Integrating Federated Learning and Differential Privacy for Secure Anomaly Detectionin Smart Grids

Mohammadreza Mohammadi
Digital System, RISE Research Institutes of Sweden
Sweden
mohammadreza.mohammadi@ri.se

Rakesh Shrestha
Digital System, RISE Research Institutes of Sweden
Sweden
rakesh.shrestha@ri.se

Sima Sinaei*
Digital System, RISE Research Institutes of Sweden
Sweden
sima.sinaei@ri.se

Abstract

Anomaly detection is essential for ensuring the safe and efficient operation of industrial systems like smart grids. Smart grid stations handle sensitive data and are often hesitant to share it with third-party servers for centralized anomaly detection. Federated Learning (FL) offers a viable solution to this issue by enhancing anomaly detection in smart grids without compromising data privacy. We present a method for developing an unsupervised anomaly detection system using FL applied to a synthetic dataset that mimics a real-world smart grid system's behavior. We focus on utilizing FL's long short-term memory autoencoder in short, LSTM-AE for anomaly detection. However, there are concerns about potential privacy breaches in the FL system. Hence, to address this issue, we propose to integrate differential privacy (DP) with FL for anomaly detection by adding artificial noise to parameters at the client side before aggregation. This method ensures data privacy while maintaining the convergence of federated learning algorithms. Moreover, this research determines the optimal privacy level to balance noise scale and model accuracy. Our findings suggest a criterion for selecting the right privacy budget of DP based on the requirement of the system to provide good level of privacy in the system while maintaining the f1-score of FL-based anomaly detection system greater than 90%.

CCS Concepts

• **Computing methodologies** → Artificial intelligence; Distributed artificial intelligence.

Keywords

Anomaly Detection, Federated Learning, Smart Grid, Federated Learning, Differential Privacy

ACM Reference Format:

Mohammadreza Mohammadi, Rakesh Shrestha, and Sima Sinaei. 2024. Integrating Federated Learning and Differential Privacy for Secure Anomaly Detectionin Smart Grids. In *2024 8th International Conference on Cloud and Big Data Computing (ICCBDC 2024), August 15–17, 2024, Oxford, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3694860.3694869

*Corresponding author.

ICCBDC 2024, August 15–17, 2024, Oxford, United Kingdom

ACM ISBN 979-8-4007-1725-3/24/08
https://doi.org/10.1145/3694860.3694869

1 Introduction

A smart grid is an advanced electric grid that leverages communication technologies for electricity generation, transmission, and distribution in an efficient, sustainable, and reliable manner. Smart grid system generates a huge amount of data from various sources, such as sensors, meters, and other devices, all of which require real-time analysis. While these systems offer numerous benefits, they also face significant technological challenges [1]. Anomaly detection is essential in industrial systems like smart grids to ensure their safe and efficient operation [2]. In the context of smart grids, anomaly detection involves identifying activity patterns that deviate from the expected performance [3]. These deviations may indicate malfunctions, cyber-attacks, or other irregularities that can lead to system instability, power surge, or even blackouts [4]. It is better to detect these types of anomalies to prevent the grid system or mitigate the impact of such events. Smart grid stations are typically located in various locations, raising privacy concerns about sharing data with a third-party server for centralized anomaly detection systems. Federated Learning offers a viable solution to these issues, enhancing anomaly detection in smart grid systems while safeguarding user privacy [3][5]. Some of the advantages of FL are as follows:

(i) FL allows model training on distributed data without transmitting sensitive information to a centralized server, thereby protecting user privacy and data security [6].
(ii) It enables better model generalization by utilizing diverse and large-scale decentralized data sources [7].
(iii) It reduces communication and computation costs by performing local updates on user devices and aggregating only the necessary model updates [8].
(iv) FL supports continuous learning on edge devices, which is particularly beneficial in scenarios with limited network access or a need for real-time data processing [9].

However, FL faces several challenges, with privacy being a major concern. By connecting users in a collaborative model and increasing the number of training iterations, the FL setup becomes vulnerable to new privacy attacks [10]. The FL protocol involves exchanging gradients and parameters between the server and clients, which could allow adversaries to learn details about the neural network model, including its parameters, structure, and output. These attacks are categorized into three groups: model inversion, property inference, and membership inference attacks [11].

