Open Access ISSN: 3007-1437 (Online), ISSN: 3007-1429 (Print)

IRAQI Academic Scientific Journals

Alkadhim Journal for Computer Science (KJCS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

KJCS ALKADHIM JOURNAL FOR COMPUTER SCIENCE

Federated Learning for Early Detection of Advanced Persistent Threats in IoT Networks

[1]Bassam Noori Shaker*

[1]*Computer Science Department, College of Computer Science and Information Technology, University of Al-Qadisiyah, Al Diwaniyah*– Country

Article information

Article history:
Received: November, 27, 2025
Accepted: December, 13, 2025
Available online: December, 25, 2025

Keywords:
Intrusion detection system,
federated learning,
initial compromised

Corresponding Author:
Bassam Noori Shaker
bassamsat@qu.edu.iq

DOI:
https://doi.org/10.61710/kjcs.v3i4.140

Abstract

. In the era of connected IoT devices, ensuring cyber security while preserving data privacy is increasingly critical. Federated learning offers a promising approach by enabling collaborative training of detection systems without sharing raw data. This paper presents a novel federated Intrusion Detection System (IDS) based on XG Boost algorithm, and for the first time designed to detect initial compromise (I.C.) phase of Advanced Persistent Threats (APTs) in distributed Internet of Things (IoT) environments. By leveraging the federated framework, the IDS achieves robust detection across multiple devices while maintaining privacy and minimizing computational overhead. Extensive simulation results indicated that our proposed method achieved a precision of 97%, recall of 100%, and F1-score of 98%, providing a practical and efficient solution for real-world IoT security challenges.

1. Introduction

The widespread proliferation of Internet of things (IoT) devices has revolutionized industries and daily life, connecting everything from household appliances to industrial machinery through the Internet [1]. However, this rapid adoption has also introduced significant security challenges, particularly in environments with constrained resources and evolving threats. Advanced Persistent Threats (APTs) is stand out as one of the most sophisticated and prolonged forms of cyber-attacks in the recent years. APTs are orchestrated by highly skilled attackers who aim to infiltrate net-works that maintain undetected access and extract sensitive information or cause disruption over an extended period [2] [3]. These attacks are typically executed in multiple phases which including reconnaissance, initial compromise (I.C.), persistence, privilege escalation, lateral movement, data exfiltration, and covering tracks.

The I.C. phase of APT is a critical stage where attackers gain a foothold in the target system often exploiting IoT device vulnerabilities. In this phase the attackers may deploy phishing campaigns, exploit unpatched software, or utilize insecure device configurations to breach the network [4]. IoT devices are particularly vulnerable during this phase due to their limited processing power, lack of regular updates, and inadequate authentication

mechanisms [5]. After the device compromised the attackers may install backdoors or malware to secure their access and proceed with deeper infiltration [6][7]. To combat this, organizations must focus on securing IoT entry points by implementing strong access controls, patch management systems, real-time threat detection tools and Intrusion Detection Systems (IDS). The last, plays a crucial role by monitoring and analyzing incoming network traffic to detect and respond to anomalies or malicious activities in real-time [8][9]. These measures are essential to counter APTs effectively and safeguard IoT infrastructures despite their inherent constraints.

The role of IDS as a classification problem in detecting attacks based on the analyzing the network traffic and distinguish between normal and malicious activities [10]. By leveraging machine learning (ML), IDS treats incoming network traffic as data points consisting of features such as IP addresses, ports, protocols, and payload sizes that serve as inputs to a classification algorithm and these features are not equally relevant to detecting malicious traffic, and they may including irrelevant ones which increase the time required for training the classification algorithm and for predicting traffic. By reducing number of features can significantly improve the efficiency of an IDS by shortening training time, reducing model complexity and enhancing detection accuracy and this will be particularly efficient in IoT environments [11].

Building on this, Federated Learning (FL) provides an effective paradigm for addressing both efficiency and privacy challenges in IDS development by enabling devices to train models locally and share only model updates, ensuring that raw data remains private while still contributing to a global model[12]. This decentralized approach not only safeguards sensitive information but also facilitates a collaborative defense mechanism that aggregates knowledge from distributed devices to adapt to evolving attack patterns. By combining FL with lightweight detection algorithms tailored to IoT constraints, IDS can achieve high detection accuracy, enhanced scalability, and robust privacy protection, making it well-suited for securing IoT ecosystems against sophisticated threats such as APTs.

The main contribution of this study is the development of an IDS based on federated learning that can identify APT attacks at an early stage, specifically during the Initial Compromise (I.C.) phase. The proposed framework enables collaborative training among clients while preserving their privacy by keeping all data local. By integrating FL with an APT-focused detection model, the approach enhances early-stage threat identification, maintains strong privacy protection across distributed environments, and provides a practical and robust solution suitable for real-world deployment.

2. Related Works

Several studies have explored the use of FL for IDS in IoT environments [13]. Most of these studies employ various deep learning (DL) architectures during the development process. For instance, Rey et al. [14] propose a framework that employs two types of DL methods: Multi-Layer Perceptron's (MLPs) for supervised learning and Auto Encoders (AE) for unsupervised learning, to detect malware on IoT devices. Their study on the N-BaIoT dataset demonstrate that the federated approach achieves nearly the same performance (99.38% accuracy) as the centralized model (99.42% accuracy) when evaluated on a new, unseen device. Similarly, the authors in [15] compare unsupervised AE with supervised Deep Neural Networks (DNNs) in an FL-based IDS, concluding that the unsupervised AE model trained via FL is the best overall performance, particularly excelling in achieving a lower False Positive Rate. While Regan et al. [16] proposed FL model used an AE to detect botnet attacks. The model achieved 98% anomaly detection accuracy using features like source IP, MAC-IP, and destination IP.

Other works have integrated more complex DL architectures into FL frameworks for industrial applications, