Sustainable and Lightweight Defense Framework for Resource Constraint Federated Learning Assisted Smart Grids Against Adversarial Attacks

Attia Shabbir[1], Habib Ullah Manzoor[2,3], Kamran Arshad[4,5], Khaled Assaleh[4,5], Zahid Halim[1], and Ahmed Zoha[2]

[1]Faculty of Computer Science, Ghulam Ishaq Khan Institute
[2]James Watt School of Engineering, University Of Glasgow
[3]Department of Electrical Engineering, University of Engineering and Technology
[4]Department of Electrical and Computer Engineering, College of Engineering and Information Technology, Ajman University
[5]Artificial Intelligence Research Centre, Ajman University

September 05, 2024

Sustainable and Lightweight Defense Framework for Resource Constraint Federated Learning Assisted Smart Grids Against Adversarial Attacks

Attia Shabbir[1], Habib Ullah Manzoor[2,3], Kamran Arshad[4,5], Khaled Assaleh[4,5], Zahid Halim[1], Ahmed Zoha[2*]

[1]Faculty of Computer Science, Ghulam Ishaq Khan Institute, Topi, Pakistan.
[2]James Watt School of Engineering, University Of Glasgow, Glasgow, G12 8QQ, United Kingdom.
[3]Department of Electrical Engineering, University of Engineering and Technology, Lahore, Pakistan.
[4]Department of Electrical and Computer Engineering, College of Engineering and Information Technology, Ajman University, Ajman, UAE.
[5]Artificial Intelligence Research Centre, Ajman University, Ajman, UAE.

Contributing authors: gcs2256@giki.edu.pk; h.manzoor.1@research.gla.ac.uk; k.arshad@ajman.ac.ae; k.assaleh@ajman.ac.ae; zahid.halim@giki.edu.pk; Ahmed.Zoha@glasgow.ac.uk;

Abstract

Energy networks face challenges in managing and securing the vast data generated by smart grids. Federated Learning (FL) offers a cost-effective, privacy-aware solution for model training, addressing customer privacy and data breach concerns. However, FL is susceptible to adversarial attacks, particularly data poisoning, which can degrade model accuracy. This study introduces a novel data poisoning attack and a mitigation framework for resource-constrained smart grids. We propose the Centroid Based Anomaly Aware Federated Averaging (CBAA-FedAvg) framework, which achieves a Mean Absolute Percentage Error (MAPE) of 2.7%, closely matching baseline performance. CBAA-FedAvg is a lightweight, sustainable solution that minimizes resource consumption through parameter quantization from 32-bit floating point to 8-bit fixed point and

dynamic clustering to reduce computational complexity. Additionally, an automatic stopping criterion is employed to optimize convergence, saving energy and time. The framework demonstrates remarkable resilience against data and model adversarial attacks, offering enhanced security and efficiency compared to state-of-the-art alternatives.

Keywords: Short-term load forecasting, Federated learning, Cyber security, Defense, poisoning attacks, False data injection

1 Introduction

Load forecasting utilizes historical load data to predict future load patterns. It plays a crucial role in developing optimal plans and making informed business decisions, such as setting real-time prices, managing load interruptions and control, and assessing available transmission capability [1]. Therefore, the accuracy of load forecasts directly impacts the cost and reliability of system operations, underscoring the importance of precise load forecasting techniques [2]. Over time, various algorithms such as time series-based mathematical models have been developed for load forecasting, incorporating diverse approaches. For instance, Autoregressive Integrated Moving Average (ARIMA) [3] is commonly used in load forecasting, relying on predefined models and historical data patterns. However, these methods often exhibit limitations in terms of robustness and adaptability.

In recent years, the power grid has experienced a rapid influx of new technologies and a surge in data generation. This growth has had a profound impact on utilities and system operators, requiring them to adapt their approaches to load forecasting [4]. To address this challenge, energy service providers can harness the power of artificial intelligence (AI) technologies to analyze the vast amount of available data. By utilizing sophisticated machine learning (ML) models, they can enhance power consumption prediction, optimize profits, implement tailored energy strategies and conduct pattern analysis without the need for explicit instructions[5]. While this approach has shown promising results in terms of enhancing adaptability and accuracy, it is crucial to address privacy concerns. As more data is collected and analyzed, there is a need to ensure the responsible and secure handling of sensitive information.
Recently, several instances of information leakage have underscored the significant privacy risk associated with traditional approaches on edge devices for load forecasting. This is due to the conventional practice of uploading all sensitive data to the cloud for model training, which is commonly seen in traditional ML methods[6, 7]. Energy consumption data, being particularly sensitive, has the potential to disclose personal information that could be exploited for malicious purposes. Household energy consumption data may reveal periods of absence at home, allowing third parties to infer household occupancy rates and the sleep/wake-up times of residents, while warehouse energy consumption data might expose additional details about production quantities or processes[8].

In this particular situation, Federated Learning (FL) [9], an approach proposed by H. Brendan McMahan in 2017, arises as a reliable alternative for distributed computing that prioritizes privacy. FL achieves this by shifting the computation to the owners of the data, enabling the collaborative training of models across devices. Importantly, this approach eliminates the need to transfer data to a central repository for model training[10]. Despite the advantages of privacy preservation, FL also offers efficiency in terms of communication resource utilization and exhibits superior scalability [11]. In recent times, researchers have shown significant interest in exploring the potential benefits of FL in various domains of the smart grid such as short-term load forecasting [12].

Indeed, despite the promising potential of FL in preserving privacy, recent studies have drawn attention to situations where FL may fall short in providing sufficient privacy guarantees. A notable example is the discovery by researchers that the original raw data can be reconstructed by analyzing the shared gradients of the model during the iterative FL process[13]. Additionally, because of the distributed nature of FL, it is susceptible to different faults/attacks. These faults occur when client nodes behave arbitrarily, which could be a result of adversarial manipulations[14, 15]. Moreover, challenges also persist, particularly regarding its susceptibility to data attacks. Its inherent lack of direct data access presents an opportunity for attackers to exploit data vulnerabilities. The focus of these attacks is to compromise the integrity of the models. Exploiting this vulnerability could result in maliciously amplified forecasting errors, upsetting the balance in power supply-demand matching. Such imbalances may lead to severe power crises, affecting the efficiency, reliability, and economics of power system operations and planning.

In addition to these vulnerabilities, FL has introduced a significant challenge: the communication bottleneck. In practice, training a ML model entails multiple iterations across different devices. During each local iteration, device's processor retains a high-dimensional vector that encompasses both the 32-bit floating-point weight values and the real values of the model's activation functions. This high bit-precision leads to significant energy consumption, substantial storage requirements, and millions of Floating-Point Operations Per Second (FLOPS) [16]. For instance, transmitting model gradients from a single entity using the GPT-3 model demands a staggering 2.8 terabytes of data, leading to substantial communication overhead [17]. Moreover, in practical scenarios, devices like smartphones and wearables may not have enough memory to support full-precision model training and clients often have varying computational resources, emphasizing the necessity for developing more sophisticated, constraint-aware strategies. Previous work on FL has addressed security and optimal communication separately. However, combining defense mechanisms with communication-efficient techniques may be impractical due to the cumulative effects on time complexity, signaling overhead, and potential accuracy degradation[17]. Hence, it is crucial to develop fault-tolerant mechanisms for FL that can handle vulnerabilities, while ensuring strong generalization performance and efficient communication simultaneously [18]. In this paper, we introduce a lightweight anomaly detection framework called Centroid Based Anomaly Aware Federated Averaging (CBAA-FedAvg) designed to mitigate the adversarial effects caused by anomalous clients in energy

networks. This framework is capable of simultaneously addressing data and model adversarial attacks while utilizing minimal device resources.

1.1 Contributions

In this work, we made significant contributions to the field of FL-aided smart grids. Firstly, we introduced a novel data poisoning attack specifically designed for FL systems. Additionally, we developed a computationally efficient and robust framework capable of detecting and mitigating both data and model posing attacks simultaneously in resource constraint environment. This unified lightweight framework eliminates the need for separate frameworks for data and model poisoning. The main contributions of our work are:

1. Proposed a deep learning model for forecasting in distributed energy grids.
2. Introduced a novel data poisoning attack called "Adversarial flipping Attack," undetectable by traditional statistical methods like mean, median, and z-score[19]. This attack reveals a new vulnerability in FL systems, underscoring the necessity for more robust defense mechanisms.
3. Developed a model flipping attack for poisoning the models in the FL setting.
4. Presented CBAA-FedAvg, a novel framework capable of simultaneously mitigating the effects of both data and model poisoning attacks. This unified lightweight system enhances the overall security and robustness of FL-aided smart grids.
5. Evaluated the computational resource utilization of the proposed CBAA-FedAvg alongside two existing state-of the-art methods.

1.2 Related Work

This section outlines the relevant literature concerning load forecasting and privacy-preserving techniques within FL.

1.2.1 Non-Fedrated Load Forecasting

Load forecasting techniques, including AI and ML algorithms such as artificial neural networks, ARIMA, non-parametric regression, and support vector machines, have gained popularity in smart grids [20, 21]. Long Short-Term Memory (LSTM), a specific algorithm, has shown promise in short-term load forecasting but has not yet achieved desired levels of accuracy [22, 23]. Efforts to improve energy consumption forecasting have included the use of sequence-to-sequence LSTM models for one-minute resolution data and genetic algorithms for hyperparameter tuning. However, challenges remain in finding the optimal combination of parameters [24, 25]. The generalization of data-driven models to new datasets is also a significant concern [22]. To address these challenges, researchers have explored data aggregation through clustering and pooling techniques, aiming to reduce variance and overfitting [26, 27]. However, centralized load forecasting methods raise privacy concerns due to the transmission of detailed consumption data over networks [28]. Various approaches have been proposed to protect user identity, such as assigning shared serial numbers based on geographic proximity, but this method complicates individual client treatment due to anonymity

[29]. To sum up, none of the previously mentioned papers effectively address both user privacy and prediction accuracy together, but FL shows promise in overcoming these challenges.[11].

