AJS

Algerian Journal of Sciences
Volume 2 – Issue 1 – PP. 25 – 36

FedKNN Aggregator for Fast Convergence and Low Overhead in Robust FL-Based IDS to Secure Smart Farming

Rafika Saadouni[1*], Amina Khacha[1], Chirihane Gherbi[1], Yasmine Harbi[1], Zibouda Aliouat[1]
[1]LRSD Laboratory, Ferhat Abbas University of Setif 1, Setif, Algeria.

Abstract

The increasing reliance on IoT devices in smart farming systems has heightened the need for robust cybersecurity measures to protect against cyber threats. This paper presents an innovative Intrusion Detection System (IDS) framework designed to enhance the security of smart farming devices through Federated Learning (FL). This approach preserves privacy and enables distributed training across clients without sharing data. Our approach reduces communication overhead by achieving model convergence with only a few rounds. To accomplish this goal, we propose a novel aggregator algorithm called Federated K-Nearest Neighbors (Fed- KNN), which leverages the K-Nearest Neighbors (KNN) machine learning model. Additionally, we use a hybrid CNN-GRU model to construct our IDS, and we use a metaheuristic Genetic Algorithm (GA) to improve it. The framework optimizes nine key hyperparameters server-side and distributes them to clients for local training. Experimental results on the Edge-IIoT dataset demonstrate the effectiveness of the proposed method, achieving an accuracy of 99.97%, an F1-score of 99.90%, and a low false positive rate (FPR) 2, 84×10^{-5}% on multi-attack detection with just 4 rounds.

Keywords: Aggregation, Smart Farming, IDS, GA, KNN, CNN, GRU, Edge-IIoT, Federated Learning (FL).

Article History

Received: 13-11-2024
Accepted: 19-02-2025
Published: 30-09-2025

* Corresponding author
Email (rafika.saadouni@univ-setif.dz)

1 Introduction

According to the United Nations (UN), the global population is expected to grow to 9 billion by 2050, an increase of approximately one-third from current levels [1]. This growing population will require substantial quantities of food, and traditional agricultural and farming techniques will not be sufficient to meet the increased demand. As a result, the world is likely to adopt intelligent systems based on the Internet of Things (IoT) to enhance both the quality and quantity of food production. Smart Farming represents the integration of IoT technologies within the agriculture sector, aiming to digitally and electronically monitor various aspects such as crop health, environmental factors, soil quality, fertilization, and irrigation. By utilizing smart networking and mobility [2], Smart Farming enhances the adaptability of agricultural operations, ensuring seamless interoperability and better connectivity with both customers and suppliers. Furthermore, it encourages the adoption of innovative business models, thereby boosting the efficiency and productivity of farming practices [3, 4].

To meet these demands, a significant amount of data must circulate within the network between devices and users. This extensive data flow increases the network's vulnerability to attacks and malicious data, which can disrupt the functioning of smart farming systems. Such attacks can lead to inaccurate monitoring, faulty decision-making, and compromised crop management, ultimately affecting productivity and causing significant financial losses and food supply disruptions. Integrating intelligent intrusion detection systems (IDS) is a key solution for addressing security challenges, and systems based on federated learning (FL) have emerged as a promising approach in recent research [5]. These systems utilize distributed data training, enabling collaborative model development that enhances performance while preserving the privacy of users and devices [6]. This method enables the development of strong models without requiring the exchange of sensitive data [8, 9].

Even though this approach is effective, it faces significant challenges. The large volume of communication data generated by numerous training rounds and random parameter selection can lead to network bottlenecks, increased latency, higher energy consumption, and an elevated risk of data interception, tampering, or leakage of sensitive information.

Our paper focuses on achieving model convergence with fewer rounds by introducing a novel aggregator based on machine learning, departing from traditional mathematical strategies. We also incorporate bio-inspired techniques for hyperparameter optimization (HPO) to select the most effective parameters, thereby accelerating the training process [7]. Our main contributions are:

- Build an FL-based IDS using our previously proposed hybrid model (CNN-GRU).
- Integrate a fine-tuning approach with Genetic Algorithm (GA) [11] into the server of the FL system.
- Propose a new aggregator method based on the KNN algorithm called FedKNN to achieve model convergence in fewer rounds.
- Assess the model on multiclass classification and evaluate it using a new real dataset, Edge-IIoT.

