energies

MDPI

Article

Federated Deep Learning Model for False Data Injection Attack Detection in Cyber Physical Power Systems

Firdous Kausar [1,*,†], Sambrdhi Deo [1,†], Sajid Hussain [2,†] and Zia Ul Haque [1,*]

[1] Mathematics and Computer Science Department, Fisk University, Nashville, TN 37208, USA; sdeo06@my.fisk.edu
[2] School of Applied Computational Sciences, Meharry Medical College, Nashville, TN 37208, USA; sajid.hussain@mmc.edu
* Correspondence: fkausar@fisk.edu (F.K.); zhaque@fisk.edu (Z.U.H.)
† These authors contributed equally to this work.

Abstract: Cyber-physical power systems (CPPS) integrate information and communication technology into conventional electric power systems to facilitate bidirectional communication of information and electric power between users and power grids. Despite its benefits, the open communication environment of CPPS is vulnerable to various security attacks. This paper proposes a federated deep learning-based architecture to detect false data injection attacks (FDIAs) in CPPS. The proposed work offers a strong, decentralized alternative with the ability to boost detection accuracy while maintaining data privacy, presenting a significant opportunity for real-world applications in the smart grid. This framework combines state-of-the-art machine learning and deep learning models, which are used in both centralized and federated learning configurations, to boost the detection of false data injection attacks in cyber-physical power systems. In particular, the research uses a multi-stage detection framework that combines several models, including classic machine learning classifiers like Random Forest and ExtraTrees Classifiers, and deep learning architectures such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The results demonstrate that Bidirectional GRU and LSTM models with attention layers in a federated learning setup achieve superior performance, with accuracy approaching 99.8%. This approach enhances both detection accuracy and data privacy, offering a robust solution for FDIA detection in real-world smart grid applications.

Keywords: cyber-physical power systems; state estimation; federated learning; privacy preservation; deep learning; smart grid; false data injection attack; bidirectional LSTM; bidirectional GRU; attention layers

Citation: Kausar, F.; Deo, S.; Hussain, S.; Ul Haque, Z. Federated Deep Learning Model for False Data Injection Attack Detection in Cyber Physical Power Systems. *Energies* **2024**, *17*, 5337. https://doi.org/10.3390/en17215337

Academic Editors: Marcin Niemiec and Robert Ryszard Chodorek

Received: 28 September 2024
Revised: 19 October 2024
Accepted: 22 October 2024
Published: 26 October 2024

1. Introduction

With the adoption of data-intensive machine-learning techniques by science, technology, and business, the exponential growth of machine learning (ML) and deep learning (DL) technologies has revolutionized a number of industries, including healthcare, finance, and marketing. As a result, there is an increasing use of evidence in decision-making across a wide range of fields, including manufacturing, law enforcement, healthcare, education, and marketing [1]. The system has gained considerably in capability and application due to these advancements. Yet, the performance of ML and DL models is inherently contingent upon the quality of the data they process. High-quality data ensures accurate model predictions and reliable outcomes, whereas compromised data can lead to erroneous results with potentially catastrophic projections or consequences.

The term "data integrity" describes the process of making sure that data is trustworthy and original at all times. It includes data consistency, correctness, and completeness, all of which are important for ML and DL models to have trustworthy information.

Cyber-Physical Power Systems (CPPSs) and other decentralized environments are vulnerable to cyber-attacks because data in these systems often originates from different

locations. Multiple types of attacks, such as man-in-the-middle, false data injection, and denial-of-service (DoS) attacks, can undermine data integrity and affect the performance of ML and DL systems. Several cyberattacks have been reported in the energy business during the past few decades. This begins with the proclamation of the first major attack in 1982 often referred to as the "Siberian Pipeline Incident" [2].

A growing number of sophisticated cyberattacks have targeted the electrical infrastructure of the United States. It is anticipated that the record-breaking number of 185 significant cyber and physical attacks reported in 2022 will be surpassed in 2023. There has been a dramatic increase in system vulnerabilities as a result of the surge in cyber attacks; between 2022 and 2023, over sixty new vulnerabilities were discovered daily, for a total of approximately two thousand vulnerabilities [3].

These attacks have historically resulted in explosions, losses of millions of dollars, and loss of life. A growing frequency of these instances poses a serious threat [4]. Data integrity should be prioritized while using ML and DL technologies because of the risks involved. As part of this process, it is necessary to build strong security measures, collect correct data, validate and update data often, and stay current with updated cybersecurity methods.

The term "false data injection attack" (FDIA) was initially used in relation to smart grid technologies in 2011 [5]. Despite the word's seeming genericity, it really describes a specific attack scenario where an attacker deceives sensor readings in a way that introduces undetectable errors into state variables and value computations. The conventional bad data detectors usually miss malicious measurements injected by an attacker who knows the system topology.

The state estimation process relies on measurements taken from sensors. An adversary can compromise with this estimation by feeding the estimator false data. The essence of the attack is to alter the original state and deviate it slightly in the direction of a false state. At the same time, the attacker injects a carefully designed set of false data into the inputs to the process, which has the appearance of good measurement data [6]. This attack aims to compromise a system's estimations without activating any detection mechanism. The attacker makes it seem as if the compromised system is still functioning normally and that the data being fed into it is also normal. If the compromised system can remain undetected for a specific amount of time, the attacker can create a fake state estimation while keeping the system seeming normal. A stealth attack severely compromises the state estimation process for real-time data systems. This is highly significant because reliable state estimation is a must for critical infrastructure cybersecurity and is particularly important for systems such as smart grids, real-time industrial control systems, or IoT networks.

Modern information and communication technology have brought power systems one step closer to fully automating and streamlining their operations. Even though these new developments allow for the efficient running of operations, they also lay the system open to complex cyber threats [7]. Such threats are capable of disrupting vital infrastructures, and have already done so in several well-known instances, including the power grid attacks on the Ukrainian power authority in 2015 and 2016, as well as the ransomware assault on the Colonial Pipeline in 2021 [8].

Most grid operators depend on real-time estimates of their power system's current state to make accurate operational decisions that maintain the system's reliability. Attackers are increasingly focusing their attacks on state estimation, mostly via denial of service (DoS) and FDIA attacks [9]. These cyberattacks underscore the imperative for robust state estimation infrastructures and sophisticated detection systems to maintain not only secure, but also resilient operations in the smart grid. Recent research has shown the evolution of detection methods that use machine learning to find various kinds of adversarial activity in CPPSs. Since these methods "learn" from the data they process rather than relying on the type of model used by earlier generations, they outperform those methods [10]. There is a need for more effective and advanced methods of cybersecurity analysis. Unsupervised learning techniques will predominate in the next generation of solutions, which can effectively detect malicious activity in CPPS networks [11]. Furthermore, the integra-

tion of model-based and data-driven approaches presents considerable opportunities for strengthening the reliability of state estimation and enhancing the capacity to identify and categorize anomalies during real-time operations [12].

The study highlighted specific vulnerabilities in the way the smart grid estimates its state, and demonstrated how attackers could manipulate measurements to alter the system's perceived state [13]. The research results indicated that even an attacker with limited information about the grid's configuration could submit data that would result in state estimation being manipulated in a way that could cause operational problems in the smart grid [14]. A cyberattack of this nature targets a critical system particularly well. It poses such a threat because it deliberately introduces false data into the system. In addition, FDIAs can severely threaten power system security. Power systems are critical infrastructure, and an attack on them can have devastating economic and societal effects.

Ensuring the resilience of machine learning and deep learning models in the face of sophisticated evasion attacks is essential, especially in critical applications where the consequences of failure are severe. To this end, data scientists and researchers must apply meticulous data engineering practices to ensure that the quality of the data used to train and test these models is not compromised. However, in a decentralized context where data flows from multiple sources, maintaining data integrity becomes more challenging. Federated learning, which trains models across multiple decentralized devices without sharing raw data, offers a promising solution. Within the framework of Federated Learning (FL), initially proposed by Google in 2016, numerous devices work together under the supervision of a central server to build a machine learning model without revealing any personally identifiable information [15]. Federated learning reduces the possibility of sensitive data exposure by storing raw data locally and only sharing model changes. With data processed locally, the attack surface is reduced, making it more difficult for attackers to inject erroneous data over the network.

By sharing the training process, federated learning automatically supports data integrity. A global model is created by aggregating model updates made by each device using its local data. Since compromising the whole model would need coordinated attacks on several devices at the same time, this decentralized method makes it more difficult for fraudulent data injection to happen. Anomaly detection and mitigation can be aided by the collaborative nature of federated learning. One way to detect possible fraudulent data injections is to flag updates from devices that are noticeably different from the rest for additional analysis. To further protect data from manipulation, these devices perform cross-verification. The flexibility and scalability of federated learning make it an ideal solution for situations where the security of data are paramount. Industries like smart grids, autonomous vehicles, and distributed healthcare systems, which collect data from numerous locations, might greatly benefit from its decentralized operation.

Modern power systems are becoming complex cyber-physical systems due to the quick advancements in sensor, computer, and communication network technology. Therefore, evaluating and improving cyber-physical system security is crucial for the future electrical grid. An important danger to the security and availability of CPPSs is FDIAs. Incorrect decision-making and operational failures in CPPSs might result from these attacks, in which attackers modify the data inputs. Therefore, it is crucial to identify and address these. The electricity grid is vulnerable to major interruptions and damage caused by these attacks that go unnoticed and result in poor operational decisions.

A potential strategy to improve the security, privacy, and resilience of power systems is to use deep learning approaches based on advanced federated learning to detect and mitigate these types of attacks. The main contribution of this article is the detection of false data injection attacks using federated-based deep learning. We use a multi-stage detection framework.

- We perform machine learning-based binary classification where multiple algorithms are tested to differentiate between natural events and FDIAs attacks on CPPSs. This baseline model identifies probable security breaches and FDIAs.

- We implement advanced deep learning models on binary classification to identify subtle indicators of FDIAs by capturing complicated patterns and interdependencies in high-dimensional PMUs data.
- Addressing privacy concerns and enhancing model robustness in real-world scenarios is accomplished through the utilization of a federated learning architecture. It is a decentralized method where multiple stations take part in the training process to generate the collaborative model without sharing any of their datasets or with the centralized server. It will generate a generalized attack detection model with the capability to deal with different attack scenarios and system setups. We implement various deep learning algorithms, including LSTM, GRU, bidirectional LSTM, bidirectional GRU, and Transformer under federated learning architecture

The efficacy of each stage in our detection framework is assessed using industry-standard evaluation metrics. We compare the performance of traditional machine learning, centralized deep learning, and federated deep learning approaches. This comprehensive evaluation not only quantifies the detection accuracy, but also considers crucial factors such as false positive rates and computational efficiency.