There are two types of privacy-preserving strategies in FL. The first method operates at the data level, ensuring data privacy during the transmission of clients' parameters throughout FL training and limiting attackers' access to clients' datasets. Examples of data-level privacy-preserving approaches include Secure Multiparty Computation (SMC) [12] and Homomorphic Encryption (HE). The second type, known as content-level techniques, involves modifying training samples or models to prevent adversaries from accessing clients' datasets. Global Differential Privacy (GDP) [13] and Local Differential Privacy (LDP) are two effective content-level privacy-preserving approaches. To ensure the privacy of sensitive data and the convergence of FL algorithms, we propose a solution in this research that uses local differential privacy with Stochastic Gradient Descent (SGD). We evaluate our approach by varying the level of noise added to local gradient updates and measuring model accuracy and loss. The rest of this paper is organized as follows. In Section 1, we introduce preliminary concepts related to federated learning, and differential privacy, while Section 2 covers previous work on Anomaly Detection in Smart Grid Systems. The system design is described in Section 3. Section 4 describes the experimental setup and results obtained by using the proposed approach. In this section, we analyze the relationship between model accuracy and the noise scale. Finally, Section 5 provides the conclusion of the paper.

2 BACKGROUND AND RELATED WORK

2.1 Federated Learning

Federated Learning (FL) is a new machine learning paradigm built on the principles of distributed data and training. It allows learning techniques to be applied directly on edge devices or on-device, rather than relying on centralized data processing [13]. The concept of FL originated with Google's Gboard, which learns new words and phrases without sending the data to Google's cloud server [14]. Since then, numerous advancements in FL have aimed to enhance privacy and prevent data leakage by decoupling data from model training. Notable FL frameworks, such as those presented in [15][16][17] are based on stochastic gradient descent (SGD), a widely used optimization method for models like neural networks and logistic regression.

In particular, [18] introduced the FedAvg algorithm, which uses iterative model averaging to distribute model parameters equally across local models. Additionally, [19] proposed the FedSup framework for cloud-edge-client drowsiness detection, which efficiently detects uncertainty in images, optimizes computing resources, reduces communication rounds, minimizes uploaded data, and improves central model accuracy by integrating locally trained models. Despite these advancements, FL still faces privacy challenges, as FL algorithms can be targeted by hostile attackers. Addressing these issues remains a critical focus for the ongoing development and implementation of FL systems.

2.2 Differential Privacy

FL cannot fully ensure the privacy of sensitive data, as it might still be leaked or retrieved by the adversaries. Privacy guarantees in FL models are typically provided using either data-level approaches (e.g., homomorphic encryption, secure multi-party computation, blockchain) or content-level approaches (e.g., local and global differential privacy) [20][21]. This work focuses on content-level differential privacy due to the high computational costs and communication overheads of cryptographic techniques. Differential privacy is a strong mathematical framework that assures individual data points in a dataset cannot be easily deduced from the aggregated output. DP protects individual records by adding random noise to data or model parameters, making it difficult for nodes to determine whether a specific data point was included in the learning process [22]. Privacy is measured using parameters such as privacy budget and sensitivity; smaller values indicate better privacy preservation. [23] introduced the DPFedAvgGAN framework, employing differential privacy to defend against GAN-based attacks in FL environments. Ghazi et al. [24] improved FL model privacy by combining shuffling with differential privacy and using an "invisible cloak" algorithm, though this introduced uncertainty into model parameters, potentially impairing training. [25] highlighted the advantages of differential privacy and demonstrated its use in multi-agent systems, reinforcement learning, transfer learning, and distributed ML, although with limited empirical results.