The remainder of this article is structured as follows: Section 2 reviews related work. Section 3 introduces the proposed IDS. Section 4 details the experimental setup of the study, while Section 5 presents the results of our experimental analysis and comparisons. Finally, Section 6 concludes the work and outlines future directions.

2 Related Work

In this section, we review recent studies that propose building robust Intrusion Detection Systems (IDSs) using federated learning (FL) within IoT environments. These studies are relevant to our research and are summarized and compared to our work in Table 1.

Olanrewaju-George et al. [12] combined a supervised Deep Neural Network (DNN) model and an unsupervised AutoEncoder (AE) model to build an IDS using a FL mechanism to secure IoT infrastructures. The AE was employed to detect deviations in network traffic from normal to abnormal, while the DNN was used to classify the abnormal traffic into specific types. They integrated a hyperparameter optimizer based on random search within their framework to select the best model parameters. The architecture was evaluated using the N-BaIoT dataset, demonstrating an accuracy of 90.39% and an F1-score of 93.12%.However, the authors did not specify the number of rounds used to train their model, and they only evaluated their work on binary classification.

To protect wireless sensor networks (WSN), Bukhari et al. [13] implemented an IDS within the network, utilizing a federated learning approach to maintain privacy. They developed the IDS by combining a Stacked Convolutional Neural Network (SCNN) with Bidirectional Long Short-Term Memory (Bi-LSTM), incorporating Principal Component Analysis (PCA) for feature extraction. The effectiveness of the system was evaluated using the WSN-DS and CIC-IDS-2017 datasets, where the FL-IDS achieved accuracies of 99.7% and 99.93%, respectively.

Jia et al. [14] introduced a new aggregation method to enhance the security of Artificial Intelligence of Things (AIoT) devices, known as the Fed-Dynamic Gravitational Search Algorithm (Fed-DGSA). This approach optimizes the weight updating process in FL local models. When applied to the NSL-KDD dataset for both binary and multi-class classification, it achieved an accuracy of 97.8%.

To maintain security and privacy within IoT-based vehicular networks, Bhavsar et al. [15] developed an FL-based IDS incorporating both Logistic Regression (LR) and CNN. To further enhance detection accuracy, they integrated the Pearson Correlation Coefficient (PCC) for feature extraction. Their work was evaluated using two datasets: the older NSL-KDD dataset and the more relevant Car-Hacking dataset. The CNN model achieved an accuracy of 97.08% on the NSL-KDD dataset and 99.92% on the Car-Hacking dataset.

Umair et al. [16] developed a collaborative IDS using FL to enhance intrusion detection while pre- serving data privacy. Their approach employed Dynamic Weighted Aggregation Federated Learning (DWAFL), which adjusts model aggregation weights based on individual client performance. The ExtraTreesClassifier was used for feature selection, and data was evenly distributed across clients using an Identically Distributed (IID) technique. Experiments on the N-BaIoT dataset achieved 92.2% accuracy with 5 clients, 94.2% with 10 clients, and 93.2% with 15 clients.

In another work [17], they presented an IDS leveraging dynamic FL aggregation to enhance the security of IoT devices. This work employed a Random Forest (RF) for feature selection, and a CNN served as the classifier. Their approach achieved an accuracy of 87.98% on the CSE-CIC-IDS2018 dataset in multi-class classification.

While the reviewed studies provide valuable insights and contribute significantly to the field, they also have limitations that our work aims to address. Some studies utilize outdated datasets and evaluate models only for binary classification. Only one study employed parameter fine-tuning to enhance model performance. Many of the studies do not specify the number of training rounds used, and those that do often test the model over a large number of rounds, resulting in substantial communication overhead. Additionally, most studies overlook the importance of the aggregation method, relying on default aggregators like FedAvg.

3 Intrusion Detection Approach

3.1 IDS Deployment for Enhanced Security in Smart Farming

The network architecture designed to enhance smart farming security is illustrated in Figure 1. The architecture consists of smart farming devices organized into groups. Each group is connected to an edge server equipped with an Intrusion Detection System (IDS) for real-time threat monitoring and initial data processing. These edge servers also manage long-term data storage and connect to network resources, such as routers, which facilitate data transmission and connectivity. The network resources layer ensures smooth interaction between the edge servers and the cloud data centers. Finally, the cloud data centers, situated at the top layer, provide robust storage, advanced processing capabilities, and communication with users, thereby completing the network's comprehensive data management and security framework.

