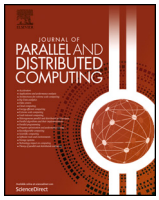
Journal of Parallel and Distributed Computing 193 (2024) 104951

Contents lists available at ScienceDirect

Journal of Parallel and Distributed Computing

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



Anomaly detection based on LSTM and autoencoders using federated learning in smart electric grid

Rakesh Shrestha [a,*], Mohammadreza Mohammadi [a], Sima Sinaei [a], Alberto Salcines [c], David Pampliega [b], Raul Clemente [b], Ana Lourdes Sanz [b], Ehsan Nowroozi [d], Anders Lindgren [a]

[a] RISE Research Institutes of Sweden, Industrial System Department, Sweden
[b] Schneider Electric, Spain
[c] TST, Cantabria, Spain
[d] Department of Business and Computing, Ravensbourne University London, United Kingdom

ARTICLE INFO

Keywords:
Autoencoders
Anomaly detection
Federated learning
Smart grid
Data privacy
And cyber-security

ABSTRACT

In smart electric grid systems, various sensors and Internet of Things (IoT) devices are used to collect electrical data at substations. In a traditional system, a multitude of energy-related data from substations needs to be migrated to central storage, such as Cloud or edge devices, for knowledge extraction that might impose severe data misuse, data manipulation, or privacy leakage. This motivates to propose anomaly detection system to detect threats and Federated Learning to resolve the issues of data silos and privacy of data. In this article, we present a framework to identify anomalies in industrial data that are gathered from the remote terminal devices deployed at the substations in the smart electric grid system. The anomaly detection system is based on Long Short-Term Memory (LSTM) and autoencoders that employs Mean Standard Deviation (MSD) and Median Absolute Deviation (MAD) approaches for detecting anomalies. We deploy Federated Learning (FL) to preserve the privacy of the data generated by the substations. FL enables energy providers to train shared AI models cooperatively without disclosing the data to the server. In order to further enhance the security and privacy properties of the proposed framework, we implemented homomorphic encryption based on the Paillier algorithm for preserving data privacy. The proposed security model performs better with MSD approach using HE-128 bit key providing 97% F1-score and 98% accuracy for K = 5 with low computation overhead as compared with HE-256 bit key.

1. Introduction

It is essential to shift from traditional electric distribution systems to a smart electrical grid for a greener society and a sustainable planet. By switching to the smart grid, information related to physical infrastructure is replaced with a digital one. The smart electric grid provides several advantages over the traditional electric distribution system [1]. However, it is not always simple to transform from a traditional physical system to a digital infrastructure as it introduces some risks and issues. We must be sensible about the risks that the new smart electrical system introduces, and we need to be prepared for the security and privacy issues they raise. These issues can be solved by implementing an anomaly detection system and machine learning technique such as Federated Learning (FL), which secure the smart grid without interruption in power sharing. The anomaly detection discovers data patterns whose characteristics are statistically different from those data samples available during the training process, which are considered normal. While FL preserves the privacy of the data generated by the substations by sharing only gradients to the central server.

An overview of a smart electric grid system and its components are shown in Fig. 1. The main components of the smart grid system are smart management, smart infrastructure, and smart protection. The smart management system provides advanced control, management services, and functionalities based on its smart system. Its main function

* Corresponding author.
E-mail addresses: rakesh.shrestha@ri.se (R. Shrestha), mohammadreza.mohammadi@ri.se (M. Mohammadi), sima.sinaei@ri.se (S. Sinaei), asalcines@tst-sistemas.es (A. Salcines), david.pampliega@se.com (D. Pampliega), raul.clemente@se.com (R. Clemente), ana.sanz@se.com (A.L. Sanz), e.nowroozi@rave.ac.uk (E. Nowroozi), anders.lindgren@ri.se (A. Lindgren).

https://doi.org/10.1016/j.jpdc.2024.104951
Received 14 September 2023; Received in revised form 27 April 2024; Accepted 24 June 2024
Available online 4 July 2024

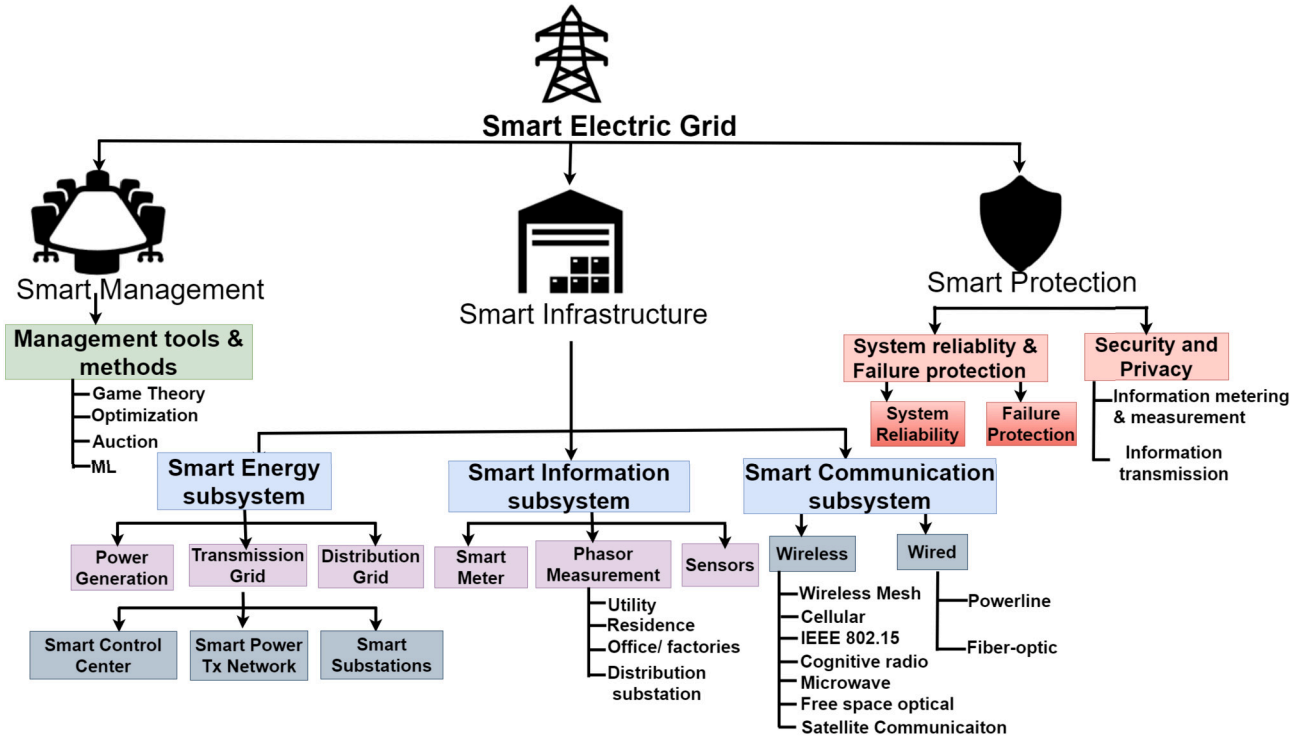


Fig. 1. Overview of smart electric grid system.

is correlated to energy productivity improvement, supply and demand equilibrium, emission control, management cost reduction, and utility growth. The smart protection system delivers sophisticated grid reliability analysis, failure protection, as well as security and privacy protection services. It also provides a smart grid electric failure protection system, addresses cybersecurity problems, and preserves the privacy of the information more effectively. Furthermore, the smart infrastructure system consists of a smart energy subsystem, a smart information subsystem, and a smart communication subsystem. The smart energy subsystem includes power generation systems such as nuclear power plant, oil Plant, coal plant, Transmission grid downgrades high-voltage electricity allowing the electricity to be safely distributed at lower voltages locally and also redirect power to other transmission lines feeding nearby areas and the distribution grid transports the generated power to end users. In addition, the transmission grid includes smart control centers, smart power transmission networks, and smart substations. While smart information subsystem consists of smart meter, phasor measurement units and other sensors that transmit information between the users and the utility providers. And the smart communication subsystem includes both wired and wireless technologies as a baseline for communication between entities [2].

In this article, we will consider the smart energy subsystem that is based on energy service providers (ESPs) or utility companies rather than energy data owners (EDOs). We address the challenges affecting smart grids by securely integrating IoT and AI devices in an electrical substation. The integration of these technologies will strengthen grid connectivity and grid monitoring. It also offers the possibility to optimize operational decisions and use of equipment, as well as protect the electrical network against undesired fluctuations in the quality of the network or equipment. We propose anomaly detection based on Long Short-Term Memory (LSTM) and autoencoder to detect anomalies in the sensor data obtained in the smart grid infrastructure. This provides intelligent anomaly detection followed by a Federated Learning (FL) approach to preserve the privacy of the electric grid data. The proposed anomaly detection system monitors and detects threats while federated learning can resolve the issues of data silos and privacy of industrial data.

The motivation of this paper comes from the requirement that the data collected by the smart sensors deployed at the substations are transferred to the central storage system such as a central server or Cloud storage for knowledge extraction. This might cause severe data misuse, data manipulation, or data privacy leakage. Moreover, due to privacy concerns, the smart grid industries are unwilling to exchange their data with others, which hinders in creating high-quality comprehensive anomaly detection model. In such a context, developing a high-quality detection model to identify anomalies in smart grids is a challenging and a difficult process.

To address these issues, distributed learning based on federated learning is implemented, where only the gradients are shared with the central server, and the data remains on the devices. Federated Learning presents a promising solution for maintaining data privacy while enabling machine learning (ML) models to be trained on decentralized devices. By training ML models on local client devices without transferring raw data to a central server, FL preserves data privacy and ensures compliance with regulations such as GDPR. FL reduces the risk of privacy breaches while still achieving accurate outcomes for anomaly detection applications. However, FL introduces new privacy concerns regarding the transmission of local model parameters between clients and servers, as this data could potentially be exploited by third parties to reconstruct sensitive information. To address this, we incorporate Homomorphic Encryption (HE) mechanisms [3] into FL. HE ensures privacy by encrypting local model parameters during communication and computation. Moreover, it is often necessary to consider a trade-off between accuracy and false alarm rate, especially when dealing with complex systems with multiple operational modes [4,5]. We run several experiments, and the results show that the proposed security model performs better with HE-128 bit key providing 97% F1-score with 98% accuracy for K = 5 in MSD approach with low computation overhead as compared with HE-256 bit key.

The main contributions in this paper are presented as follows:

- Design and implementation of anomaly detection techniques in a federated learning setup in the smart electric grid system.

- An anomaly detection model based on LSTM and autoencoders at the edge device to improve detection performance. The proposed model uses feature extraction layers to extract statistical features to detect anomalous characteristics.
- Comprehensive simulations to evaluate the performance of the proposed anomaly detection model based on FL. We run the simulation based on a synthetic industrial dataset.
- Implementation of homomorphic encryption based on the Paillier algorithm to guarantee the privacy and security of the model gradients throughout the training process in FL architecture. The results show that the suggested model performs better than without implementing homomorphic encryption.

The rest of this paper is organized as follows. In Section 2, we introduce the background and related works. In Section 3, we presented the system model of the proposed smart electrical grid system. Section 4 discusses the proposed idea and provides a detailed structure of the system. We show the experimental results in Section 5 and we conduct extensive performance evaluation in Section 6 and finally Section 7 provides conclusion and future works.

2. Background and related works

In this section, we discuss the background of anomaly detection based on the combination of artificial neural networks such as LSTM based on autoencoders, distributed machine learning techniques such as federated learning, and security mechanisms such as homomorphic encryption. These techniques will later be used in our proposed framework.

2.1. Anomaly detection

Anomaly detection is used to discover unusual behavior, rare occurrences, or outliers in large datasets that are not easily detected by standard statistical analysis. Machine learning techniques used in anomaly detection can learn patterns in the data and use them to easily detect unusual trends and patterns. In practice, several challenges make anomaly detection a difficult task [6,7]. It is not easy to specify every possible normal behavior, and therefore, detecting abnormal behavior can be difficult, especially at the boundaries. Moreover, anomaly detection in time series is one of the main challenges in today's industry, where a remarkable number of sensors are used to monitor various processes. Several research studies have been conducted to develop intelligent agents to handle the problem and solve remote monitoring challenges [8–11]. These agents often rely on algorithms that require offline training on "clean" or "labeled data", which is costly and labor-intensive.

There are several ML techniques used in anomaly detection [5] [12]. One of the popular ML techniques used in anomaly detection is LSTM. LSTM is a kind of recurrent neural network that is effective in learning order dependence in sequence prediction problems. In the smart grid system, there are currently a few solutions that can investigate smart grid network data, detect malicious behavior using an LSTM-Autoencoder(LSTM-AE) anomaly detection approach, and run the analytics on a resource-constrained device. There are couple of literatures based on LSTM-AE used in smart grids; thus, we will introduce simply the general method in this section. The authors in [13] discussed the work of the LSTM autoencoder model, where they used the Recurrent Neural Network (RNN) model to utilize LSTM units for encoding and decoding to perform unsupervised learning. The LSTM autoencoders compress large data by applying an encoder technique and then decompress it to retain the original structure by applying a decoder. The recurrent autoencoders are used because RNNs have proven to be effective for time series modeling and learning of unsupervised input data. The time-series input is encoded with a single LSTM layer and decoded with a second LSTM layer to recreate the input [4]. Pereira et al. [14] proposed a generic and unsupervised framework for anomaly detection based on variational recurrent autoencoder and a variational self-attention mechanism (VSAM) on time series data of a solar grid. Their model can detect anomalous patterns by using the probabilistic reconstruction metrics as anomaly scores. However, the model can not guarantee what a normal pattern in data looks like as the normal pattern can shift over time during different conditions. In [15], the authors proposed deep autoencoders with a sequence-to-sequence (seq2seq) structure based technique. They evaluated simple, variational, and attention-based autoencoders (AEA). The proposed solution used benign power consumption profiles with several deep autoencoders to address the lack of bad profiles. However, the malicious datasets are not easily available so they used pre-processed benign datasets only. The authors in [16] proposed a novel framework for time series anomaly detection by combining a bidirectional Long Short Term Memory (Bi-LSTM) architecture with an Autoencoder to identify anomalies in power grid data. The proposed mechanism established an optimal threshold beyond which an event can be classified as an anomaly. However, Bi-LSTM models can be computationally expensive to train, especially when dealing with large datasets and the authors have not investigated optimization techniques to mitigate this issue. Similarly, the authors in [17] proposed an unsupervised deep learning to identify anomalies in electricity consumption data an hour beforehand using a two-step approach. An LSTM regular predicted the next-hour sample, and LSTM autoencoder learned normal consumption features using the output from the LSTM regular as input. However, they have not discussed how they have used ML techniques for training and testing the data in detail and have not presented evaluation metrics such as accuracy, precision, recall, etc. The state-of-the-art of LSTM autoencoders that have been applied to anomaly detection focuses on detecting anomalies only and have not considered data misuse, and data leakage due to honest but curious servers and did not consider the privacy of the data. However, we consider mean standard deviation and median absolute deviation approaches for detecting anomalies in the smart grid system and compared them. These approaches can handle high-dimensional data and complex nonlinear relationships in the data. Moreover, we use the distributed FL technique with partial HE to protect the privacy of the data.

2.2. Federated learning

Many advanced systems rely on the gathering and processing of vast datasets to analyze, classify, and forecast future behaviors. Federated learning offers a solution by enabling collaborative distributed learning and training of local models, thereby yielding more efficient results while safeguarding privacy. There are generally three steps involved in FL training [18]: 1) Central server shares an initial model. 2) Participants train their local data with the initial model and share the local model with the central server, and 3) Central server aggregates the local models and shares the global model with participants. Recently, researchers have introduced novel approaches to integrate federated learning into various applications. For instance, in [19], the authors advocated for the use of distributed learning models employing gradient descent techniques to boost performance and scalability in applications handling massive datasets. Similarly, [20] introduced an edge-cloud hierarchical federated learning system, empowering multiple edge servers to perform partial model aggregation. Additionally, [21] proposed a federated learning-based model that utilizes a sampled subset of user equipment in the training process, predominantly executed by edge nodes. To minimize energy consumption and learning completion time, the subset of user equipment is refreshed during each iteration.

In the context of federated learning for Smart Energy Grid, fewer research activities have been proposed. For instance, in [22], a federated learning-based model is introduced for predicting energy demand in electric vehicles. This model enables charging stations to share their trained models without disclosing their raw datasets, allowing providers to process these models and predict electric vehicle energy demands,

optimizing consumption and pricing while aiming to minimize energy costs and maximize participant satisfaction. Another noteworthy model, IFed, discussed in [23], facilitates electric providers in assisting IoT users with power needs, ensuring local data privacy, and minimizing resource consumption through federated learning. Additionally, federated learning has been applied in smart grids, as detailed in [24], to reduce data volume used for training deep learning models, enabling household load forecasting without compromising privacy. Similarly, [25] presents a demand response algorithm leveraging federated learning among residential users, allowing decentralized control of household loads to schedule demand and achieve feasible power flow while safeguarding user privacy.

Federated learning methods play a critical role in maintaining the privacy of sensitive data where training data are distributed at the edge devices. FL has several distinct advantages over traditional centralized ML training [26]:

- Training time is reduced: Multiple devices are used to calculate gradients in parallel, which offers significant speedups.
- Inference time is reduced: Ultimately, each device has its own local copy of the model, so it can make predictions extremely quickly without relying on slow queries to the Cloud.
- Privacy is preserved: Uploading sensitive information to the Cloud presents a significant privacy risk for applications like smart grid system and healthcare devices. Privacy breaches in these settings may cause serious damage. As such, keeping data on local devices helps preserve end-users' privacy.
- Collaborative learning: FL is based on collaborative learning, which is easy and consumes less power as the models are trained on edge devices. The term implies that edge computing is a suitable environment for using FL. Therefore, the communication costs, privacy, security, and legalization issues could be alleviated by leveraging FL in the edge-cloud paradigm.

2.3. Homomorphic encryption

Homomorphic Encryption (HE) is an encryption technique where mathematical computations can be performed on encrypted cyphertexts and generate encrypted results without decrypting them first [27]. When the final results are decrypted, it matches the results of the operations as if they were performed on plain text [28]. The HE maintains the privacy of the clients' encrypted data while allowing third parties to carry out specific operations on the clients' encrypted data without decrypting the data. With homomorphic encryption, the client first encrypts the data before uploading it to the Cloud. Without knowing the contents of the encrypted data, the Cloud server uses HE to perform a mathematical algorithm on it. The client then receives the encrypted information from the Cloud and uses its secret key to decrypt the received encrypted data, maintaining the privacy of the data. The HE is classified into three types based on the number of mathematical operations performed on the encrypted message. They are as follows: Partially Homomorphic Encryption (PHE) [29], Somewhat Homomorphic Encryption (SHE) [30], and Fully Homomorphic Encryption (FHE) [31][32]. Within the proposed federated learning technique, the PHE scheme developed as a promising solution to ensure the confidentiality and privacy of smart industries data in the context of FL for anomaly detection. The Paillier cryptosystem allows the server to process and aggregate model parameters with the homomorphic property on the server without requiring decryption. One key advantage of the Paillier homomorphic cryptosystem is its resistance against attacks from an Honest-But-Curious (HBC) server. It has been designed to protect against possible privacy breaches by ensuring that ciphertexts do not reveal any information about the plaintexts. This is proven through its resilience against the chosen plaintext attack (CPA) based on the decisional composite residue problem. Consequently, PHE emerges as the most efficient partially homomorphic encryption scheme available for FL settings [33].

2.4. Anomaly detection in smart grid systems

Several research studies have been conducted to detect anomalies using machine-learning techniques in smart grid systems. We discuss selected works in this section. The authors in [34] utilized machine learning techniques such as deep learning (DL) methods to detect stealthy false data injection attacks on the state estimation of the power grid. The DL training is implemented offline to acquire a robust model, which is then deployed online to identify such type of attacks. The authors in [35] proposed an anomaly detection system based on autoencoder technique for smart home systems and their results showed that their anomaly detection is a strong constraint to sensor tampering and data corruption. However, the anomaly detection used is at the consumer side, not at the energy supplier side. In [36], the authors improved the resilience against unbalanced data by building new balanced representations using a deep representation learning approach. To identify cyber-attacks, the authors developed an ensemble deep learning method based on Random Forest classifiers to boost detection accuracy and reduce false positive rates. The results of their proposed ensemble stacked autoencoder outperform the commonly used classifiers. However, their scheme is not robust and energy efficient as they have not used federated learning techniques to compute the detection algorithm at the edge. Similarly, the authors in [37] proposed smArt gRid Intrusion Detection System (ARIES) for securing the communication system of the smart grid. In their ARIES architecture, the authors proposed three modules viz. (a) Data Collection Module, (b) ARIES Analysis Engine, and (c) Response Module. The data collection module sniffs the total network traffic statistics, which are analyzed by the ARIES analysis engine to detect anomalies, and finally, the response module informs the system operator regarding the anomalies. The performance evaluation of their scheme has an F1 score of 0.982 in the first detection layer, while the F1 score of the second and third layers reaches 0.751 and 0.853, respectively. In [38], the authors of ARIES enhanced their scheme by introducing an autoencoder-GAN architecture with novel minimization functions, taking into account both the adversarial error and the reconstruction difference. The enhanced proposed architecture was validated in four real smart grid evaluation environments that use the Modbus/TCP and DNP3 protocols.

In most of the above-mentioned works, the lack of labeled data furnishes machine learning techniques as an ideal solution for generating effective security applications as they can detect the applicable features autonomously. However, it is important that most of the previous works have not been validated with real smart grid environments and data. Moreover, they have not implemented federated learning set up to enhance the detection algorithm by running distributed learning techniques at the edge and lack encryption techniques such as homomorphic encryption to secure the data model.

3. System model

In this section, we introduce an anomaly detection technique based on an LSTM-AE along with a distributed machine learning framework, specifically federated learning with one central server and three clients in the smart grid, to detect anomalies in the system. We also discuss homomorphic encryption based on the Pailler algorithm as a secure privacy mechanism.

3.1. Anomaly detection approach

Our objective is to detect anomalies or outliers based on anomaly detection that usually partitions the dataset into a training set and an inference set. The training set consists of sensor data collected over a certain time period from several devices at the substations. Taking into consideration the most realistic cases and the formulation that is frequently employed in the anomaly detection system, the client device training data set consists of normal sensor data only collected from substation [39]. During inference, it detects anomalies in the test data set

that is obtained from the same sensor devices but at different time periods. We consider two commonly used anomaly detection approaches for detecting anomalies in the smart grid system and they are the Mean Standard Deviation (MSD) and Median Absolute Deviation (MAD) approaches.

3.1.1. Median Absolute Deviation (MAD)

MAD score is calculated as the median of the absolute deviations of each data point from the median of the entire dataset. The MAD score is a measure of dispersion because it is less impacted by extreme values or outliers in the data. It is especially beneficial for datasets with non-normal distributions [40]. A high MAD value implies that the data points are sparsely separated from the median, whereas a small MAD shows that the data points are closely grouped around the median. The MAD can be calculated as follows:

$$MAD_n = k.med_i|x_i - med_i.x_i| \tag{1}$$

where k is a scale factor that assumes normally distributed data, $med_i.x_i$ is the sample median or simply the middle value in the batch of points across the series. MAD is a dynamic approach and more resilient to outliers in a dataset than the standard deviation. In MAD, the deviations of a small number of outliers are irrelevant. MAD flags points as anomalous that have a large deviation from the median.

3.1.2. Mean Standard Deviation (MSD)

MSD score is a valuable indicator for assessing the effectiveness of anomaly detection systems because it is unaffected by the size of the data and allows for performance comparisons between various techniques. The MSD score is calculated by averaging the number of standard deviations between the true and estimated values. In the MSD score, the difference between the true and predicted values of the data samples is computed first. Then, the difference is divided by the standard deviation of the true values. Finally, the MSD values of the normalized differences are calculated as follows:

$$MSD = mean(|\frac{truevalue - predictedvalue}{std(truevalue)}|) \tag{2}$$

A lower MSD score represents that the algorithm performs better at identifying anomalies and is more accurate in predictions of the true values. This model can handle high-dimensional data more effectively than other methods and can capture complex nonlinear relationships in the data. In anomaly detection, a Mean Squared Error (MSE) is used as a reconstruction error, which is calculated as below,

$$MSE = \sum_{i=1}^{n}\sum_{j=1}^{m}(Y_{ij} - \hat{Y}_{ij})^2 \tag{3}$$

where n is the total number of observations, Y_{ij} is the ith true data samples, and $\hat{Y}_{ij}$ is the ith autoencoded data sample by the LSTM-AE, index i runs over the features and index j runs over along the sequence. A higher reconstruction error represents the divergence from the normal behavior. To detect such divergence, a threshold must be chosen so that if the reconstruction error value for an input data sample is above this threshold, then the input data is regarded as anomalous. In MSD, a threshold value can be selected as a decision point to determine how much a test sample deviates from the normal samples. This threshold detects anomalies if the observations exceed a predefined threshold value.

- *Threshold setting:* The threshold is set based on the reconstruction error of each input data sample that is considered as an indicator of anomaly. The reconstruction error i.e., MSE, is used for setting the threshold value. The threshold is usually set based on the mean and standard deviation of the reconstruction errors. In general, the threshold is formulated as below:

$$\tau_{MSD} = \mu + k.\sigma \tag{4}$$

where τ_{MSD} is the threshold, μ is the average reconstruction error of the normal data, σ is the standard deviation, and k is the constant that can be used to adjust the threshold. k needs to be adjusted based on the specific characteristics of the data. In our case, a value of 3 is typically used. Thus, we set $k=3$ [41]. The distribution of the reconstruction errors generated by the model for normal data samples is calculated to determine the threshold value. The data are labeled anomalous when the reconstruction error of the new input data exceeds this threshold.

$$\tau_{MSD} = Mean(MSE - Err) + (3 * Std(MSE - Err)) \tag{5}$$

It's also worth noting that threshold selection involves a trade-off between false negatives and false positives. A high threshold means fewer false positives (fewer anomalies are found). However, it also produces more false negatives (more real anomalies not being detected).

3.2. LSTM-autoencoder

LSTM-AE employs the attributes of both the LSTM neural network and the autoencoder, which builds the LSTM networks upon the autoencoder's encoding and decoding techniques. The encoder receives the high-dimensional input data stream as a fixed-size vector. The encoder strategy holds connections over numerous data points in a time-series sequence utilizing LSTM memory cells while continuously reducing the high-dimensional input vector representation into lower-dimensional interpretation until it approaches the latent space. The LSTM decoder regenerates and restores a fixed-size output sequence from the reduced representation of the input data in the latent space. Each feature vector data given to the LSTM-AE yields the corresponding output, from which the reconstruction error can be computed. The reconstruction error for normal feature data after processing by the LSTM-AE is typically less than the reconstruction error for abnormal data samples. The reconstruction error between the input and output data samples is obtained by training the LSTM-AE model with normal data samples. Once any abnormal data samples are fed into the trained model, the reconstruction error will be greater than that of normal data samples. The components' general working principle of the LSTM-AE is given in Fig. 2.

3.3. Federated learning model

The high-level federated learning network model for our proposed system is given in Fig. 3. The network model consists of the following elements:

- Energy Service Providers (ESPs): It generates electricity and distributes it to the public through a smart electric grid system. The ESP consists of current, voltage, power, and temperature parameters for efficient electricity distribution. It is located at the edge layer that consists of edge devices such as Remote Terminal Units (RTUs) that are responsible for monitoring the electric infrastructure, such as transformers, circuit breakers, and so on.
- Aggregators: The computing devices at the Fog layer provide Fog computing and communication services at the proximity of the utility providers. This layer acts as a bridge between the Cloud server and the edge nodes. This network model offers proximal model aggregations near the edge of the network, thus eliminating the data traffic load to the server. The Fog layer consists of network controllers, time-sensitive networks, edge devices, and several switches.
- Server: The Cloud layer consists of servers that provide significant computing, communication, and storage capacity. It also consists of orchestrator devices and additional servers, as well as provides other services.