1.2.2 Federated Load Forecasting

Because FL effectively balances privacy and prediction accuracy, the study in [12] was one of the first to apply FL to load forecasting by training an LSTM model using a real-world Texas load consumption dataset, leading to satisfactory forecasting performance. The FL-based framework proposed by [30] effectively forecasts load while assessing grid-specific metrics, ensuring satisfactory performance in load swings and curtailment. Likewise, in [31], the authors claims that they successfully attained the lowest MAPE among existing algorithms for one-step ahead forecasting in both US and Australian datasets by using variational mode decomposition,federated k-means clustering, and SecureBoost for short-term load forecasting. In [32], the authors introduce a personalized federated approach to tackle the challenge of overfitting in load forecasting models for both individual and multiple consumers. The paper acknowledges the constraints posed by limited datasets and privacy concerns, making it challenging to effectively train complex models. Several other studies [33, 34] have also employed FL in the domain of load forecasting, highlighting its potential and benefits. While aforementioned studies have made strides in leveraging FL to preserve privacy by avoiding the need for raw data sharing, they have overlooked critical privacy and security concerns. Additionally, these studies have not extensively investigated the vulnerability of FL to various faults and attacks. Attackers can intentionally contaminate the client's information whether through direct or indirect means with the aim of modifying the parameters of the target model [14, 35], thereby impacting its performance.

1.2.3 Adversarial attacks in FL

Recent studies have highlighted the vulnerability of FL to adversarial attacks, which can be classified into two categories: utility-centric threats and privacy-centric threats [28]. Our focus is utility-centric threats which involve malicious actions that tamper with data or models shown in figure 1, resulting in compromised accuracy [36].

1. **Model Poisoning Attacks:** Model poisoning attacks in FL can picturize as either model tampering or model replacement. In model tampering attacks, adversaries manipulate the local model parameters on participating devices by injecting deliberate perturbations or biases into the model updates [37, 38]. This malicious manipulation compromises the integrity and accuracy of the global model, potentially leading to unreliable predictions. On the other hand, model replacement attacks involve adversaries aiming to substitute the legitimate global model M' with a malicious one. By manipulating the model replacement process, attackers can influence the aggregation step during model updates. As a result, the compromised global model M'_c may exhibit inaccuracies or vulnerabilities, undermining the overall performance and trustworthiness of the FL system [39].

   Mitigation mechanism: To mitigate the security risks posed by poisoning

attacks and uphold the integrity of the global model in load forecasting, robust measures are essential. Several studies have employed rejection mechanisms based on error rates and loss functions to detect and exclude potentially malicious models [40]. For example, the proposed Fed-SAD [41] method enhances load forecasting accuracy and robustness by employing secure aggregation. It detects and mitigates poisoning attacks using Gaussian distribution functions to assess similarity and distance. However, its applicability may be limited by challenges in handling complex or non-Gaussian data distributions, as well as susceptibility to outliers and assumptions of linearity between variables. The framework purposed by [42] utilizes gradient quantization through the Sign Stochastic Gradient Descent (SignSGD) algorithm. Clients transmit only the 'sign' of the gradient to the control center after local model training. This approach effectively mitigates Byzantine threats and surpasses conventional FedSGD models. However, potential challenges include quantization errors introduced during gradient quantization and the necessity to ensure robustness against more sophisticated attacks. Likewise, [43] proposed Cyber-Secure Federated Deep Learning (CSFDL) method aims to enhance safe load forecasting. It employs model averaging to reduce the disclosure of gradient updates to adversaries in Federated Deep Learning (FDL). Additionally, using dropout limits neuron activations, weakening information leakage. However, selecting appropriate hyperparameters, such as the dropout rate or averaging strategy, can be challenging, necessitating extensive experimentation and validation for optimal results. Along the same line, several clustering-based frameworks have been proposed in the literature to address the challenge of model poisoning attacks in FL. One method, described in [44], involves dividing the set of weights into two clusters and eliminating the smaller cluster, assumed to contain adversarial weights, before model aggregation at the server-side. Another approach, presented in [14], utilizes Hidden Markov Models to detect poisoning attacks in FL. This method helps identify attackers before the model aggregation stage. The ZeKoC approach, proposed in [45], enables the server to autonomously split and merge weight clusters for weight selection and aggregation in FL. Analysis guarantees convergence, but the computational cost associated with this approach is high. In [35], a modified version of FedAVG is introduced to mitigate the impact of adversarial backdoors in load forecasting using FL. This modification eliminates neural network layers that significantly differ from the rest of the models. However, it is important to consider the potential challenges in terms of computational expense and the assumptions made, which may limit the effectiveness and generalizability of these approaches in real-world scenarios.

2. **Data Poisoning Attacks:** Data Poisoning attacks, on the other hand pose a significant threat to ML models, including those used in FL. In data Poisoning attacks, an adversary manipulates the training data on a specific number of devices participating in the learning process, aiming to compromise the accuracy of particular clients or the overall global model. There are two primary approaches that adversaries employ to contaminate data within an FL model. Firstly, they may directly inject false data into targeted devices to introduce bias or misleading

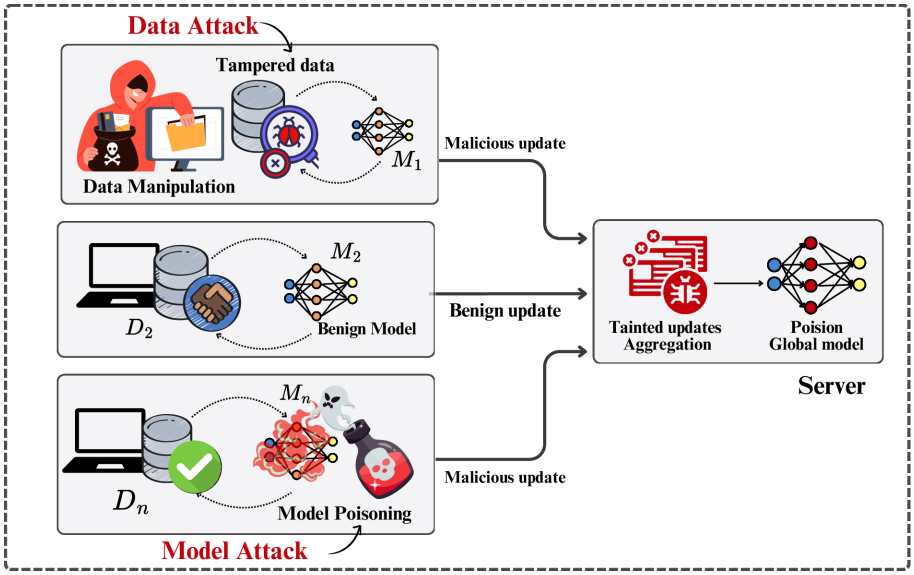


Fig. 1: Poisoning Attacks in Federated Learning

information. Secondly, adversaries may poisoned data through other devices by manipulating data on one or more devices, which then affects the overall model performance [46]. Such manipulation can lead to vulnerabilities and inaccurate predictions.

Mitigation mechanism: In the field of false data injection attack (FDIA), several approaches based on FL have been proposed. Reference [47] pioneering FL-based scheme detects FDIA in solar PV systems without raw data sharing, ensuring privacy compliance. Experimental findings validate its efficacy, marking the first FL application in power electronics literature. One approach [48] utilizes FL for generating a global model and adding artificial noise to model parameters to ensure data and differential privacy. Simulations on the IEEE 30-bus system confirm the trade-off between accuracy and privacy preservation. Another approach [49] combines graph neural networks and LSTM layers for FL based FDIA detection. This method leverages local correlations and temporal patterns, ensuring flexible and efficient training while preserving client privacy. Furthermore, [50] utilized multi-head self-attention and FL, a secure federated deep learning method integrates Transformer, FL, and the Paillier cryptosystem to safeguard privacy and security in FDIA detection. Reference [51] introduce an FL framework for collaborative anomaly detection model training, enhancing generalization, and propose an attention mechanism-based CNN-LSTM model for accurate anomaly detection. Lastly,[52] introduced an efficient cross-silo FL scheme that ensures privacy through a double-layer encryption, secret sharing, and parallel computing approach. This scheme facilitates client dropout and rejoining during training. Similarly, several related works have utilized FL to detect FDIA in various fields.

1.2.4 Optimized Load Forecasting

In FL, gradients and model parameters are usually represented as floating-point numbers. Frequent interactions between the server and clients to share high-dimensional model updates can result in low communication efficiency in FL setups. Therefore, to address this communication bottleneck, efficient communication is often achieved through well-established model compression [53] techniques that reduce the computational and memory requirements of neural networks. These include sparsification[54], which reduces the number of non-zero parameters; pruning[55], which selectively removes less important connections or nodes; and client selection, a strategy that aims to identify a subset of clients to participate in the gradient update process.

Another prominent approach, which we emphasize in this paper, is the quantization of neural networks (NN). Quantization techniques aim to decrease the precision of NN parameters from full-precision (32-bit floating point, or FP32) to lower-bit representations [56]. These techniques are commonly used to streamline both the inference and training processes of the global model, thereby improving the overall communication efficiency of the FL system. [57] proposes a non-uniform quantization based method to address the unbalanced distribution of residential load for one-hour ahead load forecasting. [42] proposed framework utilizes a gradient quantization technique grounded in the Sign Stochastic Gradient Descent (SignSGD) algorithm. In this approach, clients transmit only the sign of the gradient to the central server following local model training, significantly reducing communication overhead in a FL environment. However, there is limited work in the literature that simultaneously addresses security and communication efficiency. The FedRLA framework [37] is an energy-efficient approach that requires sharing only a single neural network layer between devices and the server during training. Similarly, CMULA-FL [58] reduces communication costs by compressing model updates, uploading only large norms, and compensating for errors by carrying them over to the next epoch, thereby enhancing model utility. However, the reduced amount of shared information could potentially result in suboptimal model updates or a slower convergence rate during training.

