$ by a value controlled by the time $\beta[j]$, where $0 < \beta[j] < 1$.
2) **Target attack set $\mathbb{TAR}$ Generation (Model Poisoning Attack)**. A set of malicious (SM^*) belongs to different $\mathbb{DN}$ colludes with each other to report a malicious targeted power consumption set $\mathbb{TAR}$, which is a set of row malicious power consumption's data, with the intention of introducing a backdoor into the global model to cause misclassifications. For instance, when Att_{ratio} is set to 10% this means that 10% of $\mathbb{DN}$ contains the TAR within its power consumption records labeled as benign since it is received from connected SM.

Temporal Convolutional Network (TCN) Model [8]. TCN is a time series CNN-based deep learning model that shows an advantage in energy theft detection. After an extensive hyperparameter tuning process, we use the TCN model with the initial learning rate set to $lr = 0.1$ for both $\mathbb{DN}$ and AS. The local batch size was fixed at 512, with local iterations equal to 1, and a total of 150 training iterations for FL.

Performance Metrics. Two performance metrics are used to evaluate the performance of our model.

1) ACC: Denotes the test accuracy achieved with TCN-based detector, calculated as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP, and FN represent the true positives, true negatives, false positives, and false negatives, respectively.

a) ACC_{def}: Denotes the test accuracy achieved with our proposed privacy-preserving defense mechanism

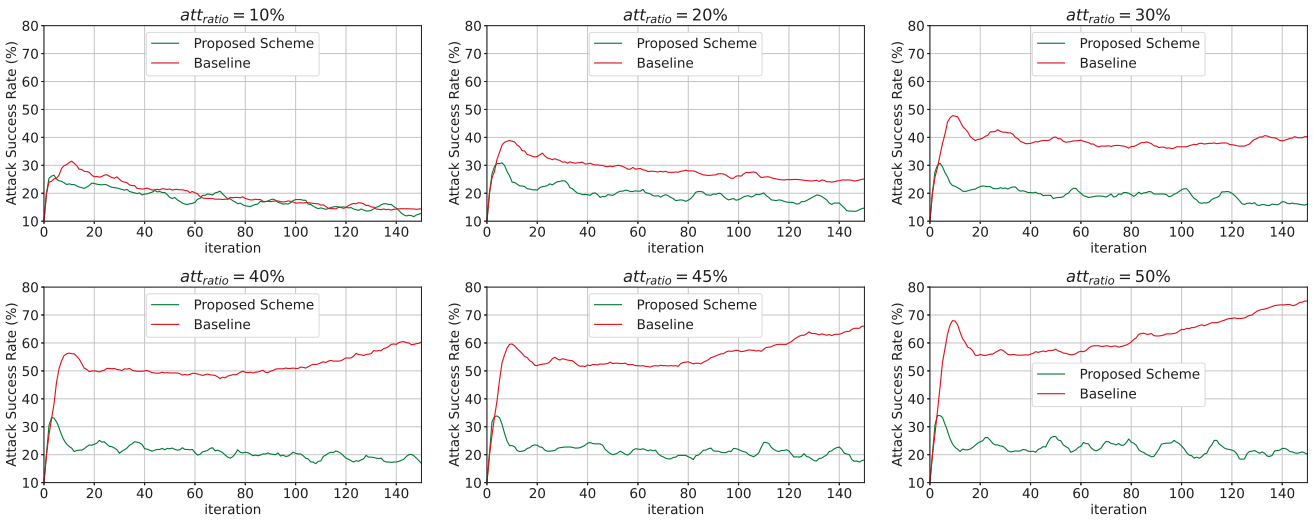

Figure 3: Impact of different attack ratio att_{ratio} on the ASR on $\mathbb{TAR}$ in presence and absence of our defense scheme

b) ACC_{no-def}: Denotes the test accuracy achieved without any defense (baseline model).

2) Attack Success Rate (ASR): The percentage of targeted examples within $\mathbb{TAR}$ that are incorrectly classified under the label desired by the colluding SM's (non-theft label).

Experimental Results. To evaluate the proposed privacy-preserving aggregation defense, we study the Att_{Ratio} parameter, which represents the percentage of injected DN^* from $\mathbb{DN}$. We employed various attack ratios att_{ratior}, ranging from the lower bound $att_{ratio} = 10\%$ to the upper bound of all defense schemes $att_{ratio} = 50\%$. We assessed our proposed scheme at different attack ratios, specifically $att_{ratio} \in \{10, 20, 30, 40, 45, 50\}$, and studied the impact of these attacks on the overall model accuracy and ASR.

Table II: Evaluating Overall Accuracy in the Presence and Absence of Privacy-Preserving Defensive Mechanisms

Attack %	10%	20%	30%	40%	45%	50%
ACC_{def}	0.934	0.932	0.925	0.924	0.925	0.919
ACC_{no-def}	0.933	0.922	0.911	0.896	0.893	0.890

Table II shows the TNN model's overall accuracy with and without our proposed scheme. With our scheme, we consistently maintain high stability and accuracy across various attack ratios. Even at the upper limit ($att_{ratio} = 50\%$), we achieve an impressive accuracy of 0.934%, thanks to the integration of cosine similarity filtering in our privacy-preserving aggregation scheme. In contrast, without our scheme, there is a noticeable impact on accuracy when Att_{ratio} is within the range of 40% to 50%. This is due to a high number of $\mathbb{DN}^*$ training over injected data $\mathbb{TAR}$, which affects the convergence of the global model. However, there is no dramatic loss of accuracy, as ACC_{no-def} reaches 0.890 at att_{ratio}=50%. This indicates that successful targeted attacks are still possible without significantly degrading overall accuracy.