The rest of this article is organized as follows. Section 2 reviews the related work. Section 3 describes the methodology. Section 4 provides the details of the implementation and evaluation of the models. Section 5 presents the results and analysis, while Section 6 concludes the paper.

2. Related Works

The integrity and reliability of CPPSs are under serious threat from FDIAs. In these attacks, adversaries insert themselves into the data stream and cause incorrect decision-making and operational failures in CPPSs. Therefore, identifying and mitigating these attacks is very important.

Liu et al. [5] first presented the idea of FDIAs in their investigation into false data injection attacks directed at state estimation in electric power grids. Their work shows that attackers can use the grid's measurement model to inject false data into it, which then gets processed by the state estimator, making the attacker "invisible". That is, the attack evades bad data detection mechanisms. Their reasoning and analytical model (which compares the attacker's effects on the model with a hypothetical "good" model) has led to significant worry in the field, because what they discussed can be thought of as a way to design a reliable attack on the power grid. This groundbreaking research established an essential base for comprehending the intricate cyberattacks directed at crucial infrastructures, such as electrical power grids, and initiated efforts to look further into how such attacks could be detected and mitigated. The concept of FDIA has been extended by Xie et al. [16] to the broader context of power market operations, where they investigate the economic impacts of such attacks. Kosut et al. [17] addressed two main points: the first was to refine the strategies used in attacks and the second to improve the algorithms used for detection. They concentrated their efforts on emphasizing the practical problems that these two points cause when you attempt to implement them in a real-world scenario.

Liang et al. [18] delved into the theoretical foundations of FDIAs, examining their physical and economic impacts and evaluating various defense strategies, including the protection of basic measurements and leveraging PMUs for enhanced protection. They emphasized the need for ongoing research to address emerging challenges and improve the resilience of power systems against FDIAs. The theoretical underpinnings of FDIAs were explored by Liang et al. [18]. They looked deeply into the physical and economic effects of such attacks, and assessed the currently known and conjectured defense strategies. These included measures to safeguard basic electrical quantities and the use of PMUs for much better protection. They suggested that research needs to continue in this important area so that power systems can be made more resilient to FDIAs.

In an extensive study, Yohanandhan et al. [19] reviewed CPPSs, concentrating on modeling, simulation, and analysis, particularly concerning matters of cybersecurity. They

accentuated the power systems' weaknesses—emanating from the cyber-physical integration—that the attackers can exploit and the serious threats they pose. One of the primary means of in-depth analyzing the FDIAs' effects on CPPS seems to be through co-simulation and real-time simulation. Musleh et al. [20] present an in-depth classification and appraisal of a range of different detection algorithms that are used to counteract FDIAs in smart grids. Their work centers mainly on the two principal categories of detection algorithms: model-based and data-driven. Within that framework, they discuss a number of different techniques—state estimation methods, residual methods, and several varieties of machine learning methods—that are used in detection algorithm construction and are representative of the kinds of mechanism that can be employed when designing detection algorithms.

The practical possibility of FDI attacks in today's smart grids was investigated by Khanna et al. [21]. They asserted that such attacks would not only be possible, but also relatively straightforward for an adversary to carry out. Their recommendations for making the smart grid more resistant to FDI attacks included ensuring the proper operation of critical sensors and improving detection of attacks in progress.

Cutting-edge methods using big data and artificial intelligence—for instance, deep learning techniques like Long Short-Term Memory (LSTM) networks and Generative Adversarial Networks (GANs)—now complement more traditional approaches to detecting false data injection attacks (FDIAs) in smart grids [22]. These new methods offer enhanced robustness against sophisticated, stealthy cyber threats and show much promise for the nearly real-time accuracy required for time-series data coming from electronic power system sensors. However, they do come with the substantial baggage of high computational costs and potential data privacy problems. Additionally, a comprehensive analysis by researchers have indicated that federated learning as an effective mechanism for detecting FDIAs [23]. Their analysis has illuminated that federated learning can be used to detect FDIAs, and that it may well be the most promising technique for doing so.

The detection of FDIAs can benefit from a decentralized, federated approach, according to Li et al. [24]. They reviewed applications of federated learning, emphasizing the approach's ability to keep users' data private and safe while still allowing the threat-detection model to function properly. The exhaustive review on federated learning by Mammen et al. [25] covered not only the architectural aspects of this new paradigm of machine learning, but also the numerous challenges that federated learning faces in real-world applications, including those related to system designs, and a lack of adequate quality control that is necessitated by the inherent data heterogeneity of federated learning. Along with these aspects, the review provided a detailed discussion of various security challenges that federated learning might encounter, such as data corruption and backdoor attacks. The authors also mentioned that federated learning might also suffer from a lack of user privacy. Gosselin et al. [26] provided even more detailed discussions of the numerous security and privacy issues that federated learning might face.

An innovative application of federated learning is found in solar energy systems. Zhao et al. [27] showed that this cutting-edge technology could just as well be used to detect FDIAs in solar energy systems while maintaining data privacy. By using a federated learning framework with a convolutional neural network, they managed to achieve an impressive detection rate with low false positives. A secure federated deep learning framework for detecting false data injection attacks was put forth by Li et al. [28] as a work to build upon for enhanced security and performance in smart grid environments. In their proposal, they combined the Transformer model with the Paillier cryptosystem to achieve a more robust and resilient secure framework for deep learning.

By taking into account the variation in measurements at the following moments, Qu et al. [29] presented a Hellinger-distance-based FDIA detection method where empirical modal decomposition (EMD) is used for filtering the irrelevant values from the recorded measurements of the state estimation process. The FDIA detection capability is further improved by using the image transform (IT) algorithms to deal with variations in measurements. It enables FDIA detection by comparing the run-time Hellinger distance with

the threshold value. It achieves high accuracy on the tested IEEE-14 bus systems simulation environment, but takes more time in detection, making it impractical for real-time FDIAs detection.

A data-driven framework to perform the system state estimation is proposed by Hallaji et al. [30] to detect and classify unobservable FDIAs by retrieving the control signal using the logged measurements. The proposed scheme achieved the highest accuracy of 97.4%, which is not sufficient for detecting FDI attacks in real-time systems. Aboelwafa et al. [31] employed a deep neural network using autoencoders with a backpropagation method to encode and decode different voltage and current measurements in the smart grid. Although this model performs well, it takes a large time to train the model. They overcome this extensive time consumed during the training process by considering an extreme learning machine (ELM) based fast forward neural network model to detect FDIAs. This model was not efficient in detecting FDIAs on some nodes.

Additionally, denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks primarily target the communication infrastructure within the smart grid, affecting smart meters, sensors, and state estimators. Defense mechanisms against these attacks include the use of encryption methods, network topology optimization, and advanced metering infrastructure [23]. Techniques such as digital twin models and reinforcement learning have been proposed to simulate these attacks in controlled environments to develop effective mitigation strategies.

For deception attacks, researchers have explored graph theoretical approaches and multilayer neural network frameworks to improve detection accuracy. These frameworks leverage real-time data analysis and anomaly detection to identify malicious activities within the Cyber-Physical Power Systems [32]. These techniques are still in development, but hold a lot of potential. They could allow for a holistic, integrated system to protect power systems from astute assaults. Moreover, recent breakthroughs indicate that the application of machine learning in combination with blockchain technology could greatly improve data integrity in advanced metering infrastructure, a key component of smart grids [22].

Recent research shows how imperative it is to enhance our detection methods in order to cope with rapidly evolving cyber threats. This work proposes an federated deep learning-based architecture to provide a decentralized, scalable solution for detecting FDIAs. Our architecture not only improves the accuracy of smart grid cyber event detection, but also their resilience against a whole host of different kinds of cyberattacks.

3. Methodology

3.1. Attack Model

The attack model describes the attacker's capabilities and access in the context of a FDIA on CPPSs. It is essential that the attacker has a knowledge of the data processing techniques and network architecture of the power grid. The attacker can use this information to find vulnerabilities and possible entry points in the system. Once the attacker gains access to the network by compromising sensors and communication link, he can inject bogus data into the network. The attacker aims to influence the power grid's state estimation mechanism, which is utilized for monitoring and controlling activities, by inserting falsified data.

Figure 1 illustrates a CPPS under attack from FDIAs. It depicts the architecture where an attacker introduces false data into a central database, altering key measurements such as current, voltage phase angles, and magnitude recorded by PMUs. The false data, once injected, misguides the control center's decision-making process, leading to potential operational disruptions in computing, control, and the physical power grid, including generation, transmission, and distribution. This schematic highlights the vulnerabilities in CPPSs, and underscores the critical need for advanced detection and mitigation strategies to ensure grid stability and security. FDIAs in power systems can be broadly categorized into several types based on their characteristics and methods of execution.