1.3 Gap Analysis

1. The majority of existing research in FL has primarily focused on detecting data attacks. However, there has been limited attention given to the development of sophisticated data tampering attacks within the FL framework, particularly in the context of energy load forecasting. This highlights a significant research gap that needs to be addressed.
2. Additionally, the literature lacks a unified security framework that can effectively address both data and model attacks in the FL setting. By expanding the scope to include different types of attacks, a more comprehensive and robust security system needs to be developed to protect FL systems from a wider range of threats.
3. Furthermore, existing research in FL often operates under the assumption that IoT devices possess unlimited resources, such as computational power and energy [59–62]. However, in real-world scenarios, FL nodes frequently face constraints in terms of energy, network connectivity, and computational capabilities. Hence, in

resource-constrained environments, the development of computationally efficient methods for detecting and mitigating attacks in FL becomes essential. This would not only improve the scalability of the FL system but also enhance its practical implementation, particularly in scenarios where energy efficiency is a critical factor.

1.4 Paper distribution

The paper is structured as follows: Section II begins with an introduction to FL. In Section III, the threat modeling section discusses proposed attacks, focusing on data and model poisoning. Next, in Section IV, the proposed mitigation approach, CBAA-FedAvg, is presented, followed by Section V, which presents experimental evaluation and results. Finally, in Section VI, a comparative analysis is conducted, and the paper concludes with key findings and implications.

2 Federated Learning-aided energy networks

Federated learning-aided energy networks represent an innovative paradigm that integrates the strengths of FL with energy system enhancing efficiency, privacy, and sustainability through distributed intelligence. FL holds significant potential, particularly in the efficient analysis and management of energy data while prioritizing privacy. By preventing direct server access to participant data, FL promises advancements in maintaining a balance between data utility and privacy protection, optimization, and overall energy system operation.

2.1 An Insight into FL: Collaborative ML for Distributed Data

FL, as a form of encrypted distributed ML, offers a distinct advantage over previous model training approaches. Unlike traditional methods, it does not necessitate the physical transfer of raw data to a centralized server. This unique characteristic makes FL an effective solution for addressing privacy concerns. In practical application scenarios, it is assumed that there are N users, denoted as C_i where $i \in \{1, 2, \ldots, n\}$ each owning their own respective database, represented as $\{D_1, \ldots, D_n\}$. It is important to note that these users cannot directly access the data of other individuals to augment their own datasets. Figure 2 provides a high-level overview of the five-step fundamental design of FL.

1. FL process begins with the distribution of an initial foundational model. This model is sent from a central server to all edge devices that are partaking in the FL arrangement. This initial model serves as the starting point for training on each individual device, allowing for the collaborative learning process to commence.
2. In the next phase, edge devices C_i take the initial model and iteratively enhance it by incorporating their locally available data D_i The objective is to solve the optimization problem $\min_{M_i} L_i(D_i, M_i)$, where L_i represents the empirical loss function specific to each edge device. This iterative process allows the edge devices to adapt the model to their specific attributes and characteristics. Consequently, these locally adapted models function as personalized versions of the generic model, tailored to accommodate the unique characteristics and data of each individual edge device.

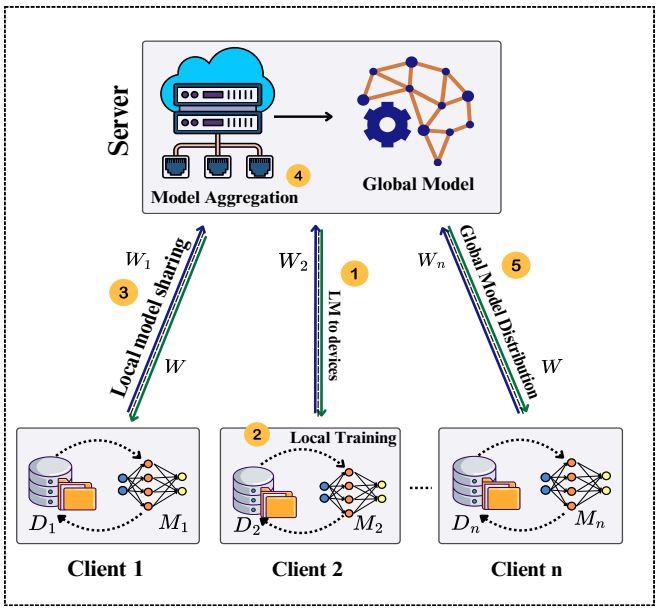


Fig. 2: Overview of Steps in Federated Learning (FL)

3. Following the creation of their trained local models $\{M_1, \ldots, M_n\}$, edge devices transmit key information of their respective models to the central server, encompassing the refined weights and biases $\{w_1, \ldots, w_n\}$. This communication stage lays the groundwork for collaborative learning, enabling the central server to collect and aggregate the updated model parameters from the edge devices.
4. In the fourth step, the central server executes an aggregation process using common techniques to merge the weights and biases received from all (or most) edge devices. This aggregation process yields a refined global model M' that encapsulates the collective insights and knowledge (W) from the local models.

$$W = \frac{1}{n} \sum_{i=1}^{n} w_i$$

5. The final phase involves distributing the newly generated global model M' back to all participating edge devices. This step signifies the completion of a communication round between the central server and the edge devices.

Steps 2–4 iteratively occur for multiple rounds, as FL algorithms typically require several rounds to converge. At the conclusion of each round, clients evaluate the model's performance using specific metrics such as Mean Absolute Percentage Error (MAPE), which are then transmitted to the server. It's noteworthy that MAPE is independent of system capacity and the unit of measurement[11]. These metrics provide the server with a global perspective on the distributed training's performance, guiding

decisions on when to conclude the process. For the aggregation, there are various policies or algorithms that can be used , with the two most widely-adopted approaches being Federated Stochastic Gradient Descent (FedSGD)[63] and Federated Averaging (FedAvg)[64]. In general, FedSGD averages the locally computed gradient at each step of the learning phase, while FedAvg averages local model updates when all clients have completed training their models. However, as mentioned earlier, irrespective of the approach employed, FL is susceptible to various privacy and security threats, which are elaborated upon as follows.

3 Threat Modeling

Threat modeling is a structured approach utilized to identify potential risks and vulnerabilities within a system by examining its various components [65]. This process entails identifying potential attackers, understanding their motivations, analyzing the methods they might employ, and assessing the potential consequences if they were successful in compromising the system's confidentiality, integrity, or availability [66]. In the realm of FL, two primary categories of adversarial attacks are recognized: privacy-centric attacks and utility-centric attacks, which include data and model poisoning. [65]. In this paper, our focus lies in mitigating data and model poisoning threats within FL, particularly for energy load forecasting. Prior to presenting our proposed defensive approach, we examine two distinct threat models within the domain of federated load forecasting.

3.1 Data Poisoning Attack:

The quality and structure of data play a critical role in the training of ML models. Poorly structured data or adversarial samples can lead to inaccurate or unreliable predictions. In the context of load forecasting, where inputs are derived from sources like smart meters that transmit readings to the cloud, there is potential for adversaries to compromise the data [8]. Despite the importance of this issue, the topic of data poisoning attacks on load forecasts has received little attention in the existing literature. FL is generally considered resilient to data attacks [67]. However, as highlighted in [67], the attack was limited to local clients, with its minimal impact on the global model.

To address this gap, this section proposes a novel data poisoning attack called the Adversarial Flipping Attack (AFA). The goal of AFA is to reveal vulnerabilities in load forecasting by affecting the global model while making minimal statistical changes to the local model.

3.1.1 Adversary's Model

In our study, we investigate a scenario in FL where a subset of participants displays malicious behavior or is controlled by a malicious adversary. We denote the percentage of these malicious participants among all FL participants as P, representing $m\%$ of the total participants.

3.1.2 Adversary's Target

The primary objective of the adversary is to compromise the efficiency and integrity of both the local client's model and the global model through an untargeted attack using the AFA on energy data in FL. The adversary seeks to manipulate predictions made by the local model, intending to deceive and impact the decision-making processes of the local client. Additionally, they aim to manipulate the global model to influence the participation of other clients in the FL system.

3.1.3 Adversary's Proficiency

To evaluate the influence of the attacker's knowledge, two types of attack scenarios can be considered: white-box attacks and black-box attacks [68]. In a white-box attack, the attacker has complete knowledge of the system, including access to the training set and the test set of i^{th} client, the model structures, and even the model updates. This comprehensive knowledge provides the attacker with a deep understanding of the inner workings of the model. However, this level of access may not always be feasible in real-world scenarios. To address more realistic scenarios, we investigated black-box attacks. In a black-box attack, the attacker lacks explicit knowledge of the model information, including the model structures or parameters. However, they can utilize the training set D_i^{tr} to train a local model M_{i} and exploit this corrupted training set to craft the attack on i^{th} client.

In our scenario, the attacker's strategy relies on daily energy data to shape their attack. Their primary capability involves tampering with datasets from the i^{th} client, which includes both the training set D_i^{tr} and the test set D_i^{test}. However, the attacker operates unaware of the specific model architecture and the functioning of the aggregator. Additionally, they are deprived of access to the global model M' and its explicit data, which is collected before the FL process commences.

Our assumption is that the aggregation process and all model updates are benign, emphasizing that the attacker's focus is solely on tampering with the client's data. This targeted data manipulation is intended to exploit vulnerabilities in the trained model, influencing its behavior to fulfill the attacker's malicious objectives.

3.1.4 Adversary's Conduct

The adversary strategically employs the Adversarial Flipping Attack (AFA) to manipulate energy data, aiming to minimize the likelihood of detection through statistical analysis. In this scenario, the attacker targets a specific client or system possessing a database with training data (D_i^{tr}) and a test set (D_i^{test}). At the designated time t, the attacker gains unauthorized access to the database when the targeted client's system is operating under normal conditions. They retrieve the D_i^{tr} and D_i^{test}, used for training and testing the local model simultaneously at time t.The attack specifically targets the daily values of the energy data allows the attacker to potentially scrutinize and alter it to create adversarial inputs capable of misleading or distorting the load prediction outcomes.