In Figure 3, we examine the influence of different att_{ratio} values on the Attack Success Rate (ASR). With our scheme, we consistently observe low ASR, even at a high att_{ratio} =50%. This highlights the effectiveness of our scheme in countering backdoor attacks. Conversely, in the absence of our scheme, the ASR remains relatively low at att_{ratio} values of 10%, 20%, and 30%, this is because benign distribution nodes ($DN \notin \mathbb{DN}^*$) computing local gradients over benign samples contribute to mitigating backdoor effects during the local updates aggregation phase. However, as the att_{ratio} increases, particularly at 40%, 45%, and 50%, the ASR tends to significantly rise with each iteration, as the number of DN^* approaches the majority. This illustrates that in scenarios with high attack ratios, the global model starts to classify a significant portion of malicious power consumption in $\mathbb{TAR}$ as benign, which is indicative of a targeted attack.

V. Conclusions

In this paper, we proposed a novel scheme to empower Federated Learning based electricity theft detection with several security features including (1) preserving consumers' power consumption profiles through preventing model inversion and membership inference attacks in FL-based electricity theft detection and (2) mitigating model poisoning attacks where poisoned power consumption profiles may be falsely labeled as benign to create backdoors for electricity theft detection. Comprehensive assessments demonstrate that the proposed scheme can successfully achieve the desired security goals and achieves a substantial reduction in the success rates of poisoning attacks, even in scenarios where 50% of detection nodes train their local models with poisoned data, all while preserving customer privacy. Our method holds great promise for enhancing the security and trustworthiness of collaborative electricity theft detection systems in smart grids.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 2244371. The authors would also like to acknowledge the Alabama Power and Mobility (AMP) Center support in conducting this work. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or AMP.

REFERENCES

[1] Z. Yan and H. Wen, "Performance analysis of electricity theft detection for the smart grid: An overview," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–28, 2021.

[2] T. Ahmad, H. Chen, J. Wang, and Y. Guo, "Review of various modeling techniques for the detection of electricity theft in smart grid environment," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 2916–2933, 2018.

[3] M. I. Ibrahem, M. M. E. A. Mahmoud, F. Alsolami, W. Alasmary, A. S. A.-M. AL-Ghamdi, and X. Shen, "Electricity-theft detection for change-and-transmit advanced metering infrastructure," *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25 565–25 580, 2022.

[4] A. Alsharif, M. Nabil, S. Tonyali, H. Mohammed, M. Mahmoud, and K. Akkaya, "EPIC: Efficient Privacy-preserving Scheme with EtoE Data Integrity and Authenticity for AMI Networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3309–3321, 2018.

[5] M. I. Ibrahem, M. Nabil, M. M. Fouda, M. M. E. A. Mahmoud, W. Alasmary, and F. Alsolami, "Efficient privacy-preserving electricity theft detection with dynamic billing and load monitoring for ami networks," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1243–1258, 2021.

[6] M. Ashraf et al., "Feddp: A privacy-protecting theft detection scheme in smart grids using federated learning," *Energies*, vol. 15, no. 17, p. 6241, 2022.

[7] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[8] M. Wen, R. Xie, K. Lu, L. Wang, and K. Zhang, "Feddetect: A novel privacy-preserving federated learning framework for energy theft detection in smart grid," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 6069–6080, 2021.

[9] M. I. Ibrahem, M. Mahmoud, M. M. Fouda, B. M. ElHalawany, and W. Alasmary, "Privacy-preserving and efficient decentralized federated learning-based energy theft detector," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 287–292.

[10] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, "Privacy-enhanced federated learning against poisoning adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4574–4588, 2021.

[11] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "Shieldfl: Mitigating model poisoning attacks in privacy-preserving federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1639–1654, 2022.

[12] A. Alsharif, M. Nabil, M. M. Mahmoud, and M. Abdallah, "EPDA: Efficient and Privacy-Preserving Data Collection and Access Control Scheme for Multi-Recipient AMI Networks," *IEEE Access*, vol. 7, pp. 27 829–27 845, 2019.

[13] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure kNN Computation on Encrypted Databases," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 139–152.

[14] A. Unterweger, S. Taheri-Boshrooyeh, G. Eibl, F. Knirsch, A. Küpçü, and D. Engel, "Understanding game-based privacy proofs for energy consumption aggregation protocols," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5514–5523, 2018.

[15] Irish Social Science Data Archive., "Commission for Energy Regulation Smart Metering Project - Gas Customer Behaviour Trial, 2009-2010. [dataset]," https://www.ucd.ie/issda/data/commissionforenergyregulationcer/, [Online; accessed on October 25, 2023].