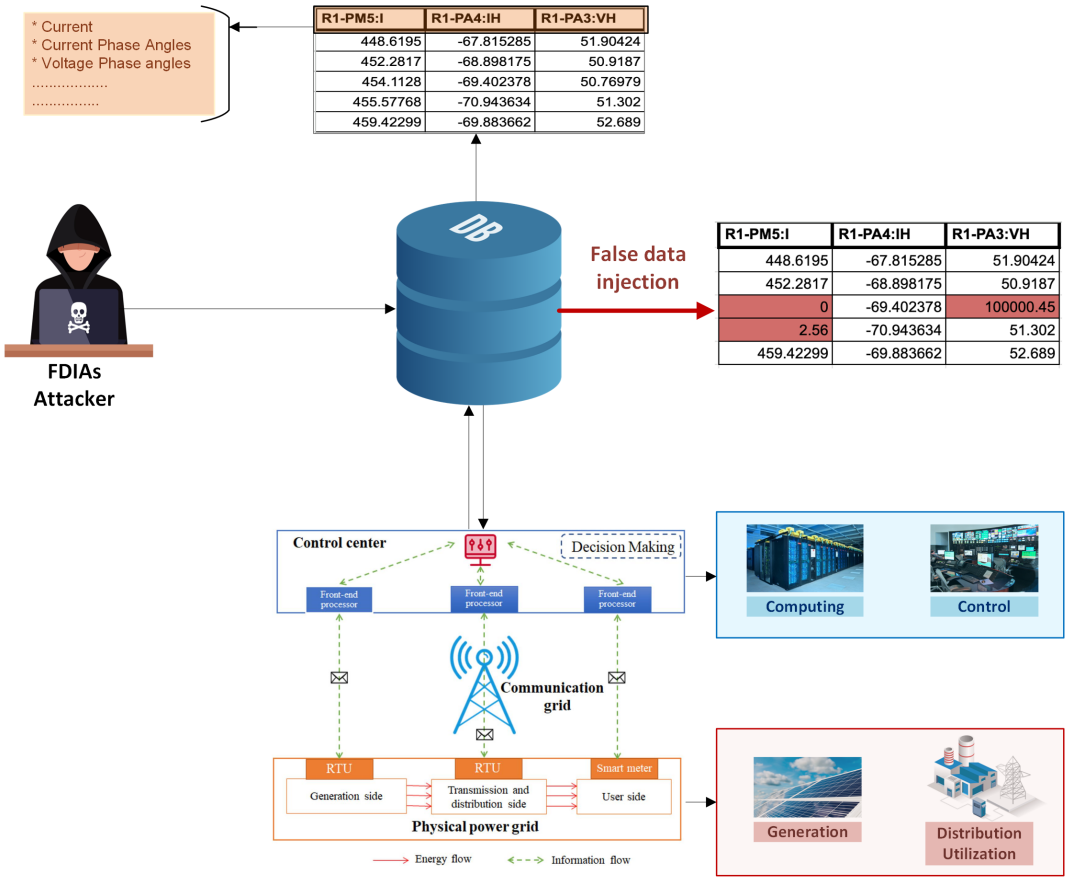


Figure 1. False Data Injection Attack in CPPS.

3.1.1. Remote-Tripping Command Injection

This attack requires the attacker to penetrate the system perimeter defense, where a command could be sent to a relay, resulting in the opening of a breaker. It can be classified into two types: Single Relay Attacks, which involve sending malicious commands to individual relays to trip breakers, and multiple relay attacks, which include sending commands to multiple relays simultaneously. This technique is implemented by inserting false commands into the relay control systems, simulating an external attack where commands are injected remotely.

3.1.2. Relay Setting Change Attack

Relays are configured with a distance protection scheme, and the attacker modifies the settings of a relay to prevent it from tripping when there is a valid fault during this attack. This can be done by changing the relay's threshold values or entirely disabling its trip function. This includes scenarios where relays are disabled for faults occurring at various percentages along the lines. It is divided into two types: (1) single relay disabled, and (2) multiple relays disabled. With multiple relays disabled, it becomes more complex to detect, and has a higher impact as it affects broader grid sections.

3.1.3. Data Injection Attack

These attacks involve altering measurement data such as current, voltage, and sequence components to simulate faults. This misleads operators, and can cause blackouts. This involves modifying the dataset values to mimic actual faults. For instance, voltage and current readings are manipulated to indicate a fault where none exists, blinding the operators.

Incorrect operational decisions brought about by such an attack can have far-reaching consequences, including but not limited to widespread power outages and inefficient pro-

cesses. The challenge for defense is to improve their strategies for detection and response. This involves protecting communication lines against unauthorized access, boosting measurement data redundancy for cross-verification purposes, and using advanced machine learning algorithms for anomaly detection.

3.2. Federated Learning Scheme

The primary objective of this research is to detect FDIAs in power systems using various machine learning and deep learning algorithms, and to compare their performance within a federated learning framework. Our research aims to develop a robust, scalable, and privacy-preserving approach to FDIA detection in modern CPPSs.

In the decentralized machine learning approach known as federated learning, numerous clients work together under the leadership of a central server to train a collaborative model. Their data remain local and secure and are never shared. This technique is particularly efficient for scenarios where data privacy is significant, like in CPPSs, and data cannot be shared directly due to regulatory or privacy constraints.

As depicted in Figure 2, we employed a federated learning configuration in our study to identify FDIAs. The main concept is to use a federated learning framework so that different clients can work together to train a global model without revealing their local data. To guarantee the security and privacy of the data, the connections between the main server and the clients are made using encrypted channels. A local model is trained by each client using its data, and the model updates (weights) are periodically transmitted to the central server. The global model is updated when the central server receives updates to the local models and applies a weighted aggregation to them.

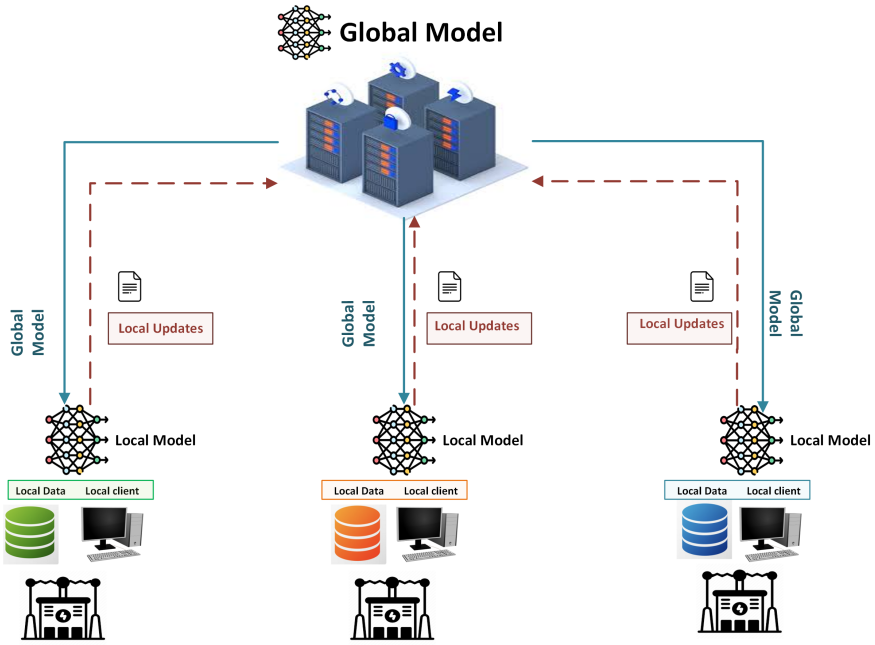


Figure 2. Federated Learning Architecture.

There are three steps to the Federated Averaging (FedAvg) algorithm's operation: selecting clients, updating models, and aggregating the results. Each iteration of the client selection phase involves picking a subset of potential clients to take part in the training. In the model update phase, each client that was chosen trains the model using its local dataset, and then sends the updated weights to the server. A new global model is formed during the aggregation phase when the server collects the weights from the clients. The quantity of training samples utilized by each client determines the weighting of the aggregation.

The Federated Learning process in our model is described by the following steps.

Each client k (where $k = 1, 2, \ldots, K$) trains its local model w_k by minimizing a local loss function $\mathcal{L}_k(w)$.

$$w_k^{(t+1)} = w_k^{(t)} - \eta \nabla \mathcal{L}_k(w_k^{(t)}),$$

$\nabla \mathcal{L}_k(w_k^{(t)})$ is the gradient of the loss function with respect to the model parameters, η is the learning rate, and $w_k^{(t)}$ denotes the model parameters at iteration t.

After training on its local dataset, each client sends its updated model weights to the central server.

$$\text{Client } k \text{ sends } w_k^{(t+1)} \text{ to the central server.}$$

The central server aggregates the received local models to update the global model $w^{(t+1)}$. The aggregation is typically a weighted average based on the number of data samples n_k at each client:

$$w^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{N} w_k^{(t+1)},$$

where the total amount of data samples across all clients is denoted as $N = \sum_{k=1}^{K} n_k$, and $\frac{n_k}{N}$ represents the weight assigned to the model from client k. For every training round, a subset $\mathcal{S}^{(t)}$ of clients are chosen to take part in the training.

$$\mathcal{S}^{(t)} \subseteq \{1, 2, \ldots, K\}.$$

The global model is updated at the server by aggregating the local models from the selected clients:

$$w^{(t+1)} = \sum_{k \in \mathcal{S}^{(t)}} \frac{n_k}{N^{(t)}} w_k^{(t+1)},$$

where $N^{(t)} = \sum_{k \in \mathcal{S}^{(t)}} n_k$ is the total number of data samples from the selected clients.

The process repeats until the model converges, typically when the change in the global model between consecutive rounds is below a certain threshold ϵ:

$$\| w^{(t+1)} - w^{(t)} \| < \epsilon.$$

3.3. Dataset Description

The 2 Classes Power Systems dataset used for this research is provided by Mississippi State University and Oak Ridge National Laboratory, which is collected from multiple interconnected stations, each equipped with PMUs and Intelligent Electronic Devices (IEDs) [33]. A PMU is a sensing device that uses time stamping enabled by the global positioning system to capture power system phasors synchronously over large geographic areas with great precision in a smart grid.

Dataset Composition and Features

The dataset comprises 37 different power system event scenarios, including both natural events (e.g., short-circuit faults, line maintenance) and malicious attack events (e.g., false data injection, remote tripping commands). These scenarios are represented in 15 CSV files, which contain a total of 78,377 rows and 129 feature columns after combining the files. Each file contains measurements recorded by PMUs during various types of events, including natural, normal, and attack events.

The dataset features measurements from PMUs and control panel logs, capturing various electrical parameters critical for power system monitoring and analysis. The configurations mirror a real-world power grid topology, with relays controlling generators, transmission lines, and circuit breakers. Each PMU captures 29 distinct measurements, including voltage and current phase angles and magnitudes, sequence components, frequency, and impedance data. These high-fidelity measurements are collected across four PMUs, resulting in 116 feature columns. Additional data from control panel logs, Snort

alerts, and relay logs supplement the PMU data, creating a comprehensive dataset that encapsulates the power system's physical state and potential cyber intrusions. These data are collected from 37 different power system event scenarios, comprising 15 CSV files. Each file contains measurements recorded by PMUs during various types of events, including natural, normal, and attack events.

The dataset includes critical features such as voltage and current phase angles, voltage and current magnitudes, frequency, and relay status. Voltage and current phase angles provide information on the phase difference between voltage and current, which is crucial for detecting anomalies in the power system. Voltage and current magnitude values help in assessing the overall health and stability of the power system by indicating the actual power being consumed and generated. Frequency measures the stability of the power system, as deviations from the standard frequency can indicate issues. Relay status includes information on the operational state of protective relays, which are essential for safeguarding the power system against faults and abnormal conditions.

The system is subjected to various scenarios, including natural events (e.g., short-circuit faults, line maintenance) and malicious attacks (e.g., data injection, remote tripping command injection, relay setting changes). These scenarios are crafted to represent realistic operational conditions and potential security threats. Different types of FDIAs described in Section 3.1 are carefully simulated in the Power System dataset provided by Mississippi State University and Oak Ridge National Laboratory.

The box plots shown in Figure 3 provide a graphical comparison between natural and attack events for different vital features, including voltage magnitude, current magnitudes, and voltage phase angles. The voltage magnitudes, median values, interquartile range (IQR), and whiskers are almost the same for both natural and attack events. However, there are some outliers in the attack events, but not significantly more than in the natural events, which show irregular voltage behavior during attack events.

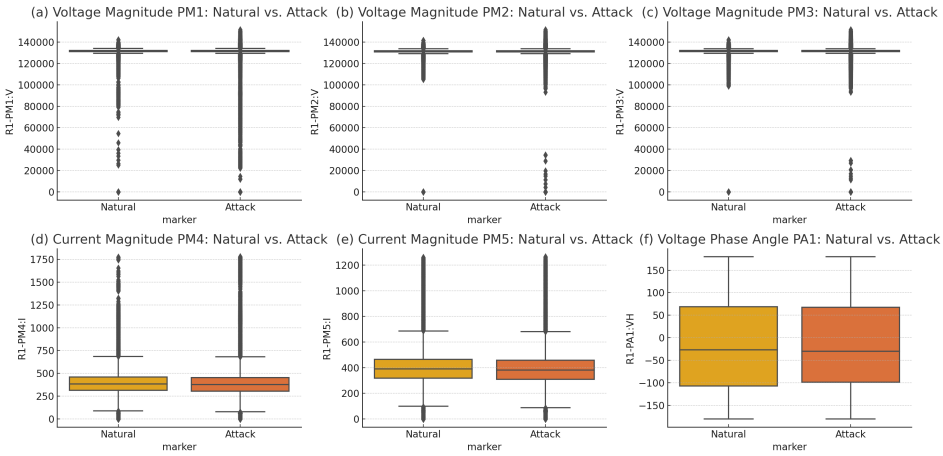
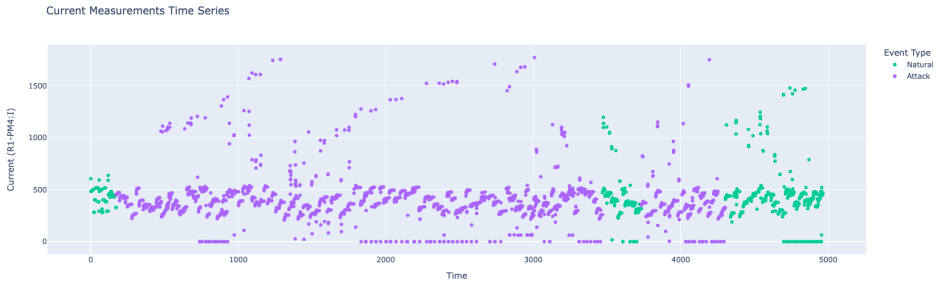


Figure 3. Box Plot.

Like voltage magnitudes, current magnitudes show no significant changes in the median, IQR, and whisker values of attack and natural events. The current magnitude of attack events has more outliers. The phase angle box plot shows no noticeable difference in variability between natural and attack events.

To visualize the behavior of electrical parameters over time, we generated time-series plots of the dataset. For instance, we added a 'time' column to create a time-series visualization of the R1-PM4:I column, which represents the Phase A current phase magnitude measured by PMU R1, and the R1-PM1:V column, which represents the voltage magnitude measured by the PMU R1 at a specific location in the power system. The color-coded scatter plot of the distribution of current is shown in Figure 4a. Attack events (purple) appear more frequently throughout the time series. There are some periods with clusters of natural events (green), but they are less common, so during the data cleaning process, we will need

to balance the data. The presence of many zero or near-zero values during attacks suggests possible data manipulation or sensor tampering.



(**a**)

(**b**)

Figure 4. Current Voltage distributions. (**a**) Distribution of Current (R1-PM4:I). (**b**) Distribution of Voltage Magnitude (R1-PM1:V).

The color-coded scatter plot of voltage distribution to distinguish between natural events and attacks is shown in Figure 4. Natural events (green points) cluster tightly within a certain range, indicating stable system operation under normal conditions. Attacks (purple points) are associated with sudden, significant voltage drops. These drops range from minor (around 100 k) to severe (as low as 20 k–40 k). Some days exhibit more attack events than others, suggesting possible strategic timing by attackers. Most voltage drops are short-lived, appearing as sharp spikes downward in the line graph. The scatter plot again reveals that attack events (purple) outnumber natural events (green) highlighting the need for balancing the data.

3.4. Data Preprocessing

First, all 15 CSV files were combined into one dataset, which had 78,377 rows and 129 columns. During the preprocessing phase, impedance columns were removed from the dataset as they were not deemed relevant for detecting FDIAs. These columns did not provide significant information for the anomaly detection models and were thus excluded to streamline the dataset. Then, Infinity values (both positive and negative) were replaced with NaNs. This is essential because infinity values can disrupt the training process of machine learning models, leading to unstable and unreliable results. Followed by that, NaN values were then imputed with the mean of their respective columns. This method ensures that the dataset remains intact without significant data loss, preserving the statistical properties of each feature.

The numerical features were standardized using the Scikit-Learn StandardScaler. This was done in order to preserve the data's statistical properties, which are necessary for it to be viable as input for algorithms such as logistic regression and neural networks. The formula for standardization is given by

$$z = \frac{(x - \mu)}{\sigma} \tag{1}$$

where z is the standardized value, x is the original value, μ is the mean of the feature, and σ is the standard deviation of the feature. Standardization was chosen rather than normalization because it better retains the original statistical properties of the data. It is also more appropriate for learning algorithms like logistic regression and neural networks, which function best when the data are normally distributed.

The numerical features, excluding the log columns, were standardized using the StandardScaler from sklearn. Standardization was preferred over normalization, because it retains the statistical properties of the data and is more suitable for algorithms like logistic regression and neural networks, which assume that the data are normally distributed. Normalization, which scales the data to a range of [0,1], would not be appropriate here, as it could distort the underlying distribution of the data, especially when there are outliers like in falsified injected data. The categorical labels in the 'marker' column were one-hot encoded. One-hot encoding converts categorical variables into a binary matrix, which allows machine learning algorithms to process categorical data efficiently. This step is crucial for models that require numerical input.

The marker column initially has categorical labels 'Natural' and 'Attack'. With one-hot encoding, the Marker column is transformed into two binary columns with each row having '1.0' in the column corresponding to its category and '0.0' in the others as shown in Table 1.

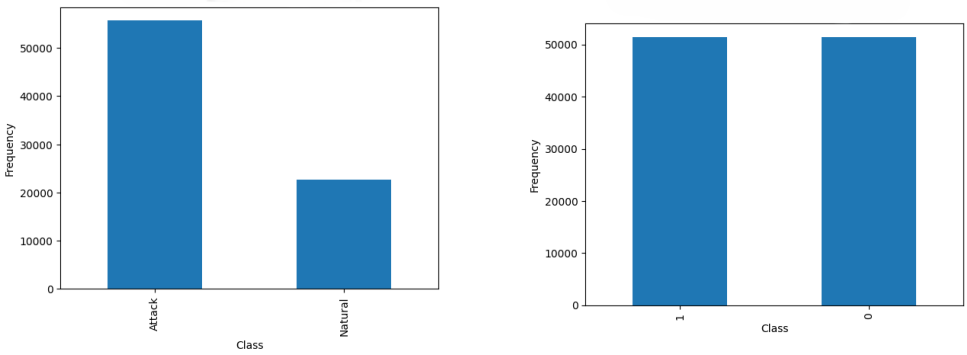
Table 1. One-Hot Encoded Labels in the 'Marker' Column.

Marker_Attack	Marker_Natural
0.0	1.0
1.0	0.0

The performance of machine learning models can be impacted, often negatively, by class imbalance. When a model is trained on data with a class imbalance, it often produces predictions that are biased toward the majority class. To counteract this effect, we used the Synthetic Minority Over-sampling Technique, commonly known as SMOTE. This is a well-known technique in the field of class imbalance that generates synthetic samples in the minority class.

In our preprocessing pipeline, we applied SMOTE to ensure that the dataset was balanced before training our machine-learning models.

Initially, there were 55,663 attack and 22,714 natural events, as shown in Figure 5a. After applying SMOTE, the distribution of natural and attack events is shown in Figure 5b.



(**a**) (**b**)

Figure 5. Distribution of Events Before and After Class Balancing Using SMOTE. (**a**) Distribution of Natural Vs Attack events before class balancing. (**b**) Class distribution after Oversampling using SMOTE.

3.4.1. Data Splitting Strategy

In the development of centralized machine learning models, the dataset is divided into training and testing subsets to evaluate the model's performance effectively. We employ the train-test split method, which ensures that the model's training is independent of the data it will be evaluated on, thus minimizing the risk of overfitting and providing a more realistic assessment of its performance.

In this study, we use an 80-20 split, where 80% of the data are allocated to the training set ($X_{\text{train}}, y_{\text{train}}$) and 20% to the test set ($X_{\text{test}}, y_{\text{test}}$). This ratio is a standard practice that offers a good balance between training the model with a substantial amount of data while still reserving a significant portion for unbiased evaluation. We specify a random_state of 21 to ensure the reproducibility of our results. Using a fixed random state guarantees that the data split remains consistent across different runs, facilitating the comparison of results and enhancing the study's reliability.

The federated learning framework partitions the dataset into multiple subsets that reproduce the individual local datasets of the clients. In this way, each client works on its own localized subset of the data; this delegates a unique, non-IID (non-independent and identically distributed) data scenario to each client. Each client's local dataset is used to train its model instance, with no direct access to any of the data held by other clients.

In federated training, each participating client trains its own local model on its own partitioned dataset. The training of local models occurs in parallel across all clients. Then, using the FedAvg algorithm, the updates to the local models are aggregated at a central server to produce an updated global model. Since only the model updates (gradients) are sent to the server, this approach does a good job of preserving the clients' data privacy.

After the federated training rounds are finished, the global model is assessed using the entire dataset. This dataset, which functions as the test set, enables the researchers to judge the generalization performance of the global model. They compute various metrics—accuracy, precision, recall, and F1 score—to obtain a clearer picture of how well the model is likely to perform across the very different, non-IID data distributions it encountered during training.