To execute the AFA attack, the adversary applies a transformation function, denoted as F , to the original daily values X_i. This function effectively mirrors and flip the

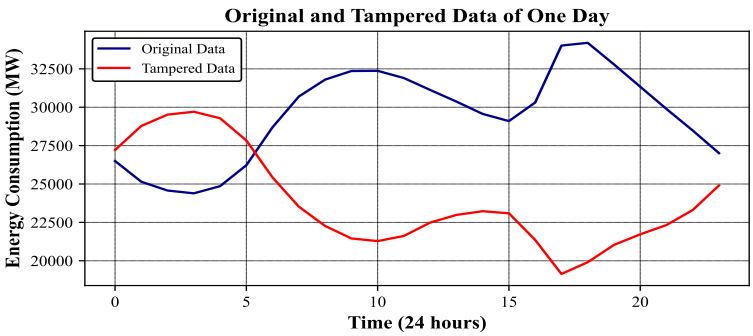


Fig. 3: Original and Tampered Data of One Day's reading

data points. For example, if the original data exhibited an increasing trend, the flipped data would show a decreasing trend, and vice versa. Figure 3 provides a visual representation of this process, showcasing the original and tampered data for a single day's reading. Mathematically, the transformation function F can be expressed as:

$$Y_i = F(X_i) \tag{1}$$

Where Y_i represents the tampered (flipped) value at time t.

$$Y_i = \begin{cases} Y_{i-1} - (X_i - X_{i-1}) & \text{if } X_i > X_{i-1} \\ Y_{i-1} + (X_{i-1} - X_i) & \text{if } X_i \leq X_{i-1} \end{cases} \tag{2}$$

The value in the flipped data at any position i depends on the value at the previous position $(i-1)$ in the flipped data and the difference between the corresponding elements in the original data.

1. If the current value is greater than the previous value, the new value is the previous new value minus the difference between the current and previous values.
2. If the current value is less than or equal to the previous value, the new value is the previous new value plus the difference between the previous and current values.

The purpose of this flipping attack is to create a manipulated version of the energy data that maintains almost a similar overall appearance to the original data, making it visually difficult to distinguish as compare the with other data manipulation techniques such as scaling or random noise insertion [67].

3.2 Model Poisoning Attack

In this specific threat model, we consider a scenario where a malicious attacker compromises a subset of clients within a FL framework. The attacker could be an insider with authorized access or an outsider who has infiltrated the system. Their main objective is to manipulate the global model, denoted as M', which is learned by the other clients in the FL framework. It is important to note that the attacker

lacks access to historical load records, preventing them from acquiring the necessary dataset for their purposes. However, we assume that the attacker has only access to the local models used for load forecasting in this particular scenario.

Such attacks on models can be broadly categorized into two types: fully poisoning attacks and partially poisoning attacks [35]. In a fully poisoning attack, the adversary aims to replace the entire model with maliciously crafted components, resulting in a complete compromise of the model's integrity. On the other hand, partially poisoning attacks target specific subsets of the model's components, such as selected neurons, with the attacker focusing on a limited portion of the model. In this paper, our focus is specifically on a partial adversarial attack, which has the capability to impact a specific layer of the model. This type of attack showcases the attacker's ability to compromise the integrity of the model by manipulating selected layers while keeping the rest of the model unchanged. To execute this attack, the attacker employs a model flipping technique that specifically targets the weights of certain layers in the neural network. These targeted weights are flipped by multiplying them by -1 [35, 37, 69].

$$w_{ij}^{'} = -1 \times w_{ij} \tag{3}$$

where $w_{ij}^{'}$ represents the flipped average weight for the j-th layer in the i-th model, where w_{ij} denotes the actuall average weight of the model. Figure 4 provides an illustration of weight density comparison of such an attack, where only the selected layers of the model is targeted.

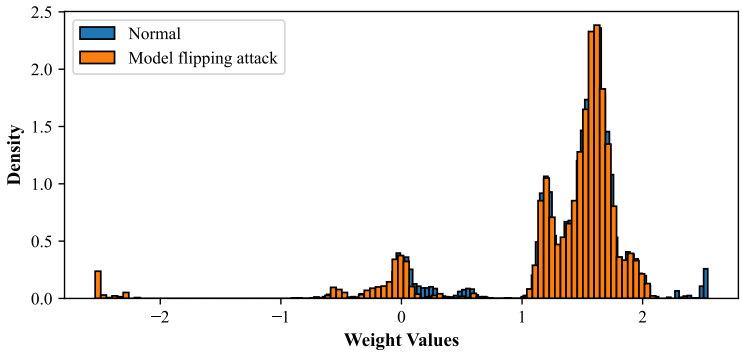


Fig. 4: Weight Density Comparison: Normal vs. Attacked Weight Distribution of a Local Model

The significance of these findings underscores the need to implement effective countermeasures and defense strategies for mitigating the impact of local data poisoning and model attacks in the domain of federated load forecasting. In the subsequent section, we will introduce a mitigation strategy called Centroid-based Anomaly-Aware (CBAA) FedAvg. This strategy has been purposefully designed to accurately detect

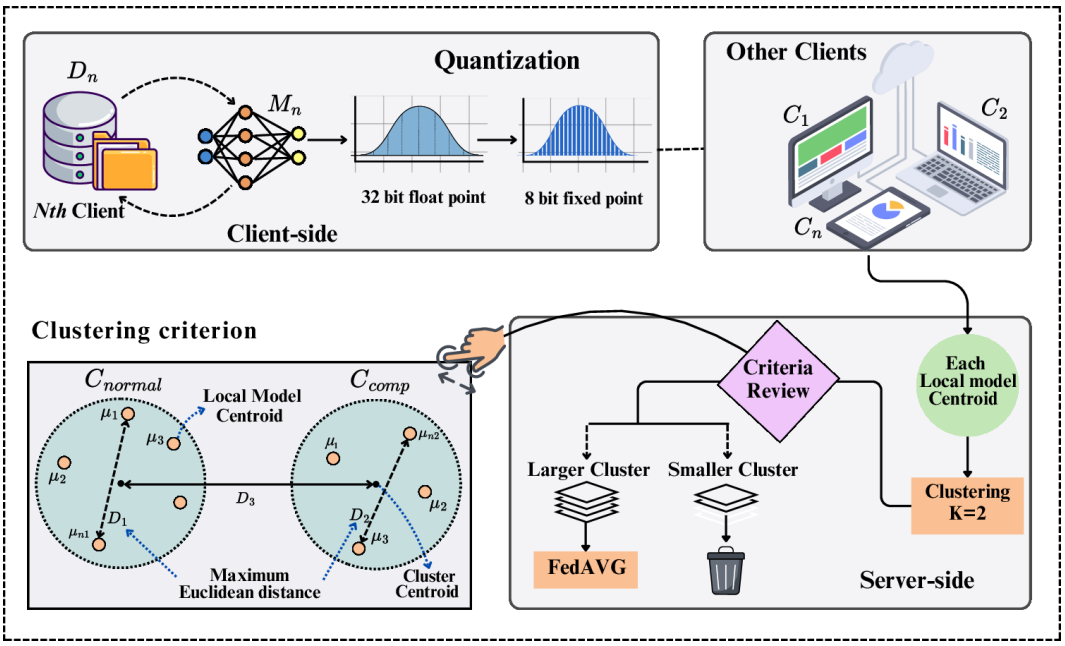


Fig. 5: Framework and visual representation of clustering criterion of centriod-based anomaly aware Fed-AVG (CBAA-FedAVG).

and address instances of data and model poisoning attacks, ensuring the integrity of the FL process in resource constraint environment.

4 Mitigation Approach: CBAA-FedAvg

The Centroid-based anomaly-aware FedAvg (CBAA-FedAvg) utilizes the aggregation framework, as shown in Figure 5. In this adapted version of FedAVG, quantized models weights are grouped using specific clustering criteria. K-means clustering is applied to the mean of the average weights across all layers of each model. The main objective of this framework is to aggregate models while excluding compromised clients. By doing so, the models from the remaining normal clients can be utilized for aggregation, leveraging their collective knowledge. This approach helps mitigate the negative impact that compromised models may have on the global model.

The algorithm for aggregation is outlined in Algorithm 2 and encompasses the subsequent steps:

1. **Weights Quantization**
   During the CBAA-FedAvg process, updates from clients' local models, represented as weights w_{ij} for the j-th layer in the i-th model, are sent to the server. In most implementations, network parameters are generally represented using 32-bit FP values. These full precision weights are quantized to lower bit representation of 8-bit fixed point values, retaining the essential characteristics of the original tensors.

The quantization function clamps an input value w_{ij} within a specified range defined by parameters n and m.

$$\bar{w}_{ij} = \text{Quantize}(w_{ij}, n, m) \tag{4}$$

For our specific case, with $n = 1$ and $m = 7$, the clamped range is from $-2^{(n-1)}$ to $2^{(n-1)} - 2^{(-m)}$. The clamped value is then scaled to a discrete set of levels using a quantization precision of 7 bits. After rounding, the function rescales the value back to the original range, enabling fine-tuned quantization. Quantization can be outlined and executed using the algorithm shown in 1, where $\bar{w}_{ij}$ represents the quantized weight.

Algorithm 1 Quantization of Weights: `Quantize`$(w_{ij}, n = 1, m = 7)$

Require: w_{ij}: Weight value to be quantized, n: Clamping range exponent, m: Precision factor

Ensure: $\bar{w}_{ij}$: Quantized weight value

1: Define clamping boundaries: $\min_{\text{val}} \leftarrow -2^{(n-1)}$, $\max_{\text{val}} \leftarrow 2^{(n-1)} - 2^{-m}$
2: $w_{ij} \leftarrow \max(\min_{\text{val}}, \min(\max_{\text{val}}, w_{ij}))$
3: $w_{ij\_\text{scaled}} \leftarrow \text{round}(w_{ij} \cdot 2^m)$
4: $\bar{w}_{ij} \leftarrow w_{ij\_\text{scaled}}/2^m$
5: **return** $\bar{w}_{ij}$

2. **Weights Averaging**
In the subsequent step, for each model i, we compute the average of the quantized weights for each layer as follows:

$$\begin{aligned} W_{i1} &= \text{Avg}(\bar{w_{i1}}), \\ W_{i2} &= \text{Avg}(\bar{w_{i2}}), \\ &\vdots \\ W_{ij} &= \text{Avg}(\bar{w_{ij}}), \quad \text{for } j = 1, 2, \ldots, l. \end{aligned}$$

Here, W_{ij} represents the quantized average weight for the j-th layer in the i-th model, with l denoting the total number of layers in the model.
Next, we calculate the centroid of the quantized average weights for each model. This centroid is a single mean value that reflects the overall weight pattern of the model:

$$\mu_i = \frac{1}{l} \sum_{j=1}^{l} W_{ij}, \quad \text{for } i = 1, 2, \ldots, n. \tag{5}$$

Here, μ_i represents the centroid of the quantized average weights across all layers for the i-th model, where n is the total number of models.