Table 1: Overview of existing FL-based IDS studies

Paper	Year	Network	Model	Dataset	Aggregator	Rounds	fine tuning method
[16]	2023	IoT	CNN-LSTM	N-BaIoT	DWAFLL	5, 10, 15	-
[12]	2024	IoT	DNN-AE	N-BaIoT	FedAvgM	-	Random Search
[14]	2024	AIoT	CNN	NSL-KDD	Fed-DGSA	-	-
[13]	2024	WSN	SCNN-Bi-LSTM	WSN-DS, CIC-IDS-2017	FedAvg	-	-
[17]	2024	IoT	CNN	CSE-CIC-IDS2018	Dynamic-FL	-	-
[15]	2024	Vehicular net-works	LR, CNN	NSL-KDD ,Car-Hacking	FedAvg, FedYogi	-	-
Our Work	2024	IoT	CNN-GRU	EdgeIIoT	FedKNN	4	GA

In this work, we propose integrating an IDS into edge servers to safeguard smart farming devices from network-wide attacks. The IDS will monitor traffic between devices and the cloud, detecting anomalies at the device level and blocking malicious traffic from reaching the cloud. We assume that the edge servers, which will act as FL servers, are equipped with robust security measures. Additionally, we assume that the devices are protected against adversarial attacks and that model weights transmitted during training are encrypted to maintain privacy.

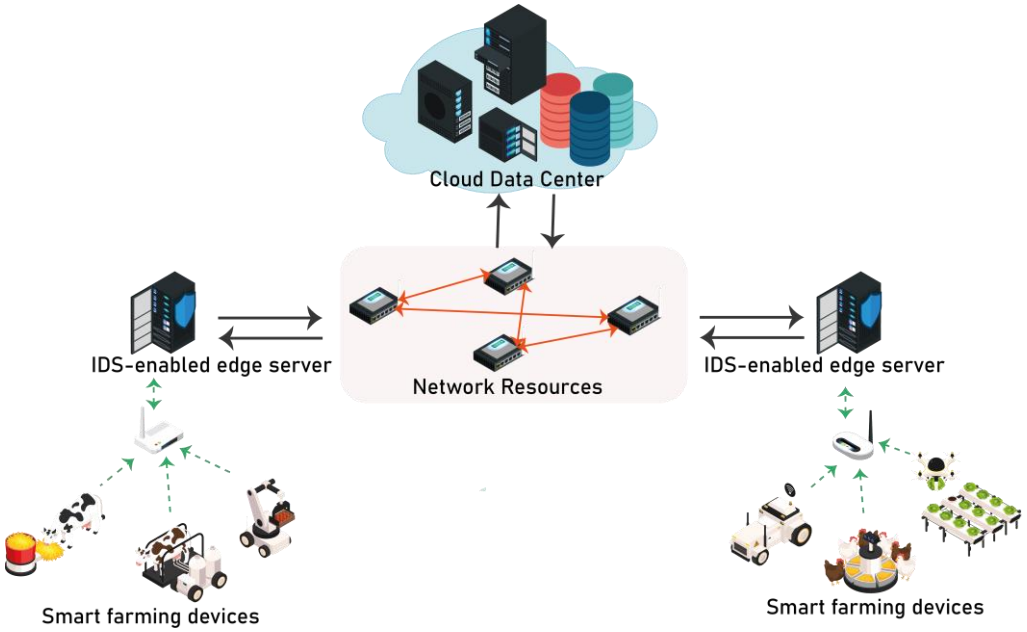


Figure 1: Proposed Smart Farming Network Architecture with IDS Deployment

3.2 Dataset & Preprocessing

The dataset used in this work is named Edge-IIoT set [18], a new cybersecurity dataset for IoT and IIoT applications developed in 2022. It includes real traffic from various IoT devices, tailored to detect intru- sions in both centralized and federated learning modes. The dataset contains 61 features and 20,952,648 samples, grouped into five attack types: DDoS, MITM, Malware, Scanning, and Injection.