3.4.2. Degree of FDIAs in the Dataset

The dataset used in this research includes both natural events and attack events, with a total of 55,663 instances of attack events (representing approximately 70% of the data) and 22,714 instances of natural events (approximately 30%). The false data injection attacks were simulated to affect various components of the power system, such as voltage and current measurements, frequency, and relay statuses, to create realistic attack scenarios. The attack scenarios simulated in the dataset include data injection attacks, remote tripping command injection, and relay setting changes. Data injection attacks involve altering sensor readings to mislead operators, thereby affecting operational decisions. Remote tripping command injection involves sending unauthorized commands to relays to manipulate the state of the power system. Relay setting changes attack scenario update the relay settings to disable protective measures during valid faults, creating vulnerabilities.

4. Models Implementation and Evaluation

The selection of machine learning and deep learning algorithms for detecting FDIAs was based on their proven effectiveness in classification tasks and their ability to handle complex data patterns. Initially, we trained several centralized models, including LSTM, GRU, and Simple RNN, to establish baseline performance metrics. Given the results, we transitioned to a federated learning approach for improved data privacy and security. Specifically, we chose to further investigate the performance of Bidirectional LSTM and Bidirectional GRU models with Attention mechanisms within the federated learning framework.

Table 2 describes the machine learning models selected for centralized training, and Table 3 describes the deep learning models selected for centralized training.

Table 2. Machine Learning Models for Centralized Training.

Machine Learning Models	Description
ExtraTrees Classifier	Known for robustness and ability to handle high-dimensional data, it creates multiple trees in a forest and averages the results, reducing overfitting.
XGBoost	Highly efficient and widely used for its performance in classification tasks.
Random Forest	Utilizes ensemble learning by constructing multiple decision trees and outputting the mode of the classes.
Logistic Regression	A fundamental algorithm for binary classification, useful for its interpretability.
Decision Tree	Developing a model that can learn decision rules to forecast the value of a target variable is a simple yet successful classification strategy.
K-Nearest Neighbors (KNN)	Finds the majority class among the k-nearest data points; it is a non-parametric classification method.

Table 3. Deep Learning Models for Centralized Training.

Deep Learning Models	Description
Simple Recurrent Neural Network (RNN)	Adaptable to time-series data; capable of learning temporal relationships.
Long Short-Term Memory (LSTM)	An improved RNN with the ability to overcome the vanishing gradient problem and capture long-term dependencies.
Gated Recurrent Unit (GRU)	Similar to LSTM but with a simplified architecture with two gates, update and reset, making it faster and computationally efficient.
Generalized Deep Learning Network (GDNN)	Custom deep learning model designed to adapt to specific data patterns.

Each machine-learning algorithm is implemented using version 1.2.2 of the Scikit-learn library in Python 3.12.7. The models were trained on the preprocessed dataset, and the deep learning models were implemented using TensorFlow version 2.16.1 and Keras version 3.4.1.

4.1. Model Selection Rationale

Recurrent models such as RNNs, LSTMs, and GRUs makes them robust and widely used for time-series and sequential data processing. Their capacity to capture temporal interdependence is the main reason for their popularity. This section discusses what conceivably makes them work and, more importantly, why that is relevant for analyzing FDIA detection.

Recurrent Neural Networks: An RNN updates its hidden state h_t using

$$h_t = \tanh(W_{ih} \cdot x_t + W_{hh} \cdot h_{t-1} + b_h) \quad (2)$$

where W_{ih} and W_{hh} are weight matrices for the input and hidden state, respectively. RNNs can capture the long-term dependencies in a signal needed to detect anomalies in a temporal signal. However, RNNs can learn only short sequences. When they are compelled to learn long sequences, they suffer from the vanishing gradient. This is why we use more advanced architectures, such as LSTMs or GRUs. A simple RNN model for centralized implementation consists of multiple RNN layers followed by dense layers. The RNN layers

had a hidden state size of 128, and the dense layers used ReLU activation with a final softmax layer for classification.

Long Short-Term Memory (LSTM): The vanishing gradient problem is met by LSTMs with a pair of gated mechanisms. They can be thought of as a cell that is divided into two parts with an internal gated structure that controls the flow of information and the cell's output, which is governed by the following equations.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(Forget gate)} \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(Input gate)} \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(Output gate)} \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad \text{(Cell state)} \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

where z_i is the input to the softmax function for class i, and C is the total number of classes. The LSTM forget gate, f_t, controls how much of the past state is retained. The input gate, i_t, manages new input, while the output gate, o_t, manages output at each step.

In the fully LSTM-based architecture for classification, a softmax activation function is used at the last output layer to distribute the model's outputs as probabilities among the possible classes. The softmax function is defined as

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \quad i = 1, \ldots, C \tag{8}$$

The LSTM model for centralized and federated learning settings includes LSTM layers with 128 units, followed by dropout layers to prevent overfitting. Dense layers with ReLU activation and a final softmax layer were used for output. LSTMs are ideal for FDIA detection because they can learn long-term dependencies in the "time series" of smart grid measurements.

Gated Recurrent Unit (GRU): The GRU makes the LSTM easier to understand and use by combining the input and forget gates into a single update gate. The GRU's state transition is equal to the combination of the cell and hidden states of the LSTM. The LSTM has two states: the cell state c_t and the hidden state h_t. The GRU has a single state h_t that behaves like the combination of c_t and h_t.

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \quad \text{(Reset gate)} \tag{9}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \quad \text{(Update gate)} \tag{10}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t] + b) \quad \text{(Candidate state)} \tag{11}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{12}$$

Compared with LSTM, GRUs converge faster, need fewer parameters, and are thus more computationally efficient. This is especially beneficial in the context of real-time detection of FDIAs in power systems. The GRU model has GRU layers with 128 units, similar to the LSTM architecture, but without the cell state mechanism for centralized and federated learning implementation

Generalized Deep Neural Networks (GDNNs): GDNNs expand upon standard deep learning models by injecting multiple dense layers with consistent normalization and dropout for better generalization. The forward pass through a dense layer with batch normalization and dropout is represented as

$$h^{(l)} = \text{Dropout}(\text{BatchNorm}(\text{ReLU}(W^{(l)} \cdot h^{(l-1)} + b^{(l)}))) \quad \text{(Dense layer with regularization)} \tag{13}$$

where $h^{(l)}$ is the output of the l-th layer, $W^{(l)}$ and $b^{(l)}$ are the weight matrix and bias for that layer, and $\text{ReLU}(x) = \max(0, x)$ is the activation function.

The final output layer uses a softmax activation function,

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \quad i = 1, \ldots, C \tag{14}$$

where z_i is the input to the softmax layer for the i-th class, and C is the total number of classes. The outputs are the sum of all class probabilities, which are determined by utilizing the softmax function. Learning intricate structures in high-dimensional data are a strength of GDNNs. This quality makes them good candidates for detecting FDIAs in the smart grid, which comprises a large, diverse set of datasets. GDNN model for centralized implementation was designed with several dense layers, each followed by batch normalization and dropout layers to enhance generalization. The final layer used softmax activation for classification.

Detecting FDIA necessitates models that can grasp both the short-term and long-term aspects of sequential data that exhibit anomalies. While RNNs lay the groundwork by capturing the temporal dependencies, their limitations make using LSTMs and GRUs for this task very desirable. LSTMs do an excellent job of capturing the long-term dependencies, while GRUs are a more computationally efficient alternative that do much the same thing. GDNNs are also employed, learning complex non-linear patterns in high-dimensional data and thus augmenting the detection framework's overall robustness.

4.2. Federated Deep Learning Models

For the federated learning, we decided to experiment various deep learning models that performed best as centralized and also further strengthened the model by changing the architecture and adding specific layers. Those models include Long short-term memory (LSTM), Gated-Recurrent Unit (GRU), LSTM with Attention Layer (LSTM-AL), GRU with Attention Layer (GRU-AL), Bidirectional LSTM with Attention Layer (Bi-LSTM-AL), Bidirectional GRU with Attention Layer (Bi-GRU-AL), and Transformer.

Bidirectional LSTM is an extension of the standard LSTM that involves duplicating the first recurrent layer in the network so that there are now two layers side by side, each of which outputs to the same next layer. One LSTM layer reads the input sequence from start to end, while the other reads it from end to start. This approach helps the model capture information from both past and future states for a given time frame, leading to improved performance in tasks involving sequential data.

The Attention Layer is a mechanism that allows the model to focus on specific parts of the input sequence, effectively identifying and giving more importance to relevant information while ignoring irrelevant parts. This layer calculates a weighted sum of all hidden states to focus on parts of the input sequence that are most relevant for the prediction task at each time step.

The Bidirectional GRU (BiGRU) operates similarly to the BiLSTM, but uses Gated Recurrent Units instead of LSTM units. GRUs are a variant of RNNs that aim to solve the vanishing gradient problem and have fewer parameters compared to LSTMs. The key differences between these advanced models and the simple LSTM or GRU models are as follows.

Bidirectional Mechanism: unlike the simple LSTM and GRU, which only process input data in one direction (forward), the bidirectional versions process data in both directions (forward and backward), enabling the capture of both past and future context.
Attention Layer: the inclusion of the Attention Layer in the bidirectional models allows for a dynamic focus on the most relevant parts of the input sequence, which is particularly useful for tasks requiring a deep understanding of specific segments of the input data.
Enhanced Performance: the combination of bidirectional processing and attention mechanism generally leads to better performance and accuracy, especially in complex sequence prediction tasks.

4.3. Model Architecture

The architecture of the Bidirectional GRU and LSTM with Attention Layer used in our research is depicted in Figure 6. The model starts with an input layer that accepts sequences of length 129. The input is then reshaped to match the requirements of the GRU layer. A Bidirectional GRU layer follows, which processes the input sequence in both forward and backward directions. This helps in capturing dependencies from both past and future contexts within the input data. The output of this layer is a sequence with 128 features. The attention mechanism is applied to the output of the Bidirectional GRU. This layer allows the model to focus on the most relevant parts of the input sequence by calculating attention scores for each time step. The output from the Attention Layer is then passed through a dense layer with 64 units to reduce the dimensionality and introduce non-linearity. A dropout layer with a dropout rate of 50% is used to prevent overfitting by randomly setting a fraction of input units to zero during training. Finally, the output layer is a dense layer with a single unit, which provides the final prediction.