2.3 Anomaly Detection in Smart Grid Systems

Several studies have explored using machine-learning techniques for anomaly detection in smart grids. For instance, [26] utilized deep learning (DL) methods to detect stealthy false data injection attacks in power grids. The DL model was trained offline and deployed online to identify attacks. [27] proposed an autoencoder-based anomaly detection system for smart homes, demonstrating strong resistance to sensor tampering and data corruption, though it was limited to the consumer side. [28] improved resilience against unbalanced data using deep representation learning and an ensemble deep learning method based on Random Forest classifiers, achieving better detection accuracy but lacking energy efficiency and federated learning techniques. [29] introduced the ARIES system for smart grid communication security, featuring modules for data collection, analysis, and response. Their system achieved high F1 scores across multiple detection layers. Enhancements to ARIES included an autoencoder-GAN architecture validated in real smart grid environments [30]. Most of these works highlight the potential of machine learning for security in smart grids due to its ability to autonomously detect relevant features. However, many studies lack validation with real-world data and do not incorporate FL or encryption techniques like homomorphic encryption for data security.

3 SYSTEM DESIGN

The proposed integrated federated learning and differential privacy for secure anomaly detection framework is given in Figure 1. The framework can be used to build secure anomaly detection models for detecting and monitoring threats and resolve the issues of data silos and privacy of industrial data by integrating FL and DP.

3.1 System Model

We employ a model known as LSTM-AE for the system model. It has one LSTM layer as an encoder, a 64-dimensional latent space, and another LSTM layer as a decoder. The loss function was Mean

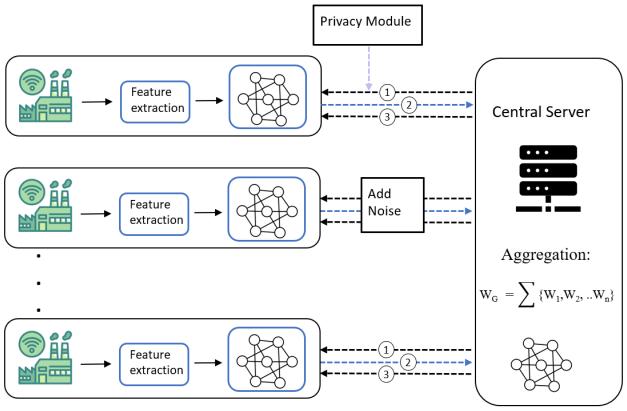

Figure 1: Proposed integrated FL and DP framework in smart grid

Square Error (MSE), and Adam was employed as the optimizer. The server's aggregation mechanism was FedAVG [8, 28].

3.2 Integrated Federated Learning Using Differential Privacy for Anomaly Detection (IFL-DP)

Differential privacy in FL is achieved by first calibrating noise on the model gradients before sharing them with the central server. Even if the model gradients are intercepted or evaluated, this noise addition keeps the central server, or any prospective adversary, from discovering sensitive information about individual data points. The clipping threshold (C) is one of the most important variables when integrating FL with DP. By limiting the number of gradients that each device can contribute, the clipping threshold reduces the sensitivity of the model gradients. In particular, the gradient norm is scaled down to this maximum value if it surpasses the threshold. This stage is critical because it standardizes the contributions from different devices, ensuring that no single device has a disproportionate influence on the aggregated model, which could otherwise lead to privacy leaks. Stronger privacy assurances are offered by a lower C, but the more considerable distortion of the gradients may lead to models that are less accurate. On the other hand, finer gradients are possible with a higher C, but the privacy assurances are compromised. A critical variable in DP is the privacy budget (ϵ) that measures the total amount of privacy lost over several learning cycles. A smaller value of ϵ denotes a stricter privacy guarantee, implying that there is more considerable noise added to the gradients, providing higher privacy protection but may result in a performance drop of the anomaly detection model. Nevertheless, there is a trade-off between the privacy budget and the usefulness of the model. We implement DP in our work by first introducing noise to the model gradients before sending them to the central server for aggregation. The following steps are involved in this process: (a)Calculating Gradients: Using its local data, each device calculates the gradients. (b) Gradient Clipping: To regulate the sensitivity of the gradients, they are clipped to a predetermined threshold. (c) Noise Addition: To ensure differential privacy, calibrated noise, usually derived from a Gaussian or Laplace distribution, is added to the clipped gradients. (d) Gradient Aggregation: To update the global model, the central server receives the noisy gradients and aggregates them. These steps ensure that the contributions from each device remain private, even in the presence of adversarial attacks or untrusted central servers.

3.3 Anomaly Detection Approach

We use threshold-based anomaly detection approach in this study for detecting anomalies [29]. In the FL setup, each client specifies a threshold based on the reconstruction errors of its training data using equation 1, where T is the threshold, M is the mean, S is the standard deviation, and E is the client's training set reconstruction errors. Then, each test set data's reconstruction errors will be computed and compared with T; if the error exceeds the threshold, it is regarded as abnormal otherwise normal. The authors of [29] claim that coefficient 3 is a best practice number that is frequently utilized in a variety of works.

$$T = M\ (E) + (3 * S\ (E))\ T = M\ (E) + (3 * S\ (E)) \quad (1)$$

4 EXPERIMENTAL SETUP and Evaluation

In this section, we discuss the dataset, simulation setup, evaluation and results for the proposed IFL-DP based anomaly detection system.

4.1 Dataset

For our analysis, we employed a synthetic industrial dataset collected from the sensors in the smart grid system. The information was supplied by a Remote Terminal Unit (RTU), an embedded device that functions as a central application component for managing public distribution networks with low and medium voltages. In particular, the experimentation used Schneider Electric's Easergy T300 RTU to produce synthetic data [27]. Data was collected hourly for nearly two months, and each record in the collection contains 14 features. It is important to note that the dataset only includes normal data because abnormal data could not have been generated due to the sensitivity of the grid system sensors. As a result, 1241 samples of normal data make up the dataset. The dataset provides helpful insights on the behavior of the grid system sensors under typical operating conditions, even in the absence of unusual data. The fourteen features in the dataset encompass essential aspects of grid system functioning, making them adequate for utilizing in the training of deep learning models intended for anomaly identification. Since we lacked any pre-existing anomalous data in our dataset, we decided to create synthetic anomalous data to evaluate the effectiveness of our trained FL models. We looked at the connections between the 14 attributes in the dataset before creating artificial anomalies as shown in Figure 2. We found that 0.93 connection between the "current-rms-t" and "current-rms-s" features, even though no significant correlations were found between the features. As a result, we choose to produce anomalous data by altering just one feature. It is an important condition because if our system can detect one outlier with only one abnormal feature, it will be able to detect other outliers with multiple inappropriate features.

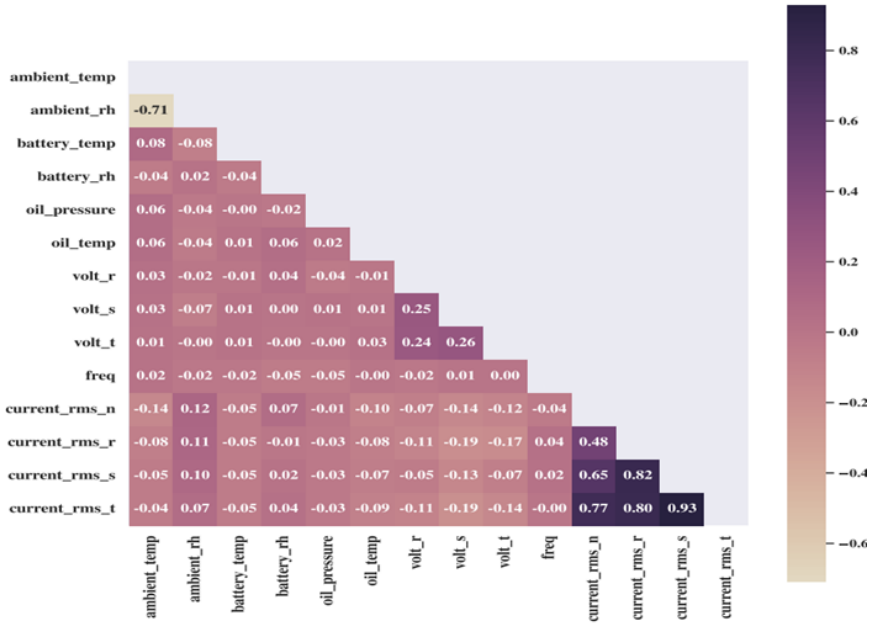

Figure 2: Correlation of features in dataset.