The dataset is provided in two CSV files: one for ML and one for DL purposes (in our work, we use the DL file). The preprocessing starts by removing duplicate samples and any NaN values, if present. This is followed by encoding the data, where categorical features are transformed into numerical data using a label encoder. Next, the data is normalized by scaling it to the range [0,1] using the MinMaxScaler method. Finally, the data is split into three sets: 80% for training, 10% for validation, and 10% for testing.

3.3 CNN-GRU Hyperparameters Optimization based GA

GA was employed to optimize the hyperparameters of the CNN-GRU model by simulating a natural selection process, where the most effective hyperparameter sets evolve over successive generations. This iterative refinement process aims to identify the optimal combination of hyperparameters for enhancing the model's performance on the given dataset. Specifically, we optimized nine hyperparameters: the number of filters, kernel size, LeakyReLU alpha value, pooling size, dropout rate, number of neurons, learning rate, batch size, and number of epochs.

Each hyperparameter combination was treated as an individual in the population. The initial population of hyperparameter sets was generated randomly, and each set was evaluated based on its performance in training the CNN-GRU model. The evaluation criterion was the validation accuracy, which reflects how well the model generalizes to unseen data.

In each generation, the fitness of each individual was assessed, and the best-performing individuals were selected as parents to create offspring for the next generation. The crossover and mutation operations were applied to introduce variability and prevent the optimization process from getting stuck in local optima. Crossover involved combining features from pairs of

parents to produce new offspring, while mutation introduced random changes to some of the offspring's hyperparameters.

This process continued for a predefined number of generations, to evolve the population toward the most effective set of hyperparameters.

3.4 Federated Learning Training Process

Before constructing the IDS and employing it for network traffic classification, it is essential to complete the training process. In our paper, we propose leveraging federated learning to enable collaborative training between a server, which manages the process and optimizes the parameters, and clients, which perform the model training. Figure 2 depicts the interaction between the server and clients throughout the model training process, which involves several steps detailed below:

1. **Initialization of the model:** The server initializes the global hybrid model with a CNN-GRU architecture proposed in our previous work [10]. This architecture consists of a CNN component that helps minimize features and a GRU component. The proposed model has demonstrated high performance in centralized learning. The server then uses its local data and a hyperparameter optimizer based on a genetic algorithm (GA) to set the parameters of the global CNN-GRU model to the best possible values (denoted as *params* in the figure).

2. **Broadcast of the CNN-GRU to the clients:** The server sends the hybrid model with the best parameters to the clients. In our proposed strategy, we involve all available clients in the training process.

3. **Training of the local model:** Each client trains their local CNN-GRU model using their local data and then sends only the weights of the trained model to the server, which helps maintain the privacy of the clients.
4. **Aggregation of Weights:** The server first collects weights from all clients and assembles a local dataset with these received weights (denoted as *Rec weights* in the figure 2. It then applies the KNN algorithm to learn from these weights and generate new weights to update the global model. After updating the global model, the server distributes the updated weights back to the clients for the next training round.

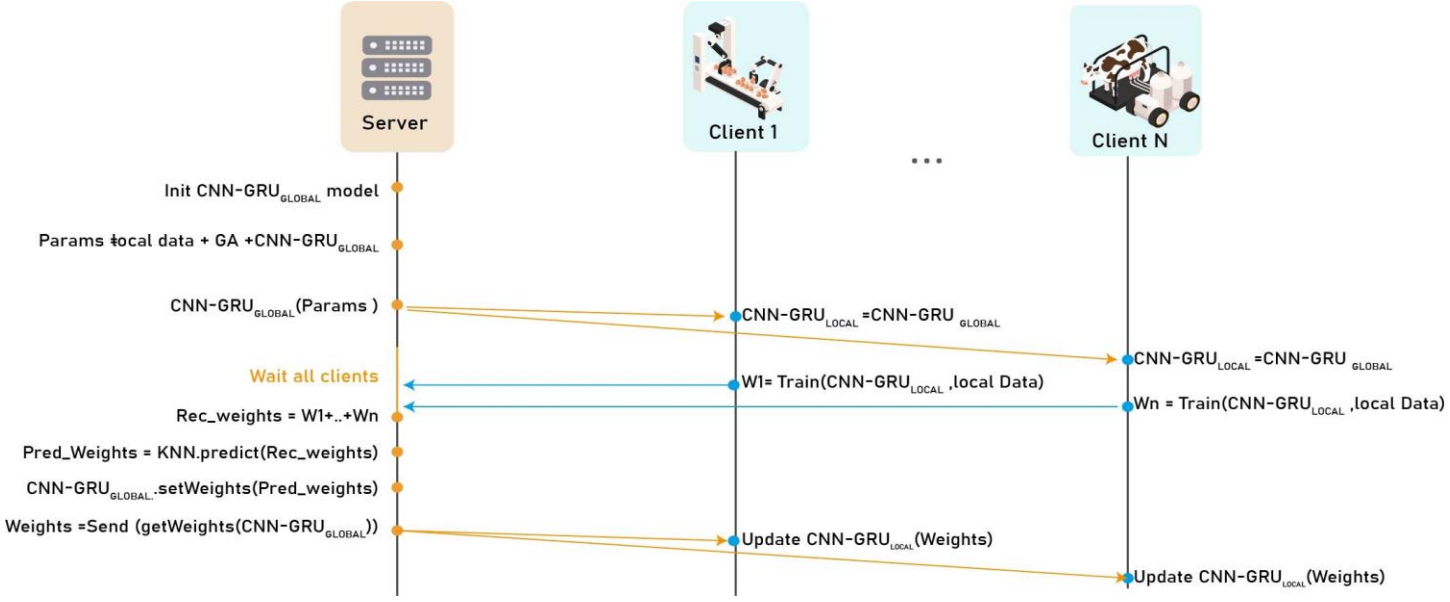


Figure 2: Client-Server Interaction Framework in Federated Learning for our proposed IDS Training

4 Experimental Setup

This section details the parameter settings used for the GA and KNN, as well as the CNN-GRU hyper- parameters that were optimized using GA. Additionally, we describe the evaluation metrics employed to assess the performance of our models.

4.1 Hyperparameter Setting

Table 2 outlines the key parameter settings for both GA and KNN.

Table 2: Parameter Settings for GA and KNN.

GA	KNN
mutation_rate = 0.1 crossover_rate = 0.5 generations=3 population-size=7	n-neighbors=3

Table 3 provides an overview of the hyperparameters optimized for the CNN-GRU model.

It details the specific range within which each hyperparameter was searched, along with the final hyperparameter value (HPO) selected by the GA.

Table 3: CNN-GRU Hyperparameters Optimized by GA.

Hyperparameter	Range	HPO
Number of Filters	F1[16, 32] F2[32, 64] F3[64, 128]	16 32 128
Kernel Size	[3, 5, 7, 9]	5
LeakyRelu(alpha)	(0.1, 0.6)	0.14812717256849303
Pooling Size	(2, 6)	2
Dropout Rate	(0.1, 0.6)	0.18124700433869764
Number of neurons	N1[20, 30] N2[30, 40] N3[40, 50]	20 40 40
Learning Rate	(0.00001, 0.01)	0.0044617071945977655
Batch Size	[256, 512, 1024, 2048]	1024
Number of Epochs	(5, 7)	6

4.2 Evaluation metrics

To evaluate the performance of the model, a confusion matrix is utilized. The following table 4 outlines the key components of the confusion matrix (TN, TP, FN, FP) and the derived evaluation metrics used to assess classification models.

Table 4: Confusion Matrix Terms and Evaluation Metrics

Metric	Description	Formula
True Positive (TP)	Correctly predicted positive instances	-
True Negative (TN)	Correctly predicted negative instances	-
False Positive (FP)	Incorrectly predicted as positive	-
False Negative (FN)	Incorrectly predicted as negative	-
Accuracy (Acc)	Proportion of correctly predicted instances.	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision (Pr)	Proportion of correct positive predictions	$\frac{TP}{TP+FP}$
Recall (Rc)	Proportion of actual positives identified correctly.	$\frac{TP}{TP+FN}$
F1-Score (F1)	Harmonic mean of precision and recall.	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$
False Positive Rate (FPR)	Proportion of negative instances incorrectly predicted as positive2.	$\frac{FP}{FP+TN}$

5 Results & Discussion

In this section, we present the results of our experiments, focusing on the performance metrics of the proposed CNN-GRU architecture across different approaches. We compare the results obtained from the centralized model with those from the federated learning setup using the GA-FedKNN approach. Additionally, we discuss the effectiveness of various aggregation methods and compare our approach with related work to highlight its contributions and advantages.