3. **K-means Clustering for Anomaly Detection:**
   In order to detect anomalies in a set of local models, each characterized by a mean weight value $\{\mu_1, \mu_2, \ldots, \mu_n\}$, we applied the K-means clustering algorithm. The goal is to group these models into k clusters based on the similarity of their μ_i values. The algorithm works by minimizing the distance between each model's μ_i and the centroid (representative point) of its assigned cluster.

   In our specific context, we set $k = 2$, which results in two clusters:

   - $\mathbf{C_{normal}}$: This cluster is assumed to contain the majority of models from honest clients.
   - $\mathbf{C_{comp}}$: This cluster is expected to contain models from a smaller number of clients whose data integrity has been compromised.

4. **Clustering Protocol and Viability:**
   To determine whether to include or exclude a client in the aggregation process, we apply the following clustering criterion:

   - Calculate the maximum Euclidean distance, $D1$, from the centroids $\{\mu_1, \mu_2, \ldots, \mu_{n1}\}$ of clients in the C_{normal} cluster, where $n1$ represents the number of clients in this cluster.
   - Calculate the maximum Euclidean distance, $D2$, from the centroids $\{\mu_1, \mu_2, \ldots, \mu_{n2}\}$ of clients in the C_{comp} cluster, where $n2$ represents the number of clients in this cluster.
   - Compute the distance, $D3$, between the centroids of the C_{normal} and C_{comp} clusters.

   Execution Choice: Based on these distances, we make the following decision

   - If $D3$ is greater than both $D1$ and $D2$, this indicates a satisfactory separation between clusters. The framework selects the larger cluster based on the number of clients (using $n1$ for normal and $n2$ for compromised) and utilizes its average weight for aggregation.
   - If the clustering criteria are not met, the framework bypasses clustering and resorts to the FedAVG algorithm, which averages the weights of all client models for aggregation.

   The underlying assumption is that compromised clients are relatively less compared to normal clients. By leveraging K-means clustering, we can distinguish these two groups based on their model weights characteristics.

5. **Stopping Criterion:**
   By employing this stopping criterion, we can effectively stop the process when the global model MAPE value ceases to improve significantly, thereby conserving computational resources and ensuring timely completion. This criterion essentially verifies if the current global model MAPE value is the lowest observed within the past 25 iterations. If so, it suggests that the algorithm or analysis has reached a stage where additional iterations or calculations are unlikely to yield substantial improvements.

Algorithm 2 CBAA-FedAvg Algorithm

Require: Communication rounds T, Participant clients N, Layers l

Initialize model weights $\{w_{ij}\}_{j=1}^{l}$ and distribute to all clients $i = 1, \ldots, N$
for $t = 1$ to T **do**
  for each client $i = 1$ to N **do**
    Compute quantized weights: $\bar{w}_{ij}^{(t)} \leftarrow \texttt{Quantize}(w_{ij}^{(t)}, n = 1, m = 7)$
    for each layer $j = 1$ to l **do**
      Calculate average: $W_j^{(i,t)} \leftarrow \text{Avg}(\bar{w}_{ij}^{(t)})$
    end for
    Determine centroid $\mu_i^{(t)} = \frac{1}{l} \sum_{j=1}^{l} W_j^{(i,t)}$
  end for
  Apply K-means clustering with $k = 2$ to centroids $\{\mu_i^{(t)}\}_{i=1}^{P}$ and partition clients into clusters $C_{\text{normal}}^{(t)}$ and $C_{\text{comp}}^{(t)}$
  Compute centroid distances:
  $D1 = \max\left(\|\mu_i^{(t)} - \text{centroid}(C_{\text{normal}}^{(t)})\|\right)$ for $i \in C_{\text{normal}}^{(t)}$
  $D2 = \max\left(\|\mu_i^{(t)} - \text{centroid}(C_{\text{comp}}^{(t)})\|\right)$ for $i \in C_{\text{comp}}^{(t)}$
  $D3 = \|\text{centroid}(C_{\text{normal}}^{(t)}) - \text{centroid}(C_{\text{comp}}^{(t)})\|$
  if ($D3 > D1$ **and** $D3 > D2$) **then**
    Select larger cluster $C_{\text{selected}}^{(t)} = \text{argmax}\left(\text{size}(C_{\text{normal}}^{(t)}), \text{size}(C_{\text{comp}}^{(t)})\right)$
    Aggregate weights for global model: $W_{t+1,j} = \frac{1}{\text{size}(C_{\text{selected}}^{(t)})} \sum_{i \in C_{\text{selected}}^{(t)}} W_j^{(i,t)}$
  else
    Use standard FedAvg: $W_{t+1,j} = \frac{1}{N} \sum_{i=1}^{N} W_j^{(i,t)}$
  end if
  Evaluate stopping criterion:
  if (Current GM MAPE > observed MAPE in last 25 iterations) **then**
    Continue to the next iteration.
  else
    Terminate FL.
  end if
end for
return Trained global model weights $\{W_{T,j}\}_{j=1}^{L}$ and final clustering results.

5 EXPERIMENT AND RESULTS

In this section, we present the outcomes of the experimental assessments conducted on three different approaches: the baseline method, our proposed data poisoning attack strategy, and model poisoning attack. We begin by outlining the dataset used and the parameters kept consistent across all trials. We then proceed to analyze and compare the performance of the proposed approach under various scenarios. Additionally, we evaluate countermeasures aimed at mitigating the attacks through purpose strategy and benchmarks. Finally, we offer a comprehensive discussion of the overall results.

5.1 Simulation Setup

The research was carried out on a desktop computer equipped with an Intel Core i5-6200U CPU running at 2.30GHz and 12.0 GB of installed RAM. The operating system utilized was Windows 10 Pro (64-bit). Visual Studio Code served as the text editor, and Python 3.11.4 was employed as the programming language.

5.1.1 Dataset & FL Setup

For our experiments, we utilize the PJME hourly dataset sourced from PJM Interconnection LLC, which is publicly available on Kaggle [70]. This dataset captures real-world energy consumption across various substations, providing valuable insights for research and analysis. In our experimental FL setup, we create ten independent clients, each contributing unique datasets. Each client's dataset consists of 14,537 samples, starting from January 1, 2017. The PJME hourly dataset is specifically tailored for load forecasting and includes five features: the value from the last hour, last day, last week, the 24-hour average, and the weekly average. Figure 6 provides an overview of the data distribution across all clients, as well as the format used in our experiments. The figure demonstrates that the differences in data distribution before and after the process are minimal, making our attack significantly more sophisticated.

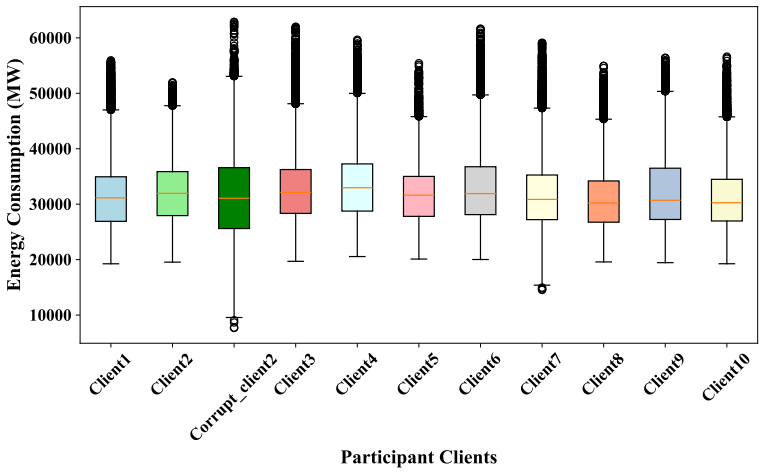


Fig. 6: Data distribution across all participant clients

5.1.2 Architectural paradigm

To enable load forecasting, we constructed a three-layered Artificial Neural Network (ANN). The first layer consisted of 100 neurons, followed by a layer with 50 neurons, and a final layer with a single neuron. All layers were equipped with Rectified Linear Unit (ReLU) activation functions. We employed the Adam optimizer, using mean

square error as the loss function, for training the network. We divided the dataset into training and testing sets using a 70/30 split. The training process was conducted using FL over 100 communication rounds. Each client underwent 5 local epochs, and a batch size of 300 was used for training at each client. To evaluate the performance of the global model, we created a comprehensive global dataset by aggregating 10% of the data from each individual client. This unified dataset was used to assess the accuracy and effectiveness of the global model in load forecasting. The simulation was programmed to terminate upon reaching the stopping criterion. This criterion checks whether the current global model's MAPE value is the lowest within the past 25 observations and efficiently halts the process, enabling the analysis of the results or outputs obtained during the simulation.

5.1.3 Evaluation criteria

In this simulation, we assess the performance of the load forecasting models using the MAPE metric. MAPE is a commonly used metric to measure the accuracy of forecasting or prediction models. It provides a normalized measure of the average absolute percentage difference between the predicted and actual values.
MAPE calculated as:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{A_i - F_i}{A_i} \right| \tag{6}$$

where A_i represents the actual value, F_i denotes the forecasted value, and n is the total number of iterations. A lower MAPE indicates a higher degree of accuracy in the load forecasting models. A MAPE value of zero would indicate a perfect prediction, where the predicted values exactly match the actual values.

5.2 Evaluation Outcomes

The results of the study demonstrate the adverse effects of Poisoning attacks on system performance within the FL framework.

5.2.1 Baseline (No Attack)

The baseline simulation serves as a reference point prior to any attack, providing a starting or default scenario for comparison with post-attack simulations. It represents the system's specifications, assumptions, and initial conditions before any disruptive event occurs. The findings from assessing the original (non-poised) data on the model are visually depicted in figure 7. This representation showcases the MAPE across the global model throughout various communication rounds, with a reported value of 2.71%.

5.2.2 Data Poisoning Attack

To analyze the impact of data poisoning attacks on the global model's MAPE (%), we examine two types of attacks: Single-Client and Multi-Client data poisoning attacks.

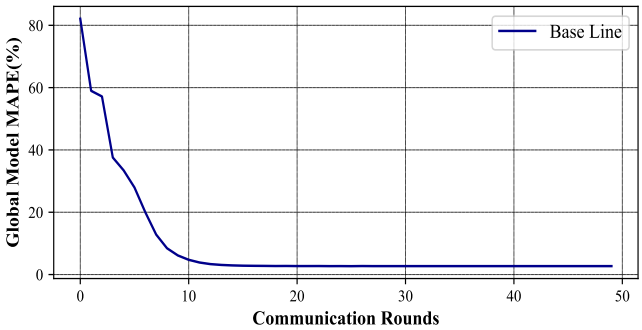


Fig. 7: Global Model MAPE (%) for the original (non-poised) data

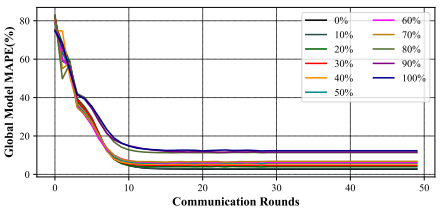
Table 1 showcases the outcome of both scenarios.

Attack on Single Client:
Figure 8a depicts the results of our analysis of a single-client data flipping attack, involving modifications ranging from 10 to 100% of the client's data. In this scenario, system performance degradation was measured as the proportion of flipped data increased. Initially, at a 10% data flipping rate, there was a slight decline in performance from 2.71% to 3.053%. Subsequently, as the flipping rate increased, the degradation became more pronounced, reaching a peak global model MAPE of 12.324% when the entire client's data was flipped. However, this increase in MAPE might not suffice to necessitate a change in the federated environment, prompting us to consider a multi-client attack.

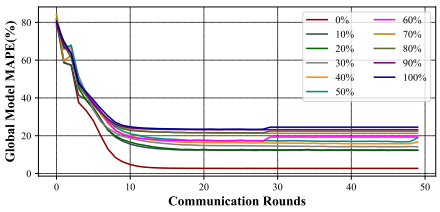
Attack on Multi clients:
In addition to single client attacks, we also investigated the impact of multiple compromised clients. The attack compromised each client's entire data, targeting 10-100% of participants. A clear pattern emerged, as presented in Figure 8b, revealing a direct correlation between the compromised client percentage and the resulting global model MAPE values. For instance, with only 10% of the participants compromised, the global model exhibited a MAPE of 12.28% which steadily rose with higher compromise rates, reaching 24.53% when all participant data was manipulated.

5.2.3 Model Poisoning Attack

In this attack scenario, merely flipping the weights of a single layer has a negligible impact on the system's performance, as shown in Figure 9a. However, when all layers' weights are flipped, indicating a full-scale poisoning attack, the consequences are substantial. The attack, primarily aimed at a single client, can disrupt the system significantly if a full poisoning attack occurs. It's crucial to recognize that additional attacks on other clients may lead to the complete destruction of the entire system.
To mitigate the risk of complete system disruption while ensuring the stealth and



(a) Single Client: Attack Range from 10% to 100%



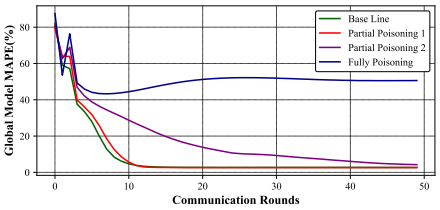
(b) Multi-client: Targeting 10-100% of Participants

Fig. 8: Comparison of Attack Ranges

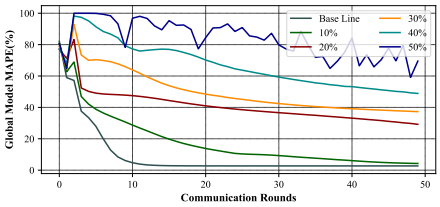
Table 1: Analysis of Global Model MAPE (%) under Data Flipping Attack: Single-Client (10-100% of Data) vs. Multi-Client (10-100% of Participants)

	Global Model MAPE (%)										
Attack Ratio (%)	0	10	20	30	40	50	60	70	80	90	100
Single Client	2.71	3.05	4.08	4.66	5.52	5.95	6.24	6.79	11.34	11.62	12.32
Multi Clients	2.71	12.32	13.39	14.43	16.23	18.2	19.34	21.25	22.47	23.22	24.53

persistence of an attack across multiple or single clients, we opt for a partial poisoning approach. This strategy involves targeting layer 1 and layer 2 specifically. When targeting 10-50% of the participants, the global model's performance experienced a significant decline, as illustrated in Figure 9b. Initially, the baseline global model MAPE stood at 2.7%. However, when compromising 10% of the participants' models, it increased to 4.3%. As the percentage of compromised clients increased, the impact on the global model became increasingly severe. This value surged to 29.2% with a compromise rate of 20% and continued to worsen with rates of 30%, 40%, and 50%, reaching 37.3%, 48.9%, and 69.5%, respectively.



(a) Comparison of Model Poisoning Attack Strategies



(b) Model Attack: Targeting 10-50% of Participants

Fig. 9: Comparison of Different Attack Strategies

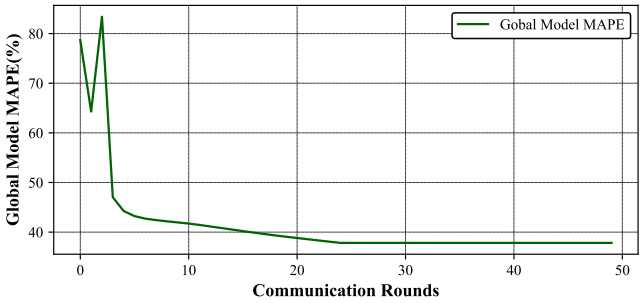


Fig. 10: Combine Attack: client 1 under data attack and client 2 and 3 under model attack

5.2.4 Data and Model Poisoning Attack

To assess the combined effect of both data and model poisoning in the same FL setting, Client 1 was subjected to a data poisoning attack, while clients 2 and 3 experienced model poisoning attacks. The resulting global model's MAPE was observed to be 36.8%, highlighting the detrimental impact of these combined attacks on the overall model's performance.

These findings highlight the vulnerability of FL to poisoning attacks, as even a small percentage of compromised participants can lead to significant disruptions in predictions. The escalating MAPE values demonstrate the effectiveness of the poisoning attacks in manipulating the global model's output, emphasizing the need for robust security measures to protect against such adversarial threats.

5.3 Attack resolution via CBAA-FedAvg

In this section, we evaluate the resilience of our proposed federated load forecasting framework, CBAA-FedAvg, against data and model poisoning attacks. Additionally, we investigate the performance of our approach under both attacks.
As above-mentioned, the global model's MAPE, diminishes notably with an increase in the number of compromised clients under both attacks. Nevertheless, CBAA-FedAvg demonstrates relatively consistent performance in load forecasting contexts, even under poisoning attacks. The rise in global model MAPE is marginal, signifying that our approach can uphold satisfactory performance despite the presence of poisoning attacks. Furthermore, the table 2 delineates the convergence behavior of CBAA-FedAvg across distinct types of poisoning attacks and varying degrees of attack intensity. It is evident that as the number of compromised clients escalates, the convergence of the global model's MAPE deteriorates swiftly. This underscores FedAVG's vulnerability to poisoning attacks, which can substantially impair its performance in practical scenarios. The divergence observed in the convergence trajectories of the global model's MAPE underscores the necessity for resilient FL algorithms.

From figure in 11, CBAA-FedAvg demonstrates consistent global model MAPE values throughout the training process, even when facing poisoned nodes, showcasing its robustness against poisoning attacks. Unlike FedAVG, CBAA-FedAvg's loss

values remain stable, even with an escalation in the number of adversarial client nodes, affirming its ability to withstand such attacks while preserving convergence properties. Remarkably, the proposed CBAA-FedAvg algorithm effectively mitigated data poisoning attacks and model poisoning attacks simultaneously, achieving efficient convergence with a global model MAPE value of 2.7%.

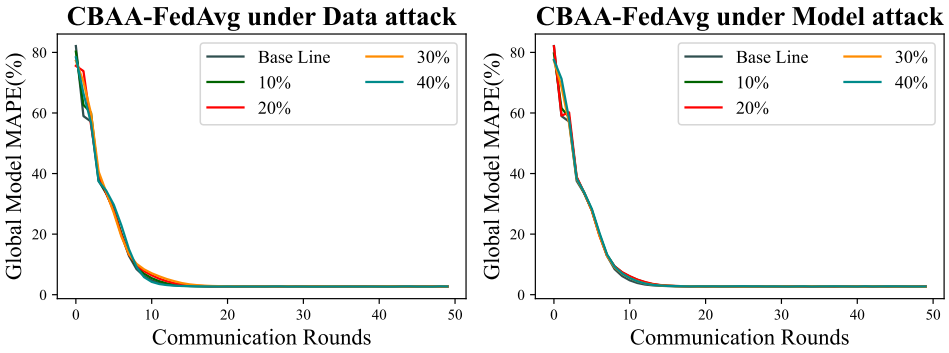


Fig. 11: CBAA-FedAVG Impact on Data and model Attacks

Table 2: Comparative Analysis of Global Model MAPE (%): FedAvg (Benchmark) vs. CBAA-FedAvg (Proposed)

Compromised Clients (%)	**Global Model MAPE (%)**			
	Model Poisoning		**Data Poisoning**	
	FedAvg	**CBAA-FedAvg**	**FedAvg**	**CBAA-FedAvg**
10	4.262	2.697	12.277	2.721
20	29.223	2.706	13.318	2.729
30	37.294	2.662	14.137	2.719
40	48.882	2.716	16.763	2.723
50	69.484	2.711	18.942	2.752

6 Comparative analysis

In our evaluation of computational efficiency, we compared the performance of our proposed CBAA-FedAvg method with two other approaches, namely Spectral clustering [44] and ZeKoC, a Zero Knowledge Clustering approach [45]. In terms of

results, all three methods, including CBAA-FedAvg, demonstrated similar levels of accuracy with a global model MAPE of approximately 2.70%. This indicates that they are comparable in their ability to forecast accurately in FL-aided smart grids. However, when considering computational efficiency, CBAA-FedAvg emerged as the most promising solution. Through a comprehensive analysis of CPU time, disk space utilization, and memory usage, we gained valuable insights shown in Figure 12 into the practical implementation and scalability of each method.
ZeKoC employs a robust clustering-based approach comprising three components: first, it internalizes k clusters, then it checks if any cluster needs further splitting. Next, it revisits all the clusters and merges those that closely resemble each other. After that, it evaluates the behavior of each cluster and discards any that are deemed problematic. This process is carried out at each communication round. This method demands substantial CPU usage and elapsed time. Spectral Clustering, on the other hand, segregates weights into two clusters and eliminates the smaller cluster deemed as an adversary. Despite achieving lower CPU usage and elapsed time compared to ZeKoC, it still falls short of CBAA-FedAvg in computational efficiency.