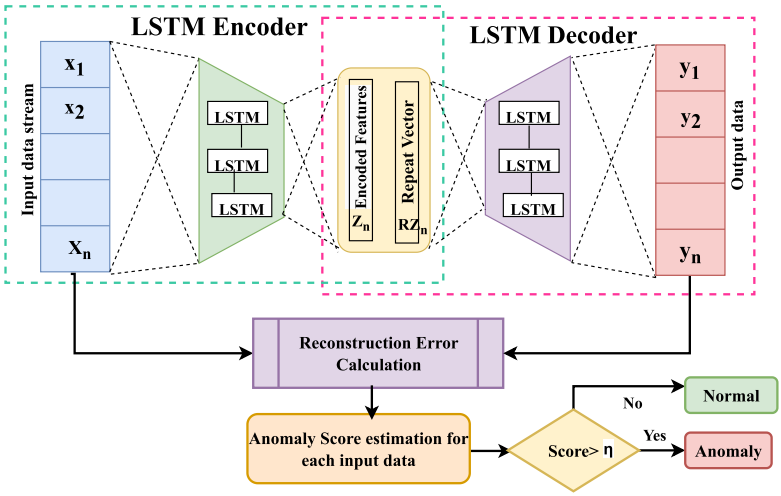


Fig. 2. General working principle of the LSTM autoencoder.

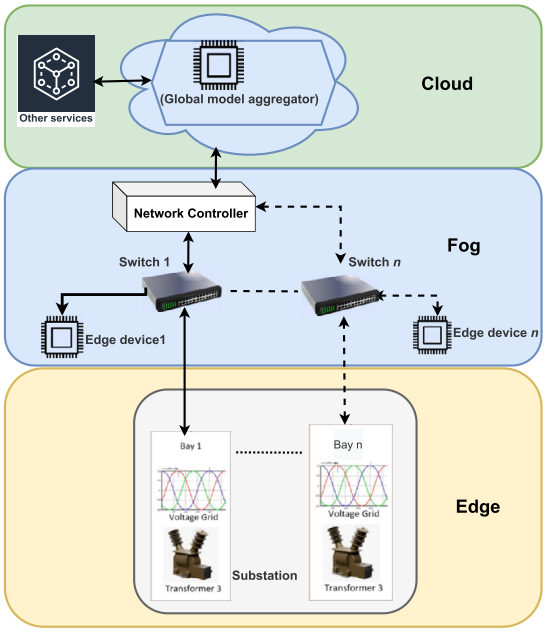


Fig. 3. High-level federated learning network model.

In this work, we use one central server in the Cloud for global data aggregation and model updates, while three FL clients are used for training the local model security [42]. The reason for using few clients is that we have limited data sets obtained from the smart grid. The overall network model will consist of a fully integrated solution ranging from sensor data collection and advanced anomaly detection based on ML models to preserve the privacy and security of the information.

3.4. Homomorphic encryption: paillier algorithm

As indicated above in Section 2.3, homomorphic encryption is beneficial for privacy preservation in a system like this. One algorithm for homomorphic encryption with many benefits, such as additive and partial multiplicative homomorphism and probabilistic encryption, is the Paillier algorithm [43], which is what will be used for homomorphic encryption in the rest of this work. The Paillier algorithm is based on the difficulty of the Decisional Composite Residuosity Assumption (DCRA), which claims that given a composite number n, evaluating whether a given residue r is a multiple of n is extremely difficult. The Paillier algorithm is discussed below [3]:

Key generation: The algorithm randomly chooses two prime numbers p and q to satisfy the greatest common divisor i.e.,

$$gcd(pq, (p-1)(q-1)) = 1 \tag{6}$$

Then, $N = p.q$ and $\lambda = lcm(p-1, q-1)$ are found, where lcm is the least common multiple. We randomly select $g \in \mathbb{Z}^*_{N^2}$ to satisfy $gcd(\breve{}(g^\lambda \bmod N^2), N) = 1$ and ensure there exists

$$\mu = (\breve{}(g^\lambda \bmod N^2))^{-1} \bmod N \tag{7}$$

where $\breve{}(x) = \frac{x-1}{N}$. The public and private keys are generated as N,g, and λ, μ, respectively.

Encryption Function ($\mathcal{E}$): Assume the plaintext message be $m \in \mathbb{Z}$ and the public key be p_k; then, the encrypting function is

$$\mathcal{E}(m, p_k) = g^m . r^N \bmod N^2 \tag{8}$$

where r is a random pad $r \in \mathbb{Z}^*_{N^2}$. Decryption Function ($\mathcal{D}$): Assume ciphertext be c and the secret key be S_k, the plaintext can be computed as follows:

$$m = \mathcal{D}(c, S_k) = \frac{\breve{}(c^\lambda \bmod N^2)}{\breve{}(g^\lambda \bmod N^2)} \bmod N = \breve{}(c^\lambda \bmod N^2) * \mu$$

Additive Homomorphic: The additive homomorphic characteristic enables the user to easily operate on the message in its ciphertext. Let the two plaintexts be m_1, m_2, and the key pair is S_k, p_k; then, we have,
$c_1 = \mathcal{E}(m_1, p_{ki}) \equiv g^{m_1} . r_1^N \bmod N^2$
$c_2 = \mathcal{E}(m_2, p_{ki}) \equiv g^{m_2} . r_2^N \bmod N^2$
thus, we have,

$$c_1 * c_2 \equiv g^{m_1+m_2} . (r_1 . r_2)^N \bmod N^2 \tag{9}$$

So, we can conclude that

$$m_1 + m_2 \bmod N = \mathcal{D}(\mathcal{E}(m_1, p_{ki}) \oplus \mathcal{E}(m_2, p_{ki}), S_{ki}) = \mathcal{D}(c_1 * c_2, S_{ki})$$

4. Proposed framework

In this section, we discuss our proposed framework and provide a detailed structure of the system. We consider Anomaly Detection based on LSTM-AE in a Federated Learning setup in smart electric grid systems, which we call ADLA-FL. The ADLA-FL framework can be used to build anomaly detection models for detecting and monitoring threats as well as the federated learning techniques that can resolve the issues of data silos and privacy of industrial data. FL allows a number of edge

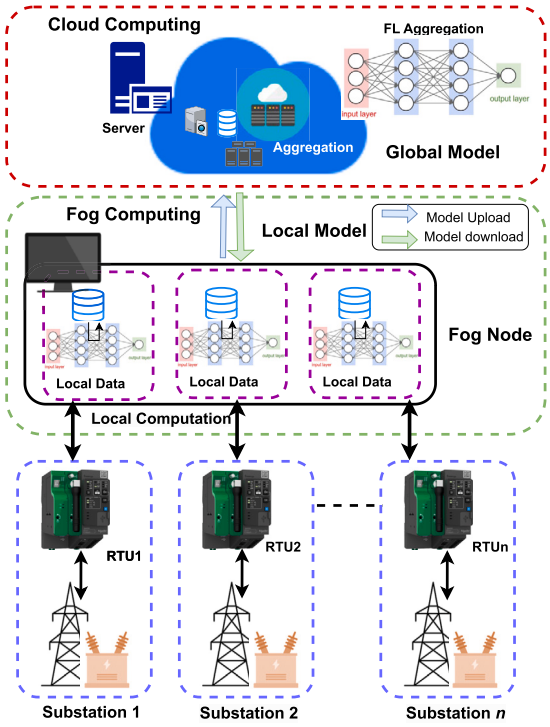


Fig. 4. Proposed anomaly detection based on FL in smart grid.

computing devices to train a model together without exchanging private information. FedAvg stands as the prevailing algorithm in federated learning, employing element-wise averaging of local model parameters with weights aligned to the sizes of client datasets. Building upon the foundations laid by FedAvg, numerous algorithms have emerged, aiming to enhance resource allocation fairness, communication efficiency, and convergence rates in federated learning. [44] Notable examples include LAG [45], Zeno [46], AFL [47], FedProx [48] and FedMA [49]. We selected FedAvg for implementation in our study due to its straightforward implementation and efficacy with homogeneous data, specifically, the independent and identically distributed (IID) data characteristic, which aligns with the context of our research. It can solve the data scarcity issue from a privacy-preserving standpoint. In our proposed FL framework, the central server is located in the Cloud, which receives gradient data from the electric substations through the intermediate Fog layers as shown in Fig. 4. The given figure is a generic figure with the architecture for larger deployment. In this case, we will use a powerful Fog node and deploy three federated learning clients at the Fog node. The Fog nodes are used to receive and compute the data from the respective RTUs and train a local ML model. Each received data sample will have an ID indicating the RTU's data it was read from. The communication protocol used to communicate between the substations and Fog node is Modbus/TCP. In this work, one RTU is located in the laboratory environment that produces synthetic data currently and will increase its number as the project progress.

At first, the central server shares the initial model with the Fog nodes located at the Fog layer. These nodes collect the local data from RTUs and preprocess to extract the training data pattern, then train their local data with the initial models and then share this local model, i.e., gradients with the central server in the Cloud. The model aggregator in the Cloud aggregates the gradients obtained from the Fog layer and updates the global model parameters. It provides a reconstruction error of observations at the final timestamp in the series. The central server then shares the updated global model with the clients. The proposed framework trains the global model until the reconstruction error of the data converges using validation data that is stored in the Cloud. The threshold selection technique uses these reconstruction errors to set an anomalous threshold for real-time detection. We use two approaches to detect the anomalies in the smart grid data, as discussed in Section 3. In the case of MSD, the mean and standard deviation of reconstruction errors are used for computing the threshold. The trained ML models can be used to directly monitor threats at the edge devices and trigger the alert in time to defend the system. The anomaly detection system at the Fog layer receives the updated training model along with the threshold from the Cloud server. At this moment, the devices can detect anomalies based on new input observations or data samples. If the reconstruction error of the new observation is above a certain anomaly threshold value, then it will be considered an anomaly else, it will be considered normal. Hence, the devices at the Fog layer can collaboratively learn an efficient global detection ML model by exchanging model parameters with a Cloud server. The data gathered by edge devices do not need to be transferred to a Cloud server for centralized processing, thus protecting electricity utility providers' privacy and lowering communication costs. However, we must also consider the privacy protection of the elements in the smart grid system. For this, we consider implementing a homomorphic encryption technique to preserve the privacy and security of the system. In this framework, we assume that the central Cloud server is honest and reliable enough to perform its designated assignment. However, it might be curious to gain knowledge of the model gradients. The key management center is expected to be an entirely honest party guaranteeing safe communication between edge devices and the Cloud server. Edge devices can be partially honest; however, they may be peculiar about the other edge devices' data. Thus, a privacy-preserving mechanism such as homomorphic encryption is required to preserve the data privacy of the smart electric grid system.