Table 5 presents the performance metrics for the GA-FedKNN model across various attack classes, including DDoS, MITM, Malware, Normal, Scanning, and Injection. The results indicate impressive accuracy, precision, recall, and F1scores for all classes, with values approaching or reaching 100%. The

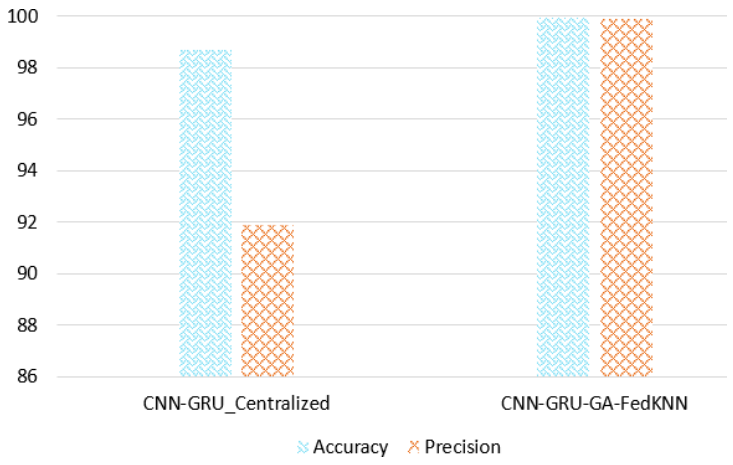


Figure 3: Performance Comparison between Centralized and Federated modes.

FPR is remarkably low across all categories, indicating the model's effectiveness in correctly

identifying and classifying different types of network traffic and attacks.

Table 5: Performance of GA-FedKNN

Class	Acc (%)	Pr (%)	Rc (%)	F1 (%)	FPR (%)
DDoS	100	100	100	100	0
MITM	99.99	99.89	99.85	99.87	5.67×10^{-5}
Malware	99.87	99.95	99.71	99.83	1.87×10^{-5}
Normal	100	100	100	100	0
Scanning	99.98	99.50	99.84	99.67	1.17×10^{-4}
Injection	100	100	100	100	0
Average	**99.97**	**99,89**	**99.90**	**99.90**	2.84×10^{-5}

5.1 Comparaison Between CNN-GRU in Federated Learning and Centralized Mode

We compared the CNN-GRU architecture from our previous work in centralized mode [10] with its implementation in a federated learning environment using the GA-FedKNN aggregation method. The results demonstrate significant improvements in accuracy and precision, as well as a considerable reduction in the FPR in the federated setup, as shown in Table 6 and Figure 3.

Table 6: Performance Metrics for CNN-GRU Architecture in Centralized and Federated modes.

Class	Acc (%)	Pr (%)	FPR (%)
CNN-GRU-Centralized[10]	98.70	91,89	0.70
CNN-GRU-GA-FedKNN	99,97	99,89	2.84×10^{-5}

5.2 Aggregation Methods Comparison

Table 7 provides a comparative analysis of the proposed GA-FedKNN aggregation method against other existing aggregation methods, all evaluated using the same CNN-GRU model optimized with GA in terms of Recall, F1-Score, and FPR.

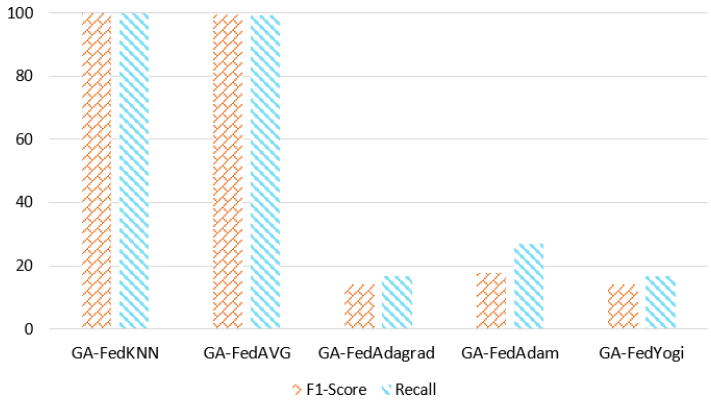


Figure 4: Performance Comparison of Different Models

The results highlight the superiority of the GA-FedKNN method, which achieved a Recall and F1-Score of 99.90% and an impressively low FPR of 2.84×10^{-5}.