Efficient Communication in CBAA-FedAvg:

The primary advantage of CBAA-FedAvg lies in its use of quantization techniques, which offer several benefits. The storage requirements for the model can be substantially reduced by using low-bit precision representations, leading to a significant decrease in power consumption. Similarly, the memory bandwidth requirements can also be significantly lowered. Perhaps the most important benefit of low-bit representations is the savings in chip area. For instance, 8-bit fixed point operations can save up to 30x energy and up to 116x area compared to 32-bit FP operations [71]. This allows for significantly better computational throughput, as the multiply-accumulate operations can be performed on low-bit processing engines, reducing the computational complexity.
To further enhance efficiency, CBAA-FedAvg employs a dynamic clustering criterion, which further reduces the computational complexity. Additionally, the algorithm utilizes an automatic stopping criterion, which saves energy and time by enabling early, yet optimized, convergence. The combination of quantization, dynamic clustering, and the stopping criterion contribute to more efficient resource utilization and quicker convergence, while also mitigating the impact of utility-centric attacks.

7 Conclusion

This study addresses the gap in existing literature by highlighting the specific threat of data poisoning attacks on load forecasting in FL and proposing a novel approach to mitigate these attacks for smart grids. The introduction of data flipping attack manipulates energy data in a way undetectable by traditional statistical methods. and Developed model flipping attack in FL setting. To address these challenges, the proposed CBAA-FedAvg framework, a unified and sustainable solution, achieving a global model Mean Absolute Percentage Error (MAPE) of 2.7%, which closely matches baseline performance while demonstrating remarkable resilience against both data and model adversarial attacks. Efficient resource utilization is achieved with low CPU

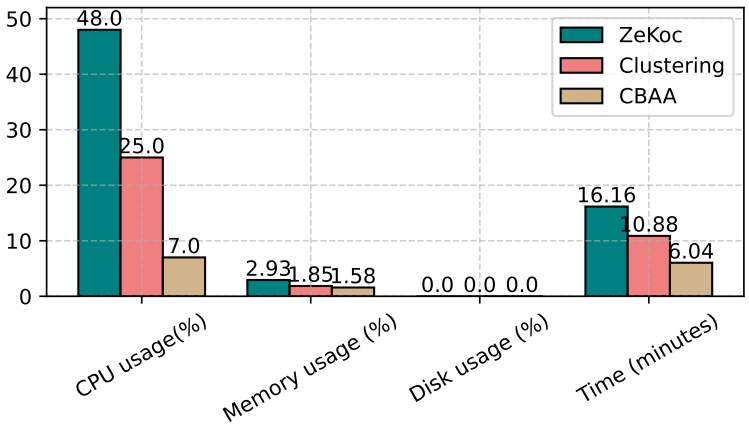


Fig. 12: Resource Utilisation of ZeKoC, Spectral clustering and CBAA-FedAVG.

usage (7%), minimal memory usage (1.58%), negligible disk usage (0%), and a total elapsed time of 362.4 seconds. This is accomplished by quantizing parameters from 32-bit floating point to 8-bit fixed point and employing a dynamic clustering criterion to further reduce computational complexity. Additionally, an automatic stopping criterion is utilized, allowing energy and time to be conserved through early yet optimized convergence. These features collectively contribute to making CBAA-FedAvg a lightweight and efficient design compared to state-of-the-art alternatives.

Funding. This Paper is supported by Ajman University Internal Research Grant No. 2023-IRG-ENIT-5. The research findings presented in this paper are solely the author(s)' responsibility.

Ethical Approval. No animal or humans were involved in this study.

Competing interests. Authors do not have any competing interest.

Authors' contributions. Attia Shabbir (Conceptualization: Equal; Data curation: Lead; Methodology: Supporting; Software: Lead; Visualization: Lead; Writing – original draft: Equal)
Habibullah Manzoor, PhD (Conceptualization: Supporting; Data curation: Equal; Formal analysis: Equal; Methodology: Equal; Resources: Equal; Software: Equal; Supervision: Supporting; Validation: Equal; Writing – original draft: Equal; Writing–review & editing: Equal)
Kamran Arshad, PhD (Formal analysis: Equal; Funding acquisition: Equal; Resources: Lead; Validation: Lead)
Khaled Assaleh (Funding acquisition: Equal; Methodology: Supporting; Project administration: Supporting; Validation: Equal; Visualization: Equal)
Zahid Halim (Formal analysis: Equal; Investigation: Equal; Resources: Equal; Supervision: Equal; Writing – review & editing: Supporting)
Ahmed Zoha (Conceptualization: Supporting; Data curation: Supporting; Formal

analysis: Supporting; Investigation: Supporting; Software: Supporting; Supervision: Equal; Writing – original draft: Supporting; Writing – review & editing: Lead)

Availability of data and materials. The used dataset is publicly available at :https://www.kaggle.com/datasets/robikscube/hourly-energy-consumption/data.

References

[1] Hobbs, B.F., Jitprapaikulsarn, S., Konda, S., Chankong, V., Loparo, K.A., Maratukulam, D.J.: Analysis of the value for unit commitment of improved load forecasts. IEEE Transactions on Power Systems **14**(4), 1342–1348 (1999)

[2] Zhang, Q., Yuan, Q., Zhou, X., Luo, X.: Research on intelligent load forecast in power system dispatching automation. In: 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT), pp. 575–578 (2021). IEEE

[3] Gupta, A., Kumar, A.: Mid term daily load forecasting using arima, wavelet-arima and machine learning. In: 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), pp. 1–5 (2020). IEEE

[4] Chen, Y., Tan, Y., Zhang, B.: Exploiting vulnerabilities of load forecasting through adversarial attacks. In: Proceedings of the Tenth ACM International Conference on Future Energy Systems, pp. 1–11 (2019)

[5] Su, Z., Wang, Y., Luan, T.H., Zhang, N., Li, F., Chen, T., Cao, H.: Secure and efficient federated learning for smart grid with edge-cloud collaboration. IEEE Transactions on Industrial Informatics **18**(2), 1333–1344 (2021)

[6] Manzoor, H.U., Khan, A.R., Al-Quraan, M., Mohjazi, L., Taha, A., Abbas, H., Hussain, S., Imran, M.A., Zoha, A.: Energy management in an agile workspace using ai-driven forecasting and anomaly detection. In: 2022 4th Global Power, Energy and Communication Conference (GPECOM), pp. 644–649 (2022). IEEE

[7] Li, K., Wang, H., Zhang, Q.: Fedtcr: communication-efficient federated learning via taming computing resources. Complex & Intelligent Systems, 1–21 (2023)

[8] Petrangeli, E., Tonellotto, N., Vallati, C.: Performance evaluation of federated learning for residential energy forecasting. IoT **3**(3), 381–397 (2022)

[9] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282 (2017). PMLR

[10] Singh, P., Singh, M.K., Singh, R., Singh, N.: Federated learning: Challenges, methods, and future directions. In: Federated Learning for IoT Applications, pp.

199–214. Springer, ??? (2022)

[11] Manzoor, H.U., Khan, A.R., Flynn, D., Alam, M.M., Akram, M., Imran, M.A., Zoha, A.: Fedbranched: Leveraging federated learning for anomaly-aware load forecasting in energy networks. Sensors **23**(7), 3570 (2023)

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

[13] Zhao, B., Mopuri, K.R., Bilen, H.: idlg: Improved deep leakage from gradients. arXiv preprint arXiv:2001.02610 (2020)

[14] Manzoor, H.U., Khan, M.S., Khan, A.R., Ayaz, F., Flynn, D., Imran, M.A., Zoha, A.: Fedclamp: An algorithm for identification of anomalous client in federated learning. In: 2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 1–4 (2022). IEEE

[15] Cao, X., Fang, M., Liu, J., Gong, N.Z.: Fltrust: Byzantine-robust federated learning via trust bootstrapping. arXiv preprint arXiv:2012.13995 (2020)

[16] Marnissi, O., El Hammouti, H., Bergou, E.H.: Adaptive sparsification and quantization for enhanced energy efficiency in federated learning. IEEE Open Journal of the Communications Society **5**, 4307–4321 (2024) https://doi.org/10.1109/OJCOMS.2024.3425531

[17] Lyu, X., Hou, X., Ren, C., Ge, X., Yang, P., Cui, Q., Tao, X.: Secure and efficient federated learning with provable performance guarantees via stochastic quantization. IEEE Transactions on Information Forensics and Security **19**, 4070–4085 (2024) https://doi.org/10.1109/TIFS.2024.3374590

[18] Manzoor, H.U., Jafri, A., Zoha, A.: Lightweight single-layer aggregation framework for energy-efficient and privacy-preserving load forecasting in heterogeneous smart grids. Authorea Preprints (2024)

[19] Curtis, A.E., Smith, T.A., Ziganshin, B.A., Elefteriades, J.A.: The mystery of the z-score. Aorta **4**(04), 124–130 (2016)

[20] Hong, T., Wang, P.: Artificial intelligence for load forecasting: history, illusions, and opportunities. IEEE Power and Energy Magazine **20**(3), 14–23 (2022)