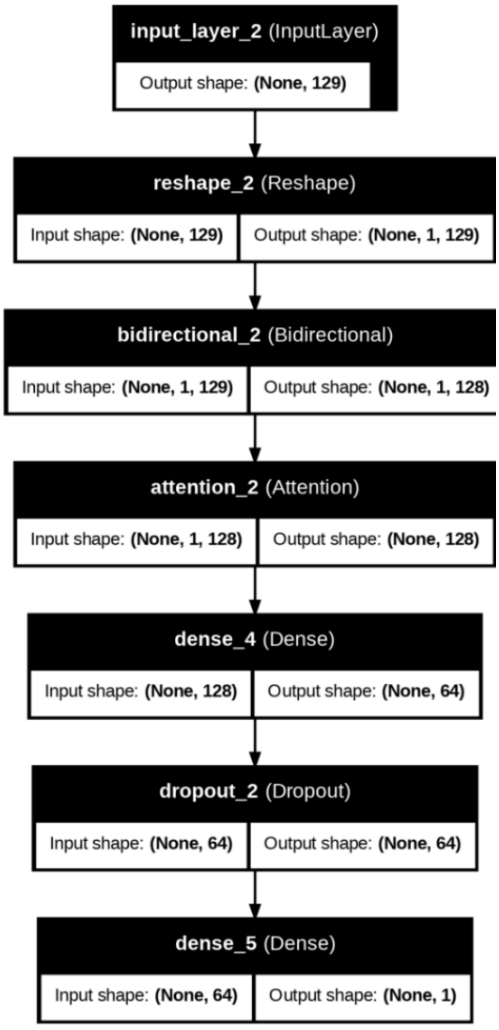


Figure 6. Bidirectional GRU/LSTM with attention layer Model.

4.4. Model Training Strategies

We trained various centralized models—like the ExtraTrees Classifier, XGBoost, and Random Forest—along with deep learning models—including the long short-term memory network, gated recurrent unit, and Transformer—to work with a labeled dataset of normal and FDIA-affected data. Our aim was to have these models learn the identifying features of false data injection. In the federated learning structure, we guaranteed that each client

trained on its subset of data in order to preserve privacy. The global model aggregated the updates from individual clients so that it could, in effect, achieve the same result as if it had direct access to the complete dataset.

The models for Bi-LSTM, Bi-GRU, and Bi-GRU with Attention Layers were constructed to concentrate on the most relevant parts of the input data when detecting FDIAs. The attention mechanism in these models calculates a weighted sum of the hidden states such that the important information is enhanced, and the irrelevant parts are ignored. Concentrating on pertinent data portions enables the models to pick up on small alterations caused by the infusion of fake data. They do this by using the relevant portions of the model's working memory when being trained on a correct version of the task. In turn, this leads to improved detection rates.

We took a stepwise approach to training the models, beginning with the simplest possible architectures and moving to more complex systems, like Bidirectional LSTM and GRU, as our understanding of the sequential data improved. This approach allowed us not only to better understand how the models work, but also how to troubleshoot faults in them when capturing that complex layer of patterns in the sequential data. Regularization techniques are employed, including the use of dropout layers, to improve the stability and performance of the models. We chose dropout because it is a simple and effective technique to use when training deep neural networks. Dropout is "applied" to a layer of neurons in the network at the time of training. When it is "applied", a random subset of the layer's neurons are temporarily turned off or deactivated, as if the model is being trained with two different architectures. The Bidirectional LSTM and GRU model architectures, illustrated in Figure 6, begin with an input layer that takes in sequences of 129. An input layer first accepts the data, which is then passed to a Bidirectional GRU layer that reads the information both forward and backward. This layer captures the spatiotemporal dependencies of data from both past and future contexts.

The attention mechanism subsequently processes the outputs of the Bidirectional GRU to give priority to the features that are most relevant for detecting FDIAs. The architecture of the model concludes with dense layers and a dropout layer that work together to prevent overfitting. The combination of these three distinct yet complementary components guarantees that the machine learning techniques will identify the FDIAs with a high degree of accuracy, and that they will possess the adaptability necessary to address a variety of different attack patterns in real-time, thus assuring their robustness.

4.5. Experimental Setup

To evaluate the effectiveness of the federated learning framework in detecting FDIAs, we conducted a series of experiments involving multiple clients and various configurations. Each experiment aimed to assess the model's performance under different conditions and attack scenarios. The experiments were conducted on a system equipped with a 13th Gen Intel(R) Core(TM) i7-13700H processor running at 2.90 GHz, 64 GB of RAM, an NVIDIA GeForce RTX 4060 GPU, and the Windows 11 operating system.

The implementation uses TensorFlow (TF) version 2.16.1, TensorFlow Federated (TFF) version 0.82.0, Pandas version 2.2.2, NumPy version 1.26.4, and Scikit-Learn library version 1.2.2.

Five clients are simulated, each with its own subset of the data. The data distribution among clients is designed to mimic real-world scenarios where data are non-Independent and Identically Distributed (non-IID).

For our federated learning framework, we implemented and evaluated several models. A GRU model was implemented with two GRU layers, followed by dense layers. An LSTM model with a similar architecture was used for comparison. For both GRU and LSTM, Attention Layers were added to focus on important features. Bidirectional GRU and LSTM models were also implemented to capture dependencies in both forward and backward directions. A Transformer-based model was employed for comparison, utilizing multi-head attention mechanisms. The models were trained with Learning Rate of 0.001, Batch Size of

64, number of rounds ranges form 200–300. Adam optimizer was used for both client-side and server-side model updates. The Adam optimizer adaptively modifies the learning rates according to the gradient's moments, using these equations to carry out the updates:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla L(\theta_t) \tag{15}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla L(\theta_t))^2 \tag{16}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{17}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{18}$$

The Adam optimizer adjusts learning rates for swifter convergence, and it does so dynamically. The way it accomplishes this involves both the first and second moments of the gradient.

The task is to perform binary classification to distinguish between normal operations and FDIA attack. Hence, we used binary cross-entropy as the loss function. This function, when implemented as a scoring rule, yields good results. It yields good results because it measures well the disagreement between the predicted probabilities and the true binary labels. The function makes it so that the models we trained really aim for probabilities that are either close to 0 (for normal operations) or to 1 (for attacks). Mathematically, the binary cross-entropy loss for a given prediction $\hat{y}$ and true label y is

$$L(y, \hat{y}) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \tag{19}$$

A combination of grid search and empirical testing was employed to adjust the hyper-parameters—learning rate, batch size, and number of training rounds—of the configuration used in the models. The models themselves were evaluated on a subset of the data using cross-validation, and during training they used the following optimization update rule.

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) \tag{20}$$

Here, θ_t represents the model parameters at iteration t, η is the learning rate, and $\nabla L(\theta_t)$ is the gradient of the loss function with respect to the parameters. This update rule ensures efficient convergence of the model parameters.

4.6. Evaluation Metrics

The following metrics are essential for evaluating the performance of our models, particularly in the context of cyberattack detection. It is critical to avoid false positives, where the model incorrectly indicates that all is well, and missed detection, where it fails to identify a significant issue that is present.

Accuracy: The overall correctness of the model is indicated by its accuracy. It is a direct measure of how many total correct predictions a model makes when using it to solve a problem. Following the most common definition, the accuracy is expressed mathematically as the number of correct predictions divided by the total number of predictions made.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{21}$$

where TP is the True Positives (correct positive predictions), TN is the True Negatives (correct negative predictions), FP is the False Positives (incorrectly predicted positives), and FN is the False Negatives (incorrectly predicted negatives).

Precision: The precision of a model indicates the number of true positives divided by the number of predicted positives made by the model. In other words, it is a measure of the correctness of the model when it predicts a positive outcome.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{22}$$

Recall: The recall (sensitivity) metric assesses how well the model can find all true positive cases when identifying positive instances.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{23}$$

F1-Score: Offers the harmonic mean of precision and recall, making it a worthwhile metric in scenarios with uneven class distributions or when both metrics have equal import.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{24}$$

5. Results and Analysis

This section presents the experimental results obtained from our study. The results include the performance metrics of various machine learning and deep learning models both in centralized and federated learning frameworks.

The evaluation metrics results for centralized machine learning algorithms used to predicting "FDIAs" in the smart grid dataset are presented in Table 4. It shows that ExtraTrees classifier has highest accuracy, precision, F1 Score, recall and lowest RMSE as compared to other algorithms.

Table 4. Evaluation of the Centralized Machine Learning Models.

ML Models	Accuracy	F1-Score	Precision	Recall
ExtraTrees Classifier	0.94	0.94	0.94	0.94
XGBoost	0.84	0.87	0.9	0.91
Random Forest	0.92	0.94	0.94	0.94
Logistic Regression	0.71	0.69	0.72	0.83
Decision Tree	0.90	0.84	0.85	0.85
KNN	0.89	0.88	0.92	0.93

The Receiver Operator Characteristic (ROC) curves for machine learning algorithms are shown in Figure 7. The area under the curve (AUC) values for each model are also indicated, providing a clear comparison of their effectiveness. The AUC attempts to summarize the overall performance of a model in a single number that ranges from 0 to 1. An AUC of 1 means the model perfectly distinguishes between the positive and negative classes (yes, it perfectly discriminates), while an AUC of 0.5 indicates that the model has no real power at all in distinguishing between the classes, and is essentially a random classifier. Better model performance in distinguishing between classes is reflected in higher AUC values. The effectiveness of each FDIA detection model using a ROC curve was demonstrated. Random Forest (RF) and Extra Trees Classifier (ETC) resulted in nearly perfect classification performance. Conversely, Logistic Regression (LR) produced the least favorable result with an AUC value of 0.69. The use of ROC and AUC values provides a more rounded evaluation of the models when compared to simple accuracy statements.

The evaluation metrics results for centralized deep learning algorithms used to predicting "FDIAs" in the smart grid dataset are presented in Table 5.

Table 5. Performance metrics for various DL models (Centralized).

DL Models	Accuracy	F1-Score	Precision	Recall
Simple RNN	0.829	0.8218	0.8287	0.81
LSTM	0.99	0.99	0.99	0.99
GDNN	0.72	0.99	0.99	0.99
GRU	0.99	0.99	0.99	0.99

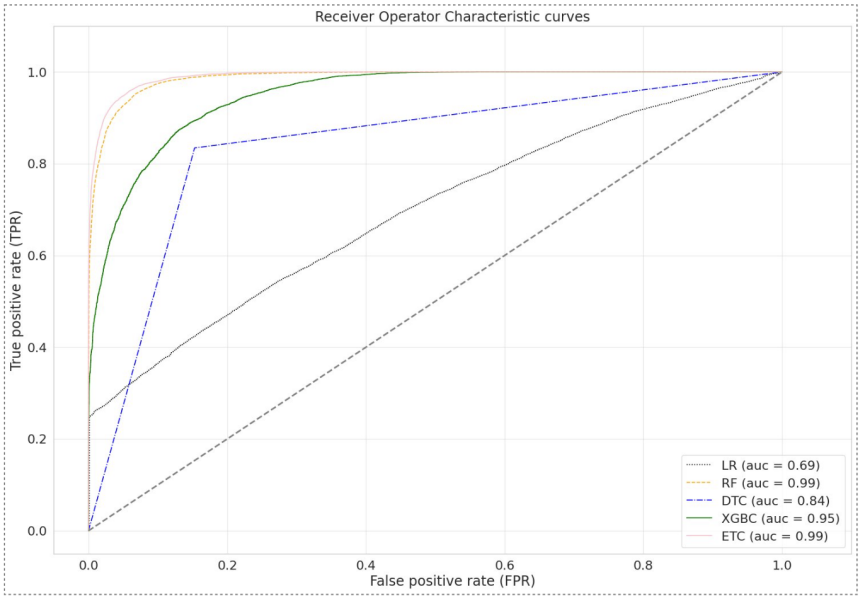


Figure 7. Receiver Operator Characteristic (ROC) curves of ML models.

Our experiments show that while the centralized learning models yield high accuracy and performance metrics, they, too, might be susceptible to FDIAs. Centralized models, such as the Extra Trees Classifier, appear very robust—this model achieved an accuracy of 0.94 and an F1 score of 0.94 (see Table 4). Despite these strong results, not only the Extra Trees Classifier, but also other deep learning models (e.g., RNN, GRU, LSTM), remain theoretically vulnerable to injection attacks. Although deep learning models like RNN, GRU, and LSTM have attained nearly flawless accuracy rates of 0.99 in standard settings, their possible vulnerability to FDIAs prompts worries about how well they would hold up in adversarial situations.

Centralized learning systems face a serious threat from false data injection attacks. Research show that these attacks take advantage of the poor defense mechanisms of certain machine learning models like RNN, GRU, and LSTM by injecting adversarial inputs that lead them to misclassify the data they are fed [34]. When these models are used for centralized learning, their architecture and virtual lack of redundant pathways make them not just vulnerable to adversarial inputs, but also a prime target for adversaries who want to introduce false data into a centralized system at a single point of entry.

Studies highlights that the lack of built-in defenses in centralized deep learning systems makes them vulnerable to adversarial attacks. This compromises the reliability of deep learning applications in a variety of settings, such as smart grids and Internet of Things systems [35]. The necessity for more considerable defenses is highlighted by these findings, because even the traditional machine learning and deep learning frameworks,

with their high accuracy, can be brought down by well-planned attacks. Our work points to the need for even more resilient defenses to better protect these frameworks.

With its decentralized method, federated learning holds potential for lessening some of our common security weaknesses. By spreading out the work of training smart algorithms, it lowers the chances that a localized attack will cause a serious drop in performance. This is important not just in power grids, but in any "critical infrastructure" where smart, adversary-resistant systems need to work with a high degree of accuracy and reliability. The detailed analysis of these results underscores the importance of adopting advanced defensive strategies like federated learning to safeguard against the ever-evolving landscape of cyber threats.

Centralized machine learning models often require aggregating data from multiple sources into a single repository, where the training and evaluation occur. This approach, while straightforward, exposes several vulnerabilities. Firstly, data centralization creates a single point of failure, making the model susceptible to data breaches and cyberattacks like FDIAs. Malicious attackers can potentially access sensitive information, leading to privacy violations. Furthermore, centralized learning necessitates the transfer of large volumes of data, which can be impractical due to bandwidth constraints and data sovereignty regulations. This can lead to increased latency and reduced efficiency, particularly in applications requiring real-time processing. The need for extensive data transfers also increases the risk of data corruption during transmission.

As discussed earlier, federated learning addresses these vulnerabilities by enabling model training across decentralized devices without sharing raw data. Instead, only model updates, such as gradients, are shared and aggregated. To evaluate the performance of federated learning models, we applied this approach to various deep learning algorithms including LSTM, GRU, LSTM with Attention Layer (LSTM-AL), GRU with Attention Layer (GRU-AL), Bidirectional GRU with Attention Layer (Bi-GRU-AL), Bidirectional LSTM with Attention Layer (Bi-LSTM-AL), and Transformer. The objective was to determine if federated learning could maintain or even improve evaluation metrics compared to centralized models.

The comparative analysis of the performance of these algorithms in federated learning settings is shown in Figure 8. Bi-GRU-AL shows the fastest increase in training accuracy, reaching nearly perfect accuracy within the first 100 rounds. It maintains high accuracy throughout the training process LSTM and GRU also perform well, achieving high training accuracy, but they take slightly longer to converge compared to Bi-GRU-AL. GRU-AL and LSTM-AL follow closely behind the Bidirectional GRU-AL, showing rapid improvements in accuracy, though they require a few more rounds to converge fully. Bidirectional LSTM-AL also achieves high training accuracy, but it shows slower convergence compared to the other models, taking more rounds to reach the same level of accuracy. All models eventually converge to a training accuracy close to 1.0, but the convergence rate varies, with Bi-GRU-AL performing the best.

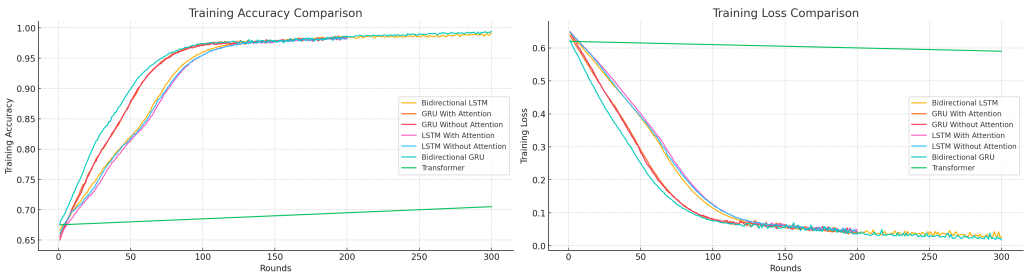


Figure 8. Comparison of Training Accuracy and Loss Across Various Federated Deep Learning Models.

Bi-GRU-AL and LSTM models exhibit the steepest decline in training loss, indicating that they quickly learn and fit the training data well. GRU also shows a rapid reduction in loss, though it slightly lags behind the Bidirectional models. LSTM and GRU-AL also

effectively reduce their training loss, but not as quickly as the Bidirectional models. All models achieve a very low training loss by the end of the training rounds, suggesting they fit the training data well. However, the Bidirectional models tend to reach lower loss values faster than others. This indicates a highly effective learning process, with the bidirectional nature and attention mechanism likely contributing to better context capture and focus on important features.

To provide a comprehensive comparison of the model performances, we evaluated the best-performing model using key metrics such as accuracy, loss, precision, recall, and F1 score on the testing dataset. Table 6 summarizes these evaluation metrics for different federated deep learning algorithms.

Table 6. Evaluation of the Federated Deep Learning Models.

Model	Test Accuracy	Test Loss	Precision	Recall	F1-Score
GRU	0.9879	0.0325	0.9783	0.9801	0.9792
LSTM	0.9891	0.0299	0.9763	0.9864	0.9813
GRU-AL	0.9939	0.0203	0.9931	0.9858	0.9894
LSTM-AL	0.9896	0.0284	0.9778	0.9865	0.9821
Bi-GRU-AL	0.9977	0.0070	0.9991	0.9928	0.9959
Bi-LSTM-AL	0.9956	0.0118	0.9957	0.9892	0.9924
Transformer	0.7278	0.5438	0.6470	0.1334	0.2212

Bi-LSTM-AL and Bi-GRU-AL models demonstrate the highest test accuracy 0.9957 and 0.9977, respectively, indicating superior predictive capabilities. GRU and LSTM models without Attention Layers exhibit slightly lower test accuracy, (0.9879 and 0.9891), showing that Attention Layers enhance the models' predictive power. Bi-LSTM-AL and Bi-GRU-AL have the lowest test losses (0.007 and 0.0118), reinforcing their reliability in making predictions. The Transformer model has the highest test loss (0.5438), indicating significant challenges in learning patterns in the data effectively. Bi-LSTM-AL and Bi-GRU-AL again lead in precision, recall, and F1-score, suggesting they not only predict accurately, but also balance False Positives and False Negatives effectively. The Transformer model lags significantly in all these metrics, particularly in recall (0.1334), highlighting its difficulty in identifying positive instances correctly. Models with Attention Layers and bidirectional configurations (Bi-GRU-AL, Bi-LSTM-AL) show faster convergence during training, achieving higher accuracy and lower loss over fewer rounds, as seen in the training metrics graph. This trend is consistent with the test metrics, where these models maintain high accuracy and low loss, indicating robustness and good generalization capabilities. Transformer model struggles significantly, suggesting it might require further tuning or might not be as well-suited for the specific federated learning tasks being evaluated. Models like Bi-GRU-AL and Bi-LSTM-AL that perform well in both training and test phases exhibit strong generalization, meaning they do not overfit the training data. This is evident from the similar trends in high test accuracy and low loss, matching their training performance. Adding Attention Layers and bidirectional structures significantly improves the performance metrics across all models. This improvement is most noticeable in models like GRU-AL and LSTM-AL compared to their without Attention Layer counterparts.

This research showed that using advanced deep learning architectures, such as Bidirectional GRU and Bidirectional LSTM with Attention Layers, significantly improved the detection of FDIAs in smart grids. We can further boost model performance by adjusting hyperparameters and investigating diverse aggregation strategies for ensembling models. In this study, we show the practicability of federated learning for real-world use cases where privacy and security of data are imperative. Facilitating decentralized model training,

federated learning permits the collaborative training of models in a way that safeguards the confidentiality of sensitive data. It is thus a feasible learning solution for a variety of sectors, not just smart grids.