To create our anomalous dataset, we first randomly choose one feature from a list of fourteen features. Next, we calculate the feature's mean and standard deviation to learn more about the feature's data distribution. We then attempt to construct some synthetic data with the same mean but a larger standard deviation than the distribution of that feature in the original dataset. Lastly, we pick a synthetic data point that exceeds a predetermined threshold from the created data. There are various ways to set this threshold. The higher the threshold, the more likely the selected data is to be selected as an anomaly due to its greater divergence from normal data. We considered three different thresholds in this work.

The formula for computing the threshold is as follows:

$$Threshold = M + (K * S) \tag{2}$$

where M is mean and S is the standard deviation of the selected feature in the original dataset, which contains only normal data. The coefficient K indicates the degree to which the anomaly can be strictly selected. In this paper, we consider three distinct thresholds supplied by varying K levels.

4.2 Data Preprocessing

This section outlines the procedures to be followed to prepare an FL anomaly detection model for training. We choose 90% of the data at random for training and 10% for testing so that we only use the normal data in our training dataset. As a result, we have 1241 training data that are split equally among the clients. Next, we normalize every client's data, keeping in mind that we only have access to their local data. Rescaling input data features to have the same scale and range is known as normalization. Normalization is used to lessen the impact of feature scale differences on training, increasing its effectiveness and efficiency. We scaled the data using the standard scaler methodology, which yields a mean of zero and a standard deviation of one. We standardized the test data for each client individually, taking into account artificial abnormalities. It indicates that we believe the client side has completed the testing process. Thus, the client model rescales the data features first and then attempts to identify the new input data as normal or anomalous when new data are available from the sensors for testing.

4.3 Simulation Setup

Regarding the FL setting, we have three clients and one server. Each client has its own ADAM optimizerwith a learning rate of 0.001 and their loss function is MSE. We had 50 rounds of training and the local epochs for each client set to 5. We used a synthetic dataset based on real-world grid system that only included the normal operation of the system. In the dataset, we had 1378 data samples in the dataset. We selected 10% of them to generate 137 synthetic anomalous samples for the test set. Then, we divided the remaining 1241 samples into two sets: (a) 90% (1116 samples) for the training set and (b) 10% (125 samples) for the test set. Finally, we divided those 1116 samples equally among clients and each of them had 372 data samples for training. The test set is shaped by 137 anomalous and 125 normal samples, which will be used for evaluating the performance of our model.

4.4 Evaluation

In this section, we will present the results of our anomaly detection technique using different models. We used three different model groups, namely: Threshold-based Centralized LSTM-AE (CEN-AE-THR), Threshold-based FL LSTM-AE (FL-AE-THR), and Threshold-based Integrated FL with Differential Privacy LSTM-AE

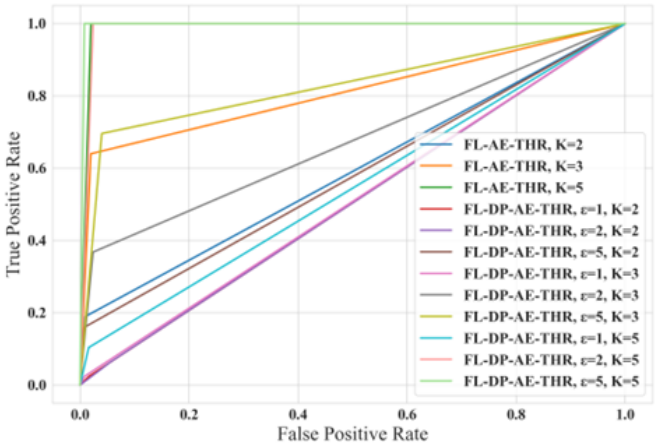

Figure 3: AUC-ROC of the model

(IFL-DP-AE-THR). All the models aim to detect anomalies by comparing the reconstruction error of the data in the test set with a threshold that is computed separately for each client based on their local reconstruction errors. As a baseline, we implemented an equivalent centralized model to the FL models, named Centralized Threshold-based LSTM-AE (CEN-AE-THR). For our DP setting, we selected different $\epsilon \in \{0.2, 0.5, 1.0, 1.5, 2.0, 5.0\}$ and C=2.0. To evaluate our anomaly detection models, we used several metrics such as Accuracy, Precision, Recall, and F1-Score. The F1-score is the harmonic mean of precision and recall and provides a balanced evaluation of both measures. In anomaly detection cases, a high F1-score ensures that the model can discover the majority of actual anomalies, while reducing the number of false positives. Another statistic we used to evaluate our models is the ROC curve.