[21] Chen, B.-J., Chang, M.-W., *et al.*: Load forecasting using support vector machines: A study on eunite competition 2001. IEEE transactions on power systems **19**(4), 1821–1830 (2004)

[22] Bouktif, S., Fiaz, A., Ouni, A., Serhani, M.A.: Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm:

Comparison with machine learning approaches. Energies **11**(7), 1636 (2018)

[23] Zheng, J., Xu, C., Zhang, Z., Li, X.: Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. In: 2017 51st Annual Conference on Information Sciences and Systems (CISS), pp. 1–6 (2017). IEEE

[24] Marino, D.L., Amarasinghe, K., Manic, M.: Building energy load forecasting using deep neural networks. In: IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society, pp. 7046–7051 (2016). IEEE

[25] Almalaq, A., Zhang, J.J.: Evolutionary deep learning-based energy consumption prediction for buildings. ieee access **7**, 1520–1531 (2018)

[26] Stephen, B., Tang, X., Harvey, P.R., Galloway, S., Jennett, K.I.: Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting. IEEE Transactions on Smart Grid **8**(4), 1591–1598 (2015)

[27] Shi, H., Xu, M., Li, R.: Deep learning for household load forecasting—a novel pooling deep rnn. IEEE Transactions on Smart Grid **9**(5), 5271–5280 (2017)

[28] Kumar, P., Lin, Y., Bai, G., Paverd, A., Dong, J.S., Martin, A.: Smart grid metering networks: A survey on security, privacy and open research issues. IEEE Communications Surveys & Tutorials **21**(3), 2886–2927 (2019)

[29] Badra, M., Zeadally, S.: Design and performance analysis of a virtual ring architecture for smart grid privacy. IEEE transactions on information forensics and security **9**(2), 321–329 (2014)

[30] Venkataramanan, V., Kaza, S., Annaswamy, A.M.: Der forecast using privacy-preserving federated learning. IEEE Internet of Things Journal **10**(3), 2046–2055 (2022)

[31] Yang, Y., Wang, Z., Zhao, S., Wu, J.: An integrated federated learning algorithm for short-term load forecasting. Electric Power Systems Research **214**, 108830 (2023)

[32] Wang, Y., Gao, N., Hug, G.: Personalized federated learning for individual consumer load forecasting. CSEE Journal of Power and Energy Systems **9**(1), 326–330 (2022)

[33] Zhang, G., Zhu, S., Bai, X.: Federated learning-based multi-energy load forecasting method using cnn-attention-lstm model. Sustainability **14**(19), 12843 (2022)

[34] Xu, C., Chen, G., Li, C.: Federated learning for interpretable short-term residential load forecasting in edge computing network. Neural Computing and Applications **35**(11), 8561–8574 (2023)

[35] Manzoor, H.U., Khan, A.R., Sher, T., Ahmad, W., Zoha, A.: Defending federated learning from backdoor attacks: Anomaly-aware fedavg with layer-based aggregation. In: 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1–6 (2023). IEEE

[36] Xia, G., Chen, J., Yu, C., Ma, J.: Poisoning attacks in federated learning: A survey. IEEE Access **11**, 10708–10722 (2023)

[37] Manzoor, H.U., Arshad, K., Assaleh, K., Zoha, A.: Enhanced adversarial attack resilience in energy networks through energy and privacy aware federated learning. Authorea Preprints (2024)

[38] Cao, X., Gong, N.Z.: Mpaf: Model poisoning attacks to federated learning based on fake clients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3396–3404 (2022)

[39] Sun, Z., Kairouz, P., Suresh, A.T., McMahan, H.B.: Can you really backdoor federated learning? arXiv preprint arXiv:1911.07963 (2019)

[40] Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX Security Symposium (USENIX Security 20), pp. 1605–1622 (2020)

[41] Li, H., Wu, S., Wang, R., Guo, Y., Li, J.: Fed-sad: A secure aggregation federated learning method for distributed short-term load forecasting. IET Generation, Transmission & Distribution **17**(22), 5090–5100 (2023)

[42] Husnoo, M.A., Anwar, A., Hosseinzadeh, N., Islam, S.N., Mahmood, A.N., Doss, R.: A secure federated learning framework for residential short term load forecasting. IEEE Transactions on Smart Grid (2023)

[43] Moradzadeh, A., Moayyed, H., Mohammadi-Ivatloo, B., Aguiar, A.P., Anvari-Moghaddam, A.: A secure federated deep learning-based approach for heating load demand forecasting in building environment. IEEE Access **10**, 5037–5050 (2021)

[44] Qureshi, N.B.S., Kim, D.-H., Lee, J., Lee, E.-K.: Poisoning attacks against federated learning in load forecasting of smart energy. In: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, pp. 1–7 (2022). IEEE

[45] Chen, Z., Tian, P., Liao, W., Yu, W.: Zero knowledge clustering based adversarial mitigation in heterogeneous federated learning. IEEE Transactions on Network Science and Engineering **8**(2), 1070–1083 (2021) https://doi.org/10.1109/TNSE.2020.3002796

[46] Sun, G., Cong, Y., Dong, J., Wang, Q., Lyu, L., Liu, J.: Data poisoning attacks on federated machine learning. IEEE Internet of Things Journal **9**(13), 11365–11375

(2021)

[47] Zhao, L., Li, J., Li, Q., Li, F.: A federated learning framework for detecting false data injection attacks in solar farms. IEEE Transactions on Power Electronics **37**(3), 2496–2501 (2021)

[48] Lin, W.-T., Chen, G., Zhou, X.: Privacy-preserving federated learning for detecting false data injection attacks on power system. Electric Power Systems Research **229**, 110150 (2024)

[49] Keçeci, C., Davis, K.R., Serpedin, E.: Federated learning based distributed localization of false data injection attacks on smart grids. arXiv preprint arXiv:2306.10420 (2023)

[50] Li, Y., Wei, X., Li, Y., Dong, Z., Shahidehpour, M.: Detection of false data injection attacks in smart grid: A secure federated deep learning approach. IEEE Transactions on Smart Grid **13**(6), 4862–4872 (2022)

[51] Liu, Y., Garg, S., Nie, J., Zhang, Y., Xiong, Z., Kang, J., Hossain, M.S.: Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach. IEEE Internet of Things Journal **8**(8), 6348–6358 (2020)

[52] Lin, W.-T., Chen, G., Huang, Y.: Incentive edge-based federated learning for false data injection attack detection on power grid state estimation: A novel mechanism design approach. Applied Energy **314**, 118828 (2022)

[53] Cheng, Y., Wang, D., Zhou, P., Zhang, T.: Model compression and acceleration for deep neural networks: The principles, progress, and challenges. IEEE Signal Processing Magazine **35**(1), 126–136 (2018)

[54] Wang, B., Fang, J., Li, H., Zeng, B.: Communication-efficient federated learning: A variance-reduced stochastic approach with adaptive sparsification. IEEE Transactions on Signal Processing (2023)

[55] Jiang, Y., Wang, S., Valls, V., Ko, B.J., Lee, W.-H., Leung, K.K., Tassiulas, L.: Model pruning enables efficient federated learning on edge devices. IEEE Transactions on Neural Networks and Learning Systems **34**(12), 10374–10386 (2022)

[56] Chen, S., Shen, C., Zhang, L., Tang, Y.: Dynamic aggregation for heterogeneous quantization in federated learning. IEEE Transactions on Wireless Communications **20**(10), 6804–6819 (2021)

[57] He, Q., Su, Y.: Residential load forecasting based on cnn-lstm and non-uniform quantization. In: 2022 12th International Conference on Power and Energy Systems (ICPES), pp. 586–591 (2022). IEEE

[58] Mao, Z., Li, H., Huang, Z., Yang, C., Li, Y., Zhou, Z.: Communication-efficient federated learning for power load forecasting in electric iots. Ieee Access **11**, 47930–47939 (2023)

[59] Zhao, L., Hu, S., Wang, Q., Jiang, J., Shen, C., Luo, X., Hu, P.: Shielding collaborative learning: Mitigating poisoning attacks through client-side detection. IEEE Transactions on Dependable and Secure Computing **18**(5), 2029–2041 (2020)

[60] Mian, A.N., Shah, S.W.H., Manzoor, S., Said, A., Heimerl, K., Crowcroft, J.: A value-added iot service for cellular networks using federated learning. Computer Networks **213**, 109094 (2022)

[61] Manzoor, S., Mian, A.N., Zoha, A., Imran, M.A.: Federated learning empowered mobility-aware proactive content offloading framework for fog radio access networks. Future Generation Computer Systems **133**, 307–319 (2022)

[62] Khan, A.R., Manzoor, H.U., Ayaz, F., Imran, M.A., Zoha, A.: A privacy and energy-aware federated framework for human activity recognition. Sensors **23**(23) (2023) https://doi.org/10.3390/s23239339

[63] Malinovskiy, G., Kovalev, D., Gasanov, E., Condat, L., Richtarik, P.: From local sgd to local fixed-point methods for federated learning. In: International Conference on Machine Learning, pp. 6692–6701 (2020). PMLR

[64] Zhong, Z., Zhou, Y., Wu, D., Chen, X., Chen, M., Li, C., Sheng, Q.Z.: P-fedavg: Parallelizing federated learning with theoretical guarantees. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, pp. 1–10 (2021). IEEE

[65] Zari, O., Xu, C., Neglia, G.: Efficient passive membership inference attack in federated learning. arXiv preprint arXiv:2111.00430 (2021)

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

[67] Shabbir, A., Manzoor, H.U., Ahmed, R.A., Halim, Z.: Resilience of federated learning against false data injection attacks in energy forecasting. In: 2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST), pp. 245–249 (2024). IEEE

[68] Shejwalkar, V., Houmansadr, A.: Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In: NDSS (2021)

[69] Manzoor, H.U., Hussain, S., Flynn, D., Zoha, A.: Centralised vs. decentralised federated load forecasting: Who holds the key to adversarial attack robustness? Authorea Preprints (2024)

[70] Mulla, R.: Hourly energy consumption

[71] Choukroun, Y., Kravchik, E., Yang, F., Kisilev, P.: Low-bit quantization of neural networks for efficient inference. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 3009–3018 (2019). IEEE