5. Experimental setup

This section describes the type of dataset used for the experimental setup and details of the experimental setup used in our proposed FL-based anomaly detection scenario. For the experiment, we use Python as it permits the use of several ML algorithms and models for anomaly detection. Some of the libraries used to process the data are Pandas, Numpy, Pytorch, and Scikit Learn [50].

5.1. Dataset

In this subsection, we analyze the proposed algorithm's performance using a synthetic industrial dataset collected from the RTU in a smart grid system. The RTU used is PowerLogic T300 from Schneider Electric, which is an embedded device that serves as an application core component for the management of low and medium-voltage public distribution networks. Obtained from an industry partner, which has been designed taking into account the generic behavior of a segment of the electric grid. We curate the dataset where the redundant data are removed to make the distribution of the dataset more balanced and reasonable. One good practice is to train the model with just normal data, and everything that differs from the normal behavior is considered as anomaly. In this way, we can detect new abnormalities or attack behavior that has never been seen before. We used a synthetic dataset based on a real-world grid system that only included the normal operation of the system. The dataset includes measures for temperature, current, and voltage for each phase, frequency, and relative humidity. In the dataset, we had 1378 data samples. There are fourteen features in the dataset that encompass critical elements of grid system operation, making them adequate for use in the training of machine learning models intended for anomaly detection. We randomly selected 125 data samples of the dataset to generate 125 synthetic anomalous samples for the test set. Then, we divided the remaining 1253 samples into two sets: Set I- 90% (1128 samples) for the training set and Set II- 10% (125 samples) for the test set. Finally, we divided those 1128 samples equally among three clients and each of them had 376 data samples for training. The test set is shaped by 125 anoma-

Table 1
Experimental setup parameters.

FL Network	1-Server and 3- clients
Evaluation Approach	MAD-score, MSD approach
Model Type	LSTM-AutoEncoder
Aggregation Method	FedAvg
Loss Function	MSE loss
Epochs	Local epochs: 4 – Global Epochs: 50

lous and 125 normal samples, i.e., 250 test data samples altogether, which will be used for evaluating the performance of our model.

5.2. Experimental setup parameters

We run the simulation on a computer with the following specifications: Intel Core i7-11800H running on Nvidia GeForce RTX 3050 graphic card with 4 GB GDDR6 and having memory of 16.0 GB RAM. We use Python 3.8 and TensorFlow 2.8.0 to build the models on the computer. The PowerLogic T300 RTU from Schneider Electric has been used as the RTU terminal. This RTU is a hardware and firmware customizable platform that serves as an application core component for the management of low-voltage and medium-voltage public distribution networks [51].

Table 1 shows the detail experimental setup parameters. In the federated learning scenario, we consider three federated learning clients and one global server at the Cloud. Each FL client has 376 samples of the training data for local model training. We use ADAM as an optimizer [52] during the training phase with a learning rate of 0.001 for neural network-based methods for a fair comparison. The local epoch was set to 4, and the global epoch was set to 50 with a batch size of 8 is used. The aggregation method used for federated learning is FedAvg. We use two different evaluation approaches for evaluating the result of the proposed framework, i.e., the MSD approach and the MAD score approach. To preserve the privacy of the smart grid system, we implement homomorphic encryption with a 128-bit key and a 256-bit key.
The simulation is carried out using one synthetic dataset available from the RTU. Thus, we consider only three clients participating in the federated learning. During the FL process, the clients use their local datasets for training a shared global model by exchanging the model parameters with the central global server. The training local data set is randomly and uniformly distributed into three clients with the same size as a local dataset to train the model. To evaluate the performance of the obtained global model, we use a testing set to test the performance.

6. Performance evaluation

We conduct extensive evaluations based on the datasets that were introduced in the previous section as a way of testing the developed algorithms to verify the effectiveness of the proposed approach. And we observe the trade-off between model accuracy and privacy empirically. Even though the datasets provided were synthetic ones, the methodology could be applied to datasets obtained from the substations in operation, as all the experimentation has considered a similar infrastructure to the one available in the substations managed by electric utilities.

6.1. Performance evaluation metrics

For performance evaluation in the proposed ADLA-FL framework, we use metrics such as Accuracy, Precision, and Recall, as F1-score similar to other machine learning research. For expressing these metrics, we need to consider some common parameters such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) for the anomaly detection process. We summarize these parameters as follows:

- True Positive (TP): It represents correctly identified anomalous samples as an anomaly among all samples,

Table 2
Confusion matrix.

		Predicted Values	
		normal	anomaly
True Values	normal	TN	FP
	anomaly	FN	TP

Table 3
Evaluation metrics and their mathematical representation.

S.No.	Evaluation Metric	Mathematical Representation
1	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
2	Precision	$\frac{TP}{TP+FP}$
3	Recall	$\frac{TP}{TP+FN}$
4	F1-Sore	$2\times\frac{Recall.Precision}{Recall+Precision}$

- True Negative (TN): It represents correctly identified benign samples as normal among all samples,
- False Positive (FP): It represents incorrectly identified benign samples as anomalous,
- False Negative (FN): It represents incorrectly identified anomalous samples as normal.

Based on the above parameters, we present the confusion matrix with four different combinations of predicted and true values that are used to determine the performance of a model. The confusion matrix is given in Table 2.

The evaluation metrics are computed using the above parameters as follows:

- Accuracy: It is the ratio of correctly identified normal samples to the total samples.
- Precision: It is the ratio of correctly identified anomalies to the total number of expected anomalies outcomes.
- Recall: It is the ratio of correctly identified anomalies to all observations in the actual assessment.
- F1-Score: It is the measure of the test's accuracy and is computed using Precision and Recall values.

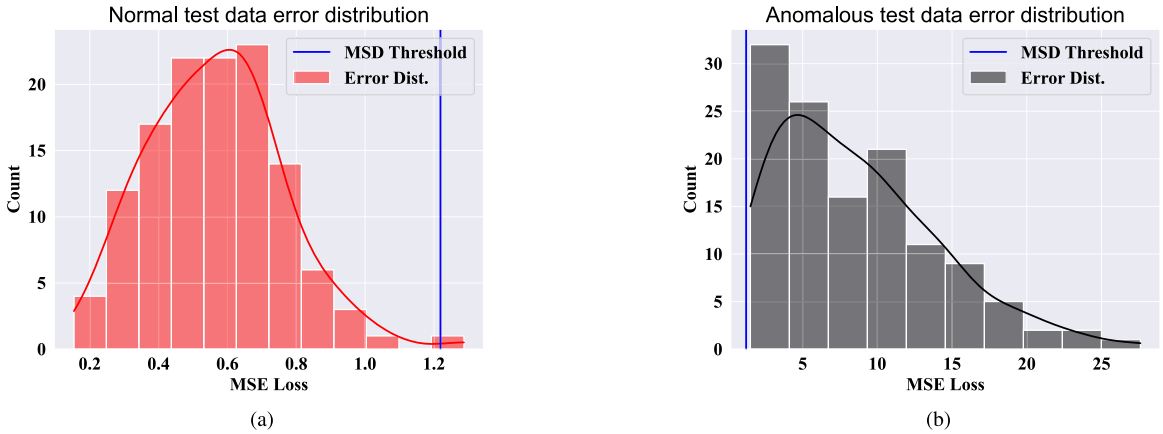
The evaluation metrics and their corresponding mathematical representation are given in Table 3.

In addition, we use the Receiver Operating Characteristic (ROC) curve as a performance evaluation metric. According to [39], the ROC curve is a visual depiction of the balance between a machine learning model's true-positive rate (TPR) and false-positive rate (FPR) across various thresholds, with TPR on the y-axis and FPR on the x-axis. The area under the ROC curve (AUC) is a statistical metric used to evaluate the efficacy of an ML model. It indicates the model's ability to correctly categorize observations into positive and negative groups. AUC-ROC is a helpful statistic for assessing model correctness since it offers a credible visual depiction of the performance of the proposed model. The AUC is computed as shown in the following equation:

$$AUC_{ROC} = \int_0^1 \frac{TP}{TP+FN} d\frac{FP}{TN+FP} \tag{10}$$

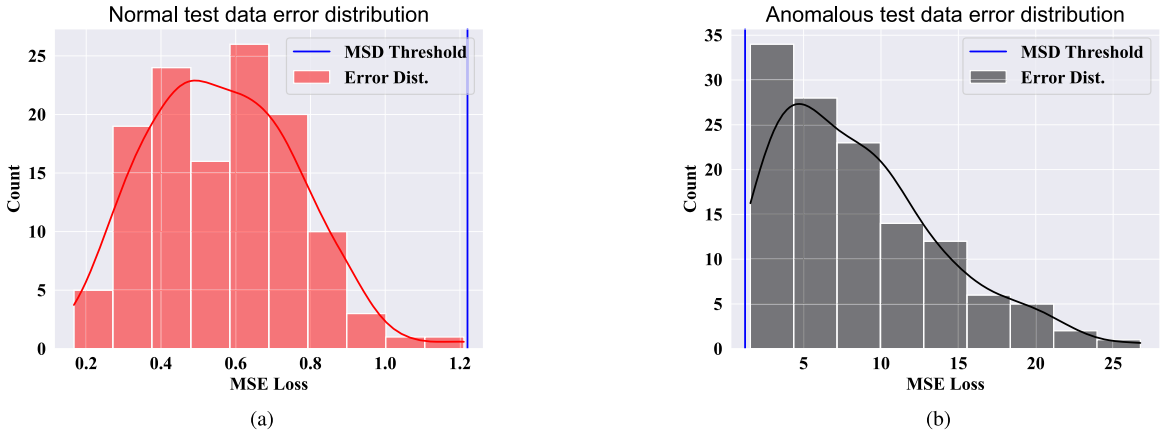
6.2. Privacy-preserving ADLA-FL with HE

In this section, we discuss the implementation of homomorphic encryption for preserving the privacy of the federated learning system. We deployed HE with a pair of key sizes 128 bits and 256 bits. For this, we use Paillier homomorphic encryption for additive cases in our simulation that was obtained from the PHE, i.e., Partially Homomorphic Encryption library in Python. The additive homomorphic encryption



(a) (b)

Fig. 5. Test data loss distribution for (a) normal and (b) anomaly data without HE.



(a) (b)

Fig. 6. Test data loss distribution for normal and anomaly data HE-128bit.

computation is much faster than multiplicative computation, which is suitable for the edge devices in the Fog layer. Each FL client encrypts the data received from the RTUs before sending it to the Cloud to perform cipher text calculation. Homomorphic encryption is applied close to the training unit, specifically between Fog Nodes and Cloud for the aggregation of encrypted data. The Cloud computes the additive operation in its ciphertext without knowing the contents of the data.

6.3. Results and discussion

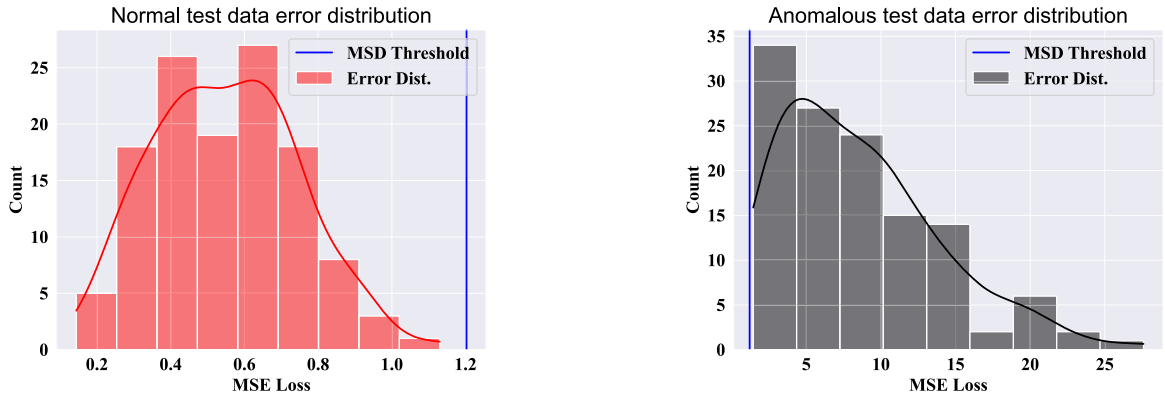
This section discusses the results obtained from our simulation based on anomaly detection using FL framework. Simulated results from the proposed framework are then analyzed to determine its efficacy. The experimental setup details show how we utilize three FL clients and a single server to test the homomorphic encryption to preserve the privacy of the FL client. We mainly used the MSD and MAD scores techniques to evaluate the performance of our proposed LSTM-AE-based architecture. We ran several simulations with (HE-128-bit key, and HE-256-bit key) and without homomorphic encryption and obtained the results. We present the distribution of normal and anomaly test data losses in terms of MSE loss for the MSD approach. Fig. 5 shows the test data loss distribution of normal and anomaly data before implementing the homomorphic encryption technique. The results show that the reconstruction error or the MSE losses are exceptionally low. In the figure, the blue line represents the threshold that is computed using Eq. (5) and is the same for both normal and anomaly results. The test data samples that are greater than the threshold value are detected as anomalies (i.e., the right side of the threshold line are anomalies).

Fig. 6 shows the test data loss distribution of normal and anomaly data after implementing the homomorphic encryption, i.e., HE-128bit key with the threshold. Similarly, Fig. 7 illustrates the loss distribution of normal and anomaly data while implementing the HE-256-bit key FL approach.

Fig. 8a shows the model performance using a confusion matrix. The confusion matrix indicates that there are 250 test data samples and among them, 125 data samples are abnormal samples. The proposed model was able to correctly identify 125 normal data samples i.e., it can correctly identify 100% of normal data samples. The model correctly identified 123 data samples out of 125 total anomalous test data samples, i.e., it can correctly identify 98.4% of normal data, i.e., TP. However, the proposed model incorrectly identified 2 data samples as normal, i.e., FN.

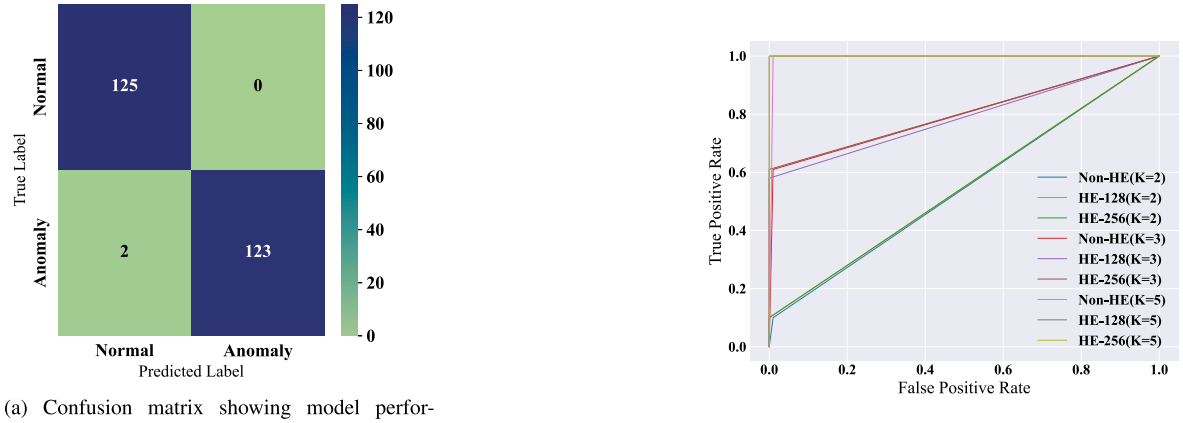
We also showed the performance of the proposed model by using the AUC-ROC curve in Fig. 8b. This curve displays the trade-off between the TPR and FPR of the proposed model. The figure displays the ROC curves for the LSTM-AE model implementing different HE encryption strategies such as non-HE, 128-bit HE, and 256-bit HE scenarios with different values of K, which are represented by corresponding colors. The AUC-ROC curve results show that the model with non-HE, HE-128, and HE-256-bit key with $K = 5$ performs better.

Our analysis included an examination of numerous approaches on our industrial dataset to determine their respective performance levels. As a result, we used One-Class SVM, Isolation Forest, and threshold-based anomaly detection methods in our conference work [53]. Our investigation found that threshold-based techniques perform better in finding abnormalities in our dataset. As a result, we have chosen to focus this study on two unique threshold-based techniques (MAD & MSD)



(a) (b)

Fig. 7. Test data loss distribution for normal and anomaly data HE-256 bit.



(a) Confusion matrix showing model performance.

(b) AUC-ROC of the model

Fig. 8. Confusion matrix and AUC-ROC curve of the model.

to compare and contrast their performance with that of our industrial dataset. Table 4 shows the detailed comparison of ADLA-FL based on the MSD approach using various performance metrics such as Recall, Precision, Accuracy, and F1-score. The table compares the proposed framework for the MSD approach with different threshold values 'K' along with HE using various encryption keys as a privacy-preserving technique. While assessing the performance of the proposed framework, the Recall performance parameter plays a crucial role. The Recall metric minimizes the number of false negative predictions. In other words, we want to make sure that all the data samples that are anomalous (TP) are correctly detected because missing an anomaly (FN) can have a significant consequence on the proposed system. The table shows that the HE provides better performance evaluation with a threshold value of K = 5. According to the results, we can verify that applying homomorphic encryption does not affect the performance of the system while providing high-level data privacy for the clients of the FL system. The only drawback of homomorphic encryption is the encryption/decryption operations computational overhead that is added to the system.

Similarly, Table 5 compares the proposed framework based on the MAD-score approach with the LSTM-AE model considering different HE strategies. From the table, it can be seen that the non-HE, HE-128-bit, and HE-256 results are similar. Furthermore, considering the results of MAD and MSD anomaly detection approaches in Tables 4 & 5, the MSD approach dominates in performance when compared to the MAD approach. For $K = 5$, i.e. anomaly samples are far from normal samples. The MSD technique performs exceptionally well, with a recall score of 98% and an F1-score of 97%. In contrast, the MAD technique produces less favorable results, with both recall and F1 scores settling around 78%, a level of performance regarded as unsatisfactory, particularly in

Table 4
Comparison of the proposed models with MSD approach with different HE and threshold values.

Approach	K	HE	Accuracy	Precision	Recall	F1-Score
MSD	K = 2	Non-HE	55%	74%	55%	44%
		HE-128	55%	75%	55%	43%
		HE-256	55%	75%	55%	43%
	K = 3	Non-HE	80%	85%	80%	79%
		HE-128	79%	85%	79%	78%
		HE-256	80%	86%	80%	79%
	K = 5	Non-HE	98%	98%	98%	97%
		HE-128	98%	97%	98%	97%
		HE-256	98%	98%	98%	97%

Table 5
Comparison of the proposed models with MAD approach with different HE and threshold values.

Approach	K	HE	Accuracy	Precision	Recall	F1-Score
MAD	K = 2	Non-HE	50%	58%	50%	34%
		HE-128	50%	57%	50%	34%
		HE-256	50%	58%	50%	35%
	K = 3	Non-HE	51%	75%	51%	35%
		HE-128	51%	76%	51%	35%
		HE-256	51%	75%	51%	34%
	K = 5	Non-HE	79%	85%	78%	78%
		HE-128	79%	85%	79%	78%
		HE-256	79%	84%	79%	78%

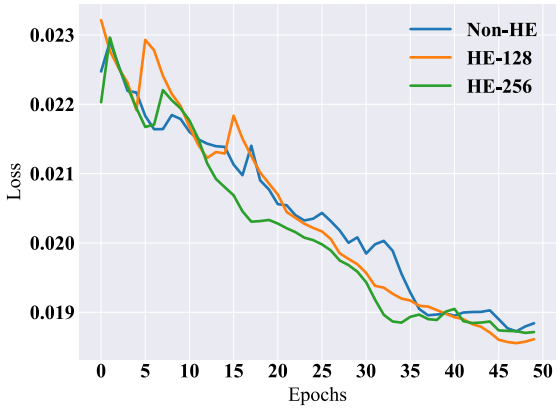


Fig. 9. Training loss performance at server side.

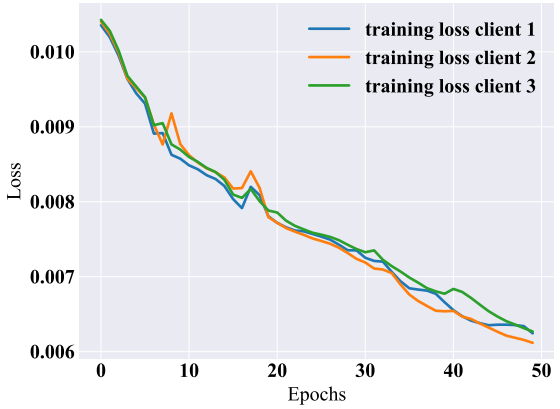


Fig. 10. Training loss performance at client side.

cases when anomaly detection is deemed uncomplicated. When $K = 3$, i.e. anomaly samples are closer to normal samples, having a satisfactory anomaly detection system is extremely important. According to the results, the MAD strategy performs sub-optimally, as evidenced by a Recall-Score of 51% and an F1-score of 35%. In comparison, the MSD strategy performs substantially better, with a Recall-score of 80% and an F1-score of 79%, respectively. For $K = 2$, i.e. anomaly samples are similar to normal samples, the inferior performance of both the MSD and MAD techniques is expected considering the intrinsic challenge of identifying such abnormalities. It's possible that a threshold-based anomaly detection strategy won't work well for finding this kind of abnormality.

Fig. 9 evaluates the training loss performance on the server side. The server training loss takes some iteration to converge the loss. Fig. 10 shows the training loss performance on the clients' side. The training loss performance on the client side and server side has the same behavior. The training loss of each of the three clients converges at around 50 epochs with a learning rate of 0.001 showing better performance of the proposed framework. While the homomorphic encryption had no negative impact on model performance, and HE models performed comparably to non-HE models. The negligible differences in the results are due to the fact that the Paillier homomorphic encryption system employs a probabilistic encryption approach that introduces a random integer into the plaintext prior to encryption, known as a "blinding factor". This guarantees that the same plaintext value creates a different ciphertext each time it is encrypted, preventing attackers from deducing plaintext information from ciphertext patterns. It should be noted that as the key size grows, the execution time of FL model training increases. Moreover, choosing the right key size for the Paillier encryption technique requires balancing security and computational efficiency, which varies depending on the use case. Larger key sizes provide several advantages,

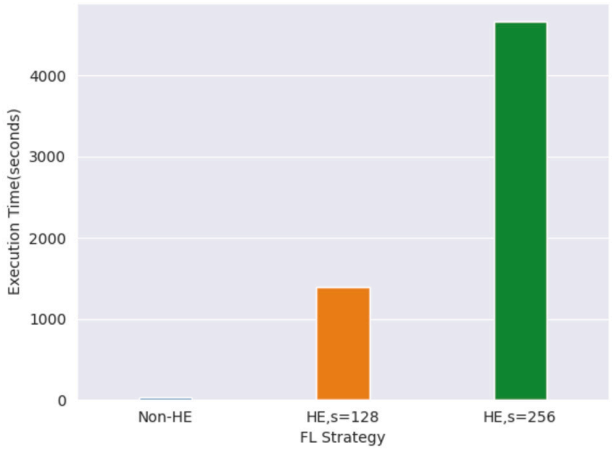


Fig. 11. Execution time taken by LSTM-AE with varying HE encryption keys in the proposed framework.

including better security against brute-force attacks, as it is computationally unfeasible for attackers to attempt all possible keys until they locate the proper one.

We analyzed the computation cost of implementing HE in the proposed framework using LSTM-AE. We compare the results using 128-bit keys, 256-bit keys, and without homomorphic encryption in the proposed framework. The result of computation time taken by the LSTM-AE methods with varying HE encryption keys is given in Fig. 11. The execution time taken to perform LSTM-AE without implementing HE is 29.61 s. The execution time taken to perform LSTM-AE with implementing HE-128bit is 1387 s, and HE-256bit is 4662 s. In practice, there is a trade-off between performance and computation time. A higher key length provides a superior level of protection against various attacks. In addition, as computational power increases, greater key sizes may make the Paillier encryption system more resistant to attacks, guaranteeing that it continues to give strong privacy guarantees. On the other hand, using a higher key length incurs a computational overhead, causing slower encryption and decryption times, as well as higher memory requirements. In general, key length should be chosen depending on a trade-off between security and performance needs. A higher key length, such as 256 bits, may be more appropriate if a high degree of privacy and security is desired. If performance is required, a shorter key length, such as 128 bits, may suffice.

7. Conclusion and future works

This paper introduces ADLA-FL, an anomaly detection based on the LSTM-AE framework implementing FL in the smart electric grid. The ADLA-FL framework employed MSD and MAD approaches for detecting anomalies that can efficiently detect outliers in industrial data. We ran extensive simulations and examined the performance metrics of the test data samples, and experimental evaluations highlighted the effectiveness of ADLA. Our outcomes indicate that the MSD approach performs better than the MAD approach on the smart electric grid system data by having an f1-score of 80% and 98% for $K = 3$ and $K = 5$, respectively. Furthermore, our solution maintains accuracy and performance close to the baseline method (without encryption) for HE cases and they ensure high privacy and security for the clients in the FL system. As HE cases add more computational overhead to the FL system, the execution time increases drastically when using HE in our framework. The results show that there is a trade-off between privacy, performance, and computation time while using homomorphic encryption. If we tend to preserve the privacy of the framework using the HE, then we need to compromise execution time and performance. Future research will focus on exploring the potential of multi-key homomorphic encryption in Federated Learning for sensitive applications like Smart Electric Grid

and implement minimum infrastructure security measures to guarantee the effective privacy of data. This includes preventing privacy leakage and collusion between devices and the server, further enhancing the system's security and reliability.

CRediT authorship contribution statement

Rakesh Shrestha: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing. **Mohammadreza Mohammadi:** Data curation, Formal analysis, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Sima Sinaei:** Conceptualization, Formal analysis, Funding acquisition, Supervision, Validation, Writing – original draft, Writing – review & editing. **Alberto Salcines:** Data curation, Formal analysis, Investigation, Resources, Supervision, Validation. **David Pampliega:** Data curation, Formal analysis, Resources, Validation. **Raul Clemente:** Data curation, Formal analysis, Investigation, Software, Validation. **Ana Lourdes Sanz:** Data curation, Formal analysis, Project administration, Resources. **Ehsan Nowroozi:** Formal analysis. **Anders Lindgren:** Formal analysis.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Dr. Sima Sinaei reports financial support was provided by European Union. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the EU ECSEL project DAIS which has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No. 101007273. The work reflects only the authors' views; the European Commission is not responsible for any use that may be made of the information it contains.

References

[1] C. Lamnatou, D. Chemisana, C. Cristofari, Smart grids and smart technologies in relation to photovoltaics, storage systems, buildings and the environment, Renew. Energy 185 (2022) 1376–1391.
[2] X. Fang, S. Misra, G. Xue, D. Yang, Smart grid - the new and improved power grid: a survey, IEEE Commun. Surv. Tutor. 14 (2012) 944–980.
[3] P. Li, Y. Zhao, L. Chen, K. Cheng, C. Xie, X. Wang, Q. Hu, Uncertainty Measured Active Client Selection for Federated Learning in Smart Grid, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 148–153.
[4] W. Wei, L. Liu, Gradient leakage attack resilient deep learning, IEEE Trans. Inf. Forensics Secur. 17 (2022) 303–316.
[5] Y. Cui, Z. Liu, S. Lian, A survey on unsupervised industrial anomaly detection algorithms, arXiv preprint arXiv:2204.11161, 2022.
[6] P. Nunes, J. Santos, E. Rocha, Challenges in predictive maintenance–a review, CIRP J. Manuf. Sci. Technol. 40 (2023) 53–67.
[7] T. Mazhar, H.M. Irfan, I. Haq, I. Ullah, M. Ashraf, T.A. Shloul, Y.Y. Ghadi, D.H. Elkamchouchi, Analysis of challenges and solutions of iot in smart grids using ai and machine learning techniques: a review, Electronics 12 (2023) 242.
[8] F. Zhou, G. Wen, Y. Ma, H. Geng, R. Huang, L. Pei, W. Yu, L. Chu, R. Qiu, A comprehensive survey for deep-learning-based abnormality detection in smart grids with multimodal image data, Appl. Sci. 12 (2022) 5336.
[9] I. Ortega-Fernandez, F. Liberati, A review of denial of service attack and mitigation in the smart grid using reinforcement learning, Energies 16 (2023) 635.
[10] M. Abdelkhalek, G. Ravikumar, M. Govindarasu, Ml-based anomaly detection system for der communication in smart grid, New Orleans, LA, USA, 2022, pp. 1–5.
[11] A. Dairi, F. Harrou, B. Bouyeddou, S.-M. Senouci, Y. Sun, Semi-Supervised Deep Learning-Driven Anomaly Detection Schemes for Cyber-Attack Detection in Smart Grids, Springer, 2023.
[12] R. Shrestha, A. Omidkar, S.A. Roudi, R. Abbas, S. Kim, Machine-learning-enabled intrusion detection system for cellular connected uav networks, Electronics 10 (13) (2021) 1549.
[13] N. Srivastava, E. Mansimov, R. Salakhutdinov, Unsupervised learning of video representations using lstms, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, vol. ICML'15.JMLR.org, 2015, pp. 843–852.
[14] J. Pereira, M. Silveira, Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Dec 2018, pp. 1275–1282.
[15] A. Takiddin, M. Ismail, U. Zafar, E. Serpedin, Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids, IEEE Syst. J. 16 (3) (2022) 4106–4117.
[16] D. Guha, R. Chatterjee, B. Sikdar, Anomaly detection using lstm-based variational autoencoder in unsupervised data in power grid, IEEE Syst. J. 17 (3) (2023) 4313–4323.
[17] M. Kardi, T. AlSkaif, B. Tekinerdogan, J.P.S. Catalão, Anomaly detection in electricity consumption data using deep learning, in: 2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), 2021, pp. 1–6.
[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: A. Singh, J. Zhu (Eds.), vol. 54, PMLR, 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html.
[19] S. Wang, T. Tuor, T. Salonidis, K.K. Leung, C. Makaya, T. He, K. Chan, Adaptive federated learning in resource constrained edge computing systems, IEEE J. Sel. Areas Commun. 37 (6) (2019) 1205–1221.
[20] L. Liu, J. Zhang, S. Song, K.B. Letaief, Client-edge-cloud hierarchical federated learning, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 2020, pp. 1–6.
[21] V.-D. Nguyen, S.K. Sharma, T.X. Vu, S. Chatzinotas, B. Ottersten, Efficient federated learning algorithm for resource allocation in wireless iot networks, IEEE Int. Things J. 8 (5) (2020) 3394–3409.
[22] Y.M. Saputra, D.T. Hoang, D.N. Nguyen, E. Dutkiewicz, M.D. Mueck, S. Srikanteswara, Energy demand prediction with federated learning for electric vehicle networks, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.
[23] H. Cao, S. Liu, R. Zhao, X. Xiong, Ifed: a novel federated learning framework for local differential privacy in power Internet of things, Int. J. Distrib. Sens. Netw. 16 (5) (2020) 1550147720919698.
[24] A. Taïk, S. Cherkaoui, Electrical load forecasting using edge computing and federated learning, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 2020, pp. 1–6.
[25] S. Bahrami, Y.C. Chen, V.W. Wong, Deep reinforcement learning for demand response in distribution networks, IEEE Trans. Smart Grid 12 (2) (2020) 1496–1506.
[26] J. Liu, J. Huang, Y. Zhou, X. Li, S. Ji, H. Xiong, D. Dou, From distributed machine learning to federated learning: a survey, Knowl. Inf. Syst. 64 (4) (2022) 885–917.
[27] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, Y. Liu, {BatchCrypt}: efficient homomorphic encryption for {Cross-Silo} federated learning, in: 2020 USENIX Annual Technical Conference (USENIX ATC 20), 2020, pp. 493–506.
[28] J. Tang, Y. Cui, Q. Li, K. Ren, J. Liu, R. Buyya, Ensuring security and privacy preservation for cloud data services, ACM Comput. Surv. 49 (1) (2016) 1–39.
[29] C. Jost, H. Lam, A. Maximov, B. Smeets, Encryption performance improvements of the paillier cryptosystem, Cryptol. ePrint Arch. (2015).
[30] J. Fan, F. Vercauteren, Somewhat practical fully homomorphic encryption, Cryptol. ePrint Arch. (2012).
[31] C. Gentry, S. Halevi, Implementing gentry's fully-homomorphic encryption scheme, in: Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2011, pp. 129–148.
[32] R. Shrestha, S. Kim, Integration of iot with blockchain and homomorphic encryption: challenging issues and opportunities, Adv. Comput. 115 (2019).
[33] J. Zhang, B. Chen, S. Yu, H. Deng, Pefl: a privacy-enhanced federated learning scheme for big data analytics, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.
[34] M. Ashrafuzzaman, Y. Chakhchoukh, A.A. Jillepalli, P.T. Tosic, D.C. de Leon, F.T. Sheldon, B.K. Johnson, Detecting stealthy false data injection attacks in power grids using deep learning, in: 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2018, pp. 219–225.
[35] T. Cultice, D. Ionel, H. Thapliyal, Smart home sensor anomaly detection using convolutional autoencoder neural network, in: 2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), IEEE, 2020, pp. 67–70.
[36] A. Al-Abassi, J. Sakhnini, H. Karimipour, Unsupervised stacked autoencoders for anomaly detection on smart cyber-physical grids, in: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2020, pp. 3123–3129.
[37] P. Radoglou Grammatikis, P. Sarigiannidis, G. Efstathopoulos, E. Panaousis, Aries: a novel multivariate intrusion detection system for smart grid, Sensors 20 (18) (2020) 5305.
[38] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras, P. Sarigiannidis, A unified deep learning anomaly detection and classification approach for smart grid environments, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021) 1137–1151.
[39] R. Shrestha, R. Bajracharya, S. Kim, 6g enabled unmanned aerial vehicle traffic management: a perspective, IEEE Access 9 (2021) 119–91 136.

[40] A. Dotis-Georgiou, Anomaly detection with median absolute deviation, pp. 1–12, 7. [Online]. Available: https://www.influxdata.com/blog/anomaly-detection-with-median-absolute-deviation/. (Accessed 26 April 2024), 2020.
[41] J. Clark, Z. Liu, N. Japkowicz, Adaptive threshold for outlier detection on data streams, Inst. Electr. Electron. Eng. Inc. 1 (2019) 41–49.
[42] K. Zhang, Y. Jiang, L. Seversky, C. Xu, D. Liu, H. Song, Federated variational learning for anomaly detection in multivariate time series, in: 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC), IEEE Computer Society, Los Alamitos, CA, USA, oct 2021, pp. 1–9. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/IPCCC51483.2021.9679367.
[43] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: J. Stern (Ed.), Advances in Cryptology — EUROCRYPT'99, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 223–238.
[44] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, F. Piccialli, Model aggregation techniques in federated learning: a comprehensive survey, Future Gener. Comput. Syst. (ISSN 0167-739X) 150 (2024) 272–293, https://doi.org/10.1016/j.future.2023.09.008.
[45] T. Chen, G. Giannakis, T. Sun, W. Yin, Lag: lazily aggregated gradient for communication-efficient distributed learning, Adv. Neural Inf. Process. Syst. 31 (2018).
[46] C. Xie, S. Koyejo, I. Gupta, Zeno: distributed stochastic gradient descent with suspicion-based fault-tolerance, in: International Conference on Machine Learning. PMLR, 2019, pp. 6893–6901.
[47] M. Mohri, G. Sivek, A.T. Suresh, Agnostic federated learning, in: International Conference on Machine Learning. PMLR, 2019, pp. 4615–4625.
[48] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, in: Proceedings of Machine Learning and Systems, vol. 2, 2020, pp. 429–450.
[49] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, Y. Khazaeni, Federated learning with matched averaging, arXiv preprint arXiv:2002.06440, 2020.
[50] S. learn 1.2.1, Sklearn.preprocessing: maxabsscaler. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MaxAbsScaler.html. (Accessed 26 April 2024), 2023.
[51] Powerlogic t300: A powerful remote terminal unit (rtu) for grid automation, Schneider Electr. 1 (2023).
[52] Z. Zhang, Improved Adam optimizer for deep neural networks, in: 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), 2018, pp. 1–2.
[53] M. Mohammadi, R. Shrestha, S. Sinaei, A. Salcines, D. Pampliega, R. Clemente, A.L. Sanz, Anomaly detection using lstm-autoencoder in smart grid: a federated learning approach, in: Proceedings of the 2023 7th International Conference on Cloud and Big Data Computing, Ser. ICCBDC'23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 48–54. [Online]. Available: https://doi.org/10.1145/3616131.3616138.