In contrast, GA-FedAVG, while still performing well, shows a slight decrease in both Recall and F1-Score compared to GA-FedKNN (as shown in Figure 4), along with a higher FPR.

However, the performance of GA-FedAdagrad, GA-FedAdam, and GA-FedYogi are significantly lower. These methods show underperformance in Recall and F1-Score values, with GA-FedAdagrad and GA- FedYogi correctly detecting only one class at 100%, while completely missing the other classes (0%). This severe underperformance is further reflected in the FPR, where these methods exhibit a drastically high FPR for one class, reaching 100%. This suggests severe underfitting and highlights the inadequacy of these methods without sufficient training rounds, necessitating more than four rounds for improvement.

Table 7: Comparison with Other Models.

Model	Rc (%)	F1 (%)	FPR (%)
GA-FedKNN	99.90	99.90	2.84×10^{-5}
GA-FedAVG	99.08	99.49	2.18×10^{-4}
GA-FedAdagrad	16.66	14.04	16.66
GA-FedAdam	27.10	17.88	12.97
GA-FedYogi	16.66	14.04	16.66

5.3 Comparison with Related Work

An overview of the performance of recent studies compared to our work is presented in Table 8. We include the number of rounds used as a critical metric to illustrate communication overhead and the volume of data transmitted across the network. Only [16] specifies the number of rounds used, which is notably high at 15. Among the related works, the highest accuracy is reported by Paper [15], with 99.92% on the Car-Hacking dataset, and the highest F1 score of 99.6% is achieved by Paper [13] on the WSN-DS dataset. Our work outperforms other studies, achieving an accuracy of 99.97% and an F1 score of 99.90% on multi-attack detection with just 4 rounds, significantly reducing the amount of data circulated in the network.

Table 8: Performance Comparison Between Our Work and Related Studies

Ref.	Dataset	Rounds	Acc (%)	F1(%)
[16]	N-BaIoT	15	93.2	92.6
[12]	N-BaIoT	-	90.39	93.12
[14]	NSL-KDD	-	97.8	92.23
[13]	WSN-DS	-	99.7	99.6
[17]	CSE-CIC-IDS2018	-	87.98	-
[15]	Car-Hacking	-	99.92	-
Our Work	EdgeIIoT	4	**99.97**	**99.90**

6 Conclusion and Future Scope

In conclusion, this work presents an innovative IDS framework designed to enhance the security of smart farming devices through a novel aggregation method in Federated Learning, combined with the meta-heuristic Genetic Algorithm (GA). The proposed framework leverages a

CNN-GRU architecture, with nine key hyperparameters optimized by the GA. The server-side component employs its local data and optimized hyperparameters to set the parameters of the global CNN-GRU optimizer, which are then distributed to clients for local training. In the experimental phase, we conducted a comprehensive evaluation of the new aggregation method, which is based on the K-Nearest Neighbors (KNN) algorithm. This method was tested on the Edge-IIoT dataset Specifically, the proposed approach achieved an impressive accuracy of 99.97% and maintained a notably low false positive rate of 2.84×10^{-4}% across five distinct types of attacks. These results underscore the effectiveness of our approach in securing IoT environments against cyber threats.

Moving forward, we aim to explore the integration of other metaheuristic algorithms to optimize hyperparameters within each client, enabling local hyperparameter optimization. Additionally, we plan to investigate alternative aggregation methods further to enhance the robustness and adaptability of the proposed framework.

References

[1] GODFRAY, H. C. J., BEDDINGTON, J. R., CRUTE, I. R., HADDAD, L., LAWRENCE, D., MUIR, J. F., PRETTY, J., ROBINSON, S., THOMAS, S. M., & TOULMIN, C. (2010). Food security: the challenge of feeding 9 billion people. Science, 327 (5967), 812–818.