6. Conclusions

This paper propose a federated learning framework for detecting FDIAs in CPPSs by employing a set of advanced deep learning algorithms, specifically the Bidirectional GRU and the Bidirectional LSTM incorporated with Attention Layers. Our results show that these additions not only improved the basic learning capabilities of the algorithms, but also enhanced the overall model performance. The model that attained the highest accuracy, 99.8%, was the bidirectional GRU with an Attention Layer. This model alleviates the vanishing gradient problem, making it a better candidate for retention of long-term dependencies that are present in the training data, and it makes predictions with almost no error. Combining deep learning and federated learning enhances detection precision and resolves vital issues involving data privacy and security. This study adds to the growing field of FDIA detection by developing a detection solution that is both robust and privacy-preserving—attributes that are desirable in any real-world application. Federated learning, when integrated with different paradigms of machine and deep learning, holds promise for improving a host of cybersecurity tasks—certainly this one, but also many related to detecting different kinds of attacks, such as DoS, ransomware, and others. Ensemble methods that combine the outputs of different models could make such detection systems even more powerful. To continue strengthening both the resilience and security of the systems that make up our critical infrastructure, we can look to two main approaches: exploring a broader range of deep learning architectures and applying federated learning more widely across different sectors. The continued development of scalable and efficient federated learning frameworks will be crucial for realizing the full potential of this approach in practical settings. Future research needs to be done on the resilience of federated learning models against adversarial attacks aimed at undermining model integrity. It will be crucial to develop techniques for reliably detecting these adversarial threats and mitigating them to ensure that the threat-detection systems we hope to build will be effective in practice.

Author Contributions: Conceptualization, F.K.; Methodology, F.K.; Software, S.D.; Validation, S.D. and Z.U.H.; Writing—original draft, F.K.; Writing—review & editing, S.H. and Z.U.H.; Supervision, S.H.; Funding acquisition, S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research supported by the Department of Energy Minority Serving Institution Partnership Program (EM-MSIPP) managed by the Savannah River National Laboratory under BSRA contract DE-EM0005266.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef] [PubMed]
2. CSIS_Journalism, S. Power Struggle: Anticipating Cyberattacks on the Electric Grid. CSIS Journalism Bootcamp 2024. Available online: https://journalism.csis.org/power-struggle-anticipating-cyberattacks-on-the-electric-grid/ (accessed on 5 October 2024).
3. Morehouse, C. Tensions at home and abroad pose growing threat to US grid. *E E News POLITICO* **2024**. Available online: https://www.eenews.net/articles/tensions-at-home-and-abroad-pose-growing-threat-to-us-grid/ (accessed on 5 October 2024).
4. Aljohani, T.; Almutairi, A. A comprehensive survey of cyberattacks on EVs: Research domains, attacks, defensive mechanisms, and verification methods. *Def. Technol.* 2024, *in press*. [CrossRef]
5. Liu, Y.; Ning, P.; Reiter, M.K. False data injection attacks against state estimation in electric power grids. *ACM Trans. Inf. Syst. Secur.* **2011**, *14*, 1–33. [CrossRef]
6. Gao, Y.; Ma, J.; Wang, J.; Wu, Y. Event-Triggered Adaptive Fixed-Time Secure Control for Nonlinear Cyber-Physical System With False Data-Injection Attacks. *IEEE Trans. Circuits Syst. II Express Briefs* **2023**, *70*, 316–320. [CrossRef]

7. Zhou, M.; Liu, C.; Jahromi, A.A.; Kundur, D.; Wu, J.; Long, C. Revealing Vulnerability of N-1 Secure Power Systems to Coordinated Cyber-Physical Attacks. *IEEE Trans. Power Syst.* **2023**, *38*, 1044–1057. [CrossRef]
8. Ali, M.; Sun, W. Securing Critical Infrastructures: Restoration from Cyber-Physical Attacks in Active Distribution Grids. In Proceedings of the 2024 IEEE Power & Energy Society General Meeting (PESGM), Seattle, WA, USA, 21–25 July 2024; pp. 1–5. [CrossRef]
9. Lu, K.D.; Wu, Z.G.; Huang, T. Differential Evolution-Based Three Stage Dynamic Cyber-Attack of Cyber-Physical Power Systems. *IEEE/ASME Trans. Mechatron.* **2023**, *28*, 1137–1148. [CrossRef]
10. Lu, K.D.; Wu, Z.G. An Ensemble Learning-Based Cyber-Attacks Detection Method of Cyber-Physical Power Systems. In Proceedings of the 2022 International Conference on Advanced Robotics and Mechatronics (ICARM), Guilin, China, 9–11 July 2022; pp. 1029–1034. [CrossRef]
11. Pinto, S.J.; Siano, P.; Parente, M. Review of Cybersecurity Analysis in Smart Distribution Systems and Future Directions for Using Unsupervised Learning Methods for Cyber Detection. *Energies* **2023**, *16*, 1651. [CrossRef]
12. Asefi, S.; Mitrovic, M.; Ćetenović, D.; Levi, V.; Gryazina, E.; Terzija, V. Anomaly detection and classification in power system state estimation: Combining model-based and data-driven methods. *Sustain. Energy Grids Netw.* **2023**, *35*, 101116. [CrossRef]
13. Costilla-Enriquez, N.; Weng, Y. Attack Power System State Estimation by Implicitly Learning the Underlying Models. *IEEE Trans. Smart Grid* **2023**, *14*, 649–662. [CrossRef]
14. Su, Q.; Wang, H.; Sun, C.; Li, B.; Li, J. Cyber-attacks against cyber-physical power systems security: State estimation, attacks reconstruction and defense strategy. *Appl. Math. Comput.* **2022**, *413*, 126639. [CrossRef]
15. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-efficient learning of deep networks from decentralized data. *arXiv* **2017**, arXiv:1602.05629.
16. Xie, L.; Mo, Y.; Sinopoli, B. Integrity data attacks in power market operations. *IEEE Trans. Smart Grid* **2011**, *2*, 659–666. [CrossRef]
17. Kosut, O.; Jia, L.; Thomas, R.J.; Tong, L. Malicious Data Attacks on Smart Grid State Estimation: Attack Strategies and Countermeasures. In Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, USA, 4–6 October 2010. [CrossRef]
18. Liang, G.; Zhao, J.; Luo, F.; Weller, S.R.; Dong, Z.Y. A Review of False Data Injection Attacks Against Modern Power Systems. *IEEE Trans. Smart Grid* **2017**, *8*, 1630–1638. [CrossRef]
19. Yohanandhan, R.V.; Elavarasan, R.M.; Pugazhendhi, R.; Premkumar, M.; Mihet-Popa, L.; Terzija, V. A holistic review on Cyber-Physical Power System (CPPS) testbeds for secure and sustainable electric power grid—Part—II: Classification, overview and assessment of CPPS testbeds. *Int. J. Electr. Power Energy Syst.* **2022**, *137*, 107721. [CrossRef]
20. Musleh, A.S.; Chen, G.; Dong, Z.Y. A survey on the detection algorithms for false data injection attacks in smart grids. *IEEE Trans. Smart Grid* **2019**, *11*, 2218–2234. [CrossRef]
21. Khanna, K.; Panigrahi, B.K.; Joshi, A. Feasibility and mitigation of false data injection attacks in smart grid. In Proceedings of the 2016 IEEE 6th International Conference on Power Systems (ICPS), New Delhi, India, 4–6 March 2016; pp. 1–6. [CrossRef]
22. Chang, Z.; Wu, J.; Liang, H.; Wang, Y.; Wang, Y.; Xiong, X. A review of Power System False data attack Detection Technology based on Big data. *Information* **2024**, *15*, 439. [CrossRef]
23. Inayat, U.; Zia, M.F.; Mahmood, S.; Berghout, T.; Benbouzid, M. Cybersecurity Enhancement of Smart Grid: Attacks, methods, and prospects. *Electronics* **2022**, *11*, 3854. [CrossRef]
24. Li, L.; Fan, Y.; Tse, M.; Lin, K.Y. A review of applications in federated learning. *Comput. Ind. Eng.* **2020**, *149*, 106854. [CrossRef]
25. Mammen, P.M. Federated Learning: Opportunities and Challenges. *arXiv* **2021**, arXiv:2101.05428.
26. Gosselin, R.; Vieu, L.; Loukil, F.; Benoit, A. Privacy and Security in Federated Learning: A survey. *Appl. Sci.* **2022**, *12*, 9901. [CrossRef]
27. Wen, J.; Zhang, Z.; Lan, Y.; Cui, Z.; Cai, J.; Zhang, W. A survey on federated learning: Challenges and applications. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 513–535. [CrossRef]
28. 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 Trans. Smart Grid* **2022**, *13*, 4862–4872. [CrossRef]
29. Qu, Z.; Yang, J.; Wang, Y.; Georgievitch, P.M. Detection of False Data Injection Attack in Power System Based on Hellinger Distance. *IEEE Trans. Ind. Inform.* **2024**, *20*, 2119–2128. [CrossRef]
30. Hallaji, E.; Razavi-Far, R.; Wang, M.; Saif, M.; Fardanesh, B. A Stream Learning Approach for Real-Time Identification of False Data Injection Attacks in Cyber-Physical Power Systems. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3934–3945. [CrossRef]
31. Aboelwafa, M.M.N.; Seddik, K.G.; Eldefrawy, M.H.; Gadallah, Y.; Gidlund, M. A Machine-Learning-Based Technique for False Data Injection Attacks Detection in Industrial IoT. *IEEE Internet Things J.* **2020**, *7*, 8462–8471. [CrossRef]
32. Whitaker, J.; Rawat, D.B. *Recent Advances in Cyberattack Detection and Mitigation Techniques for Renewable Photovoltaic Distributed Energy CPS*; Springer: Cham, Switzerland, 2023; pp. 1202–1215.
33. Adhikari, U.; Pan, S.; Morris, T.; Borges, R.; Beaver, J. Industrial Control System (ICS) Cyber Attack Datasets, Dataset 1: Power System Datasets. Available online: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets (accessed on 29 July 2024).

34. Mode, G.R.; Calyam, P.; Hoque, K.A. False Data Injection Attacks in Internet of Things and Deep Learning enabled Predictive Analytics. *arXiv* **2019**, arXiv:1910.01716.
35. Tahar, B.M.; Amine, S.M.; Hachana, O. Machine Learning-Based Techniques for False Data Injection Attacks Detection in Smart Grid: A review. In *Lecture Notes in Networks and Systems*; Springer: Cham, Switzerland, 2023; pp. 368–376.