The ROC curve shows how effectively a classifier can discriminate between normal and abnormal examples by plotting the True Positive Rates (TPR) versus the False Positive Rates (FPR). The Area Under the ROC Curve (ROC-AUC) is a scalar statistic that quantifies the classifier's overall performance over all potential operational points. A higher ROC-AUC value suggests better performance, which is represented in Figure 3.

4.5 Results

We compared the proposed models in centralized and federated setup for different synthetic anomalies and the results are presented in Table 1. It is important to find a right set of parameters for the FL-DP model in a way that maintain the good performance of it while adding noise to the parameters to have suitable level of privacy in the FL system. So, we have examined different privacy-budgets (ϵ) and reported the results in Table 1. Our goal is to have smaller ϵ, as it provides more privacy for the FL framework and have performance close to the FL case that has no DP, i.e. FL-AE-THR. According to the results, for K=5, FL-DP-AE-THR model with ϵ =1.5, 2.0, 5.0 have close performance to our baselines with f1-scores greater than 90%.

As the anomaly points become closer to normal points (e.g., K=3), all FL models show a significant decrease in performance. The FL-AE-THR model exhibits around 20% drop in performance, while the FL-DP-AE-THR model with ϵ = 1.5 and ϵ =2.0 experience a 45% and 35% reduction compared to the K=5 case. For ϵ =5.0, the drop in f1-score follows the same pattern as FL-AE-THR model. Therefore, we can expect to have ideal FL-DP-AE-THR model for 2.0<= ϵ<=5.0 where we have acceptable performance and privacy level. For K=2, where the abnormal points are very close to the normal data, all FL-based models' f1-Scores drop considerably to the range of 34-48%. While, the CEN-AE-THR model has extremely better performance for K=2 with 85% f1-score. These results suggest that threshold-based anomaly detection is the best solution when a centralized model can be trained by accessing the entire dataset. For FL models, the threshold-based model is a good choice, as it can detect abnormal data with K=3, K=5 efficiently. However, when there is a high probability of having anomaly points close to the normal data (e.g., K=2), it is better to use other approaches, as suggested in our previous work [31]. For ϵ<1.5, the FL models completely failed to detect anomalies for all K values while they have good level of privacy. These results were verified after multiple runs of the learning system.

In Figure 3, we present ROC curves plot of different FL models. The closer the ROC curve is to the upper left corner of the plot, the better the performance of the anomaly detection model. For K=5, FL-AE-THR and FL-DP-AE-THR (ϵ =5 and ϵ =2) work slightly better and achieves high TPR while maintaining low FPR. For K=3, FL-AE-THR and FL-DP-AE-THR (ϵ =5) have drop in performance on detecting anomalies while they can detect most of normal data correctly. For K=2, both FL-AE-THR and FL-DP-AE-THR (all epsilons) models are not performing well, i.e. these models detect most of anomalies data as normal data wrongly. Overall, we know that there is a trade-off between level of privacy and performance of model. The amount of noise added to model updates is controlled by ϵ parameter, the smaller ϵ provide more privacy in the FL system while it can increase the amount of noise and as a result, decrease the performance of anomaly detection system. Our goal is to find a model that has a good level of privacy and good performance, concurrently. Considering the mentioned conditions, for a ideal