[2] HARBI, Y., MEDANI, K., GHERBI, C., SENOUCI, O., ALIOUAT, Z., & HAROUS, S. (2023). A systematic literature review of blockchain technology for Internet of Drones security. Arabian Journal for Science and Engineering, 48 (2), 1053–1074. Springer.

[3] JAYARAMAN, P. P., YAVARI, A., GEORGAKOPOULOS, D., MORSHED, A., & ZASLAVSKY, A. (2016). Internet of things platform for smart farming: Experiences and lessons learnt. Sensors, 16 (11), 1884.

[4] KAMILARIS, A., GAO, F., PRENAFETA-BOLDU, F. X., & ALI, M. I. (2016). Agri-IoT: A semantic framework for Internet of Things-enabled smart farming applications. In 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT) (pp. 442–447). IEEE.

[5] KHACHA, A., ALIOUAT, Z., HARBI, Y., GHERBI, C., SAADOUNI, R., & HAROUS, S. (2024).Landscape of learning techniques for intrusion detection system in IoT: A systematic literature review. Computers and Electrical Engineering, 120, 109725. Elsevier.

[6] KHACHA, A., SAADOUNI, R., HARBI, Y., GHERBI, C., ALIOUAT, Z., & HAROUS, S. (2024). Federated learning and convolutional autoencoder for robust anomaly detection in agricultural IoT. In 2024 International Conference of the African Federation of Operational Research Societies (AFROS) (pp. 1–5). IEEE.

[7] SAADOUNI, R., GHERBI, C., ALIOUAT, Z., HARBI, Y., & KHACHA, A. (2024). Intrusion detection systems for IoT based on bio-inspired and machine learning techniques: A systematic review of the literature. Cluster Computing, 27 (7), 8655–8681. Springer.

[8] MCMAHAN, B., MOORE, E., RAMAGE, D., HAMPSON, S., & AGUERA Y ARCAS, B. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics (pp. 1273–1282). PMLR.

[9] CAMPOS, E. M., SAURA, P. F., GONZ ´ALEZ-VIDAL, A., HERN ´ANDEZ-RAMOS, J. L., BERNABE, J. B., BALDINI, G., & SKARMETA, A. (2022). Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges. Computer Networks, 203, 108661.

[10] SAADOUNI, R., KHACHA, A., HARBI, Y., GHERBI, C., HAROUS, S., & ALIOUAT, Z. (2023). Secure IIoT networks with hybrid CNN-GRU model using Edge-IIoTset. In 2023 15th International Conference on Innovations in Information Technology (IIT) (pp. 150–155). IEEE.

[11] MIRJALILI, S. (2019). Genetic algorithm. In Evolutionary algorithms and neural networks: Theory and applications (pp. 43–55). Springer.

[12] OLANREWAJU-GEORGE, B., & PRANGGONO, B. (2024). Federated Learning-based Intrusion Detection System for the Internet of Things using Unsupervised and Supervised Deep Learning Models. Cyber Security and Applications, 100068.

[13] BUKHARI, S. M. S., ZAFAR, M. H., ABOU HOURAN, M., MOOSAVI, S. K. R., MANSOOR, M., MUAAZ, M., & SANFILIPPO, F. (2024). Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with SCNN-Bi-LSTM for enhanced reliability. Ad Hoc Networks, 155, 103407.

[14] JIA, Y., LIN, F., & SUN, Y. (2024). A novel federated learning aggregation algorithm for AIoT intrusion detection. IET Communications, 18 (7), 429–436.

[15] BHAVSAR, M., BEKELE, Y., ROY, K., KELLY, J., & LIMBRICK, D. (2024). FL-IDS: Federated Learning-Based Intrusion Detection System Using Edge Devices for Transportation IoT. IEEE Access.

[16] UMAIR, M., TAN, W.-H., & FOO, Y.-L. (2023). Network Intrusion Detection using Dynamic Weighted Aggregation Federated Learning. In 2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE) (pp. 1–6). IEEE.

[17] UMAIR, M., TAN, W.-H., & FOO, Y.-L. (2024). Dynamic Federated Learning Aggregation for Enhanced Intrusion Detection in IoT Attacks. In 2024 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (pp. 524–529). IEEE.

[18] FERRAG, M. A., FRIHA, O., HAMOUDA, D., MAGLARAS, L., & JANICKE, H. (2022). Edge- IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access, 10, 40281–40306.