Table 1: Comparison of the proposed models for different synthetic anomalies

Model	ϵ	K	Accuracy	Precision	Recall	F1-Score
CEN-AE-THR	-	K=2	85%	85%	85%	85%
	-	K=3	90%	92%	90%	90%
	-	K=5	91%	93%	90%	91%
FL-AE-THR	-	K=2	54%	71%	56%	45%
	-	K=3	80%	84%	81%	80%
	-	K=5	99%	99%	99%	99%
IFL-DP-AE-THR	0.2	K=2	50%	50%	50%	37%
	0.2	K=3	50%	53%	50%	36%
	0.2	K=5	48%	34%	48%	33%
IFL-DP-AE-THR	0.5	K=2	49%	46%	49%	35%
	0.5	K=3	49%	43%	49%	34%
	0.5	K=5	49%	41%	49%	34%
IFL-DP-AE-THR	1.0	K=2	48%	38%	48%	34%
	1.0	K=3	48%	31%	48%	33%
	1.0	K=5	55%	73%	55%	45%
IFL-DP-AE-THR	1.5	K=2	52%	64%	52%	40%
	1.5	K=3	55%	66%	55%	45%
	1.5	K=5	91%	92%	91%	91%
IFL-DP-AE-THR	2.0	K=2	50%	51%	50%	38%
	2.0	K=3	67%	77%	67%	63%
	2.0	K=5	98%	98%	98%	98%
IFL-DP-AE-THR	5.0	K=2	57%	74%	57%	48%
	5.0	K=3	82%	85%	82%	82%
	5.0	K=5	99%	99%	99%	99%

private FL system, system administration should select ϵ based on requirement of the system. If it is more important to have high level of privacy and it is possible to compromise performance the ϵ need be selected closer to 2. If the requirement of the system is opposite and performance of the system is more preferred than privacy of the system, the ϵ can be selected from the values closer to 5.

5 CONCLUSION

In this paper, we introduced an approach for anomaly detection in smart grid systems using federated learning (FL) integrated with differential privacy (DP). This method enhances privacy preservation while ensuring the effective convergence of FL algorithms. By combining a long short-term memory autoencoder (LSTM-AE) within the FL framework and adding artificial noise to client-side parameters, we safeguard individual data points against privacy breaches. Our evaluation, based on a synthetic dataset reflecting real-world grid behavior, analyzed the trade-off between accuracy and privacy. Despite the introduction of noise affecting accuracy, the balance achieved maintains robust performance, i.e. f1-score larger than 90%, and strong privacy protection. The method preserves data privacy and supports the integrity and convergence of the FL process. In conclusion, integrating FL with DP presents a promising solution for secure and efficient anomaly detection in smart grids. Future research will focus on optimizing the balance between privacy and accuracy while exploring advanced noise addition techniques.

Acknowledgments

This work was supported by EU ECSEL project DAIS that has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No.101007273.

References

[1] A. Bari, J. Jiang, W. Saad and A. Jaekel, "Challenges in the Smart Grid Applications: An Overview," *International Journal of Distributed Sensor Networks,* vol. 10, p. 974682, 2014.
[2] D. Ramotsoela, A. Abu-Mahfouz and G. Hancke, "A Survey of Anomaly Detection in Industrial Wireless Sensor Networks with Critical Water System Infrastructure as a Case Study," *Sensors,* vol. 18, p. 2491, August 2018.
[3] J. Jithish, B. Alangot, N. Mahalingam and K. S. Yeo, "Distributed Anomaly Detection in Smart Grids: A Federated Learning-Based Approach," *IEEE Access,* vol. 11, pp. 7157-7179, 2023.
[4] B. K. Hammerschmitt, A. d. Rosa Abaide, F. C. Lucchese, C. C. Martins, A. S. da Silveira, J. Rigodanzo, J. V. Maccari Brabo Castro and J. A. Dall Agnol Rohr, "Non-Technical Losses Review and Possible Methodology Solutions," in *2020 6th International Conference on Electric Power and Energy Conversion Systems (EPECS)*, 2020.
[5] B. McMahan and D. Ramage, Federated Learning: Collaborative Machine Learning without Centralized Training Data.
[6] Q. Yang, Y. Liu, T. Chen and Y. Tong, "Federated Machine Learning: Concept and Applications," arXiv e-prints, p. arXiv:1902.04885, February 2019.
[7] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh and D. Bacon, Federated Learning: Strategies for Improving Communication Efficiency, 2017.
[8] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 2017.
[9] K. Katevas, E. Bagdasaryan, J. Waterman, M. M. Safadieh, E. Birrell, H. Haddadi and D. Estrin, Policy-Based Federated Learning, 2021.
[10] M. Nasr, R. Shokri & A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and

federated learning", In 2019 IEEE symposium on security and privacy (SP) (pp. 739-753).
[11] Jere, Malhar S., Tyler Farnan, and Farinaz Koushanfar. "A taxonomy of attacks on federated learning." IEEE Security & Privacy 19.2 (2020): 20-28.
[12] Wang, F., Zhu, H., Lu, R., Zheng, Y., & Li, H. (2021). A privacy-preserving and non-interactive federated learning scheme for regression training with gradient descent. Information Sciences, 552, 183-200.
[13] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.
[14] McMahan, H. B., Yu, F. X., Richtarik, P., Suresh, A. T., & Bacon, D. (2016, December). Federated learning: Strategies for improving communication efficiency. In Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain (pp. 5-10).
[15] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems, 2, 429-450.
[16] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.
[17] Wang, S., Tuor, T., Salonidis, T., Leung, K. K., Makaya, C., He, T., & Chan, K. (2019). Adaptive federated learning in resource constrained edge computing systems. IEEE journal on selected areas in communications, 37(6), 1205-1221.
[18] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017, April). Communication-efficient learning of deep networks from decentralized data. In Artificial intelligence and statistics (pp. 1273-1282). PMLR.
[19] Zhao, C., Gao, Z., Wang, Q., Xiao, K., Mo, Z., & Deen, M. J. (2023). FedSup: A communication-efficient federated learning fatigue driving behaviors supervision approach. Future Generation Computer Systems, 138, 52-60.
[20] Cui, L., Qu, Y., Xie, G., Zeng, D., Li, R., Shen, S., & Yu, S. (2021). Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures. IEEE Transactions on Industrial Informatics, 18(5), 3492-3500.
[21] Xiong, Z., Cai, Z., Takabi, D., & Li, W. (2021). Privacy threat and defense for federated learning with non-iid data in AIoT. IEEE Transactions on Industrial Informatics, 18(2), 1310-1321.
[22] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016, October). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 308-318).
[23] Augenstein, S., McMahan, H. B., Ramage, D., Ramaswamy, S., Kairouz, P., Chen, M., & Mathews, R. (2019). Generative models for effective ML on private, decentralized datasets. arXiv preprint arXiv:1911.06679.
[24] Ghazi, B., Pagh, R., & Velingker, A. (2019). Scalable and differentially private distributed aggregation in the shuffled model. arXiv preprint arXiv:1906.08320.
[25] Zhu, T., & Philip, S. Y. (2019, July). Applying differential privacy mechanism in artificial intelligence. In 2019 IEEE 39th international conference on distributed computing systems (ICDCS) (pp. 1601-1609). IEEE
[26] Ashrafuzzaman, M., Chakhchoukh, Y., Jillepalli, A. A., Tosic, P. T., de Leon, D. C., Sheldon, F. T., & Johnson, B. K. (2018, June). Detecting stealthy false data injection attacks in power grids using deep learning. In 14th International Wireless Communications & Mobile Computing Conference (IWCMC) (pp. 219-225).
[27] Cultice, T., Ionel, D., & Thapliyal, H. (2020, December). Smart home sensor anomaly detection using convolutional autoencoder neural network. In 2020 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS) (pp. 67-70).
[28] Al-Abassi, A., Sakhnini, J., & Karimipour, H. (2020, October). Unsupervised stacked autoencoders for anomaly detection on smart cyber-physical grids. In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3123-3129). IEEE.
[29] Radoglou Grammatikis, P., Sarigiannidis, P., Efstathopoulos, G., & Panaousis, E. (2020). ARIES: A novel multivariate intrusion detection system for smart grid. Sensors, 20(18), 5305
[30] Siniosoglou, I., Radoglou-Grammatikis, P., Efstathopoulos, G., Fouliras, P., & Sarigiannidis, P. (2021). A unified deep learning anomaly detection and classification approach for smart grid environments. IEEE Transactions on Network and Service Management, 18(2), 1137-1151.
[31] M. Mohammadi, R. Shrestha, S. Sinaei, A. Salcines, D. Pampliega, R. Clemente, and 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. New York, NY, USA: Association for Computing Machinery, 2023, p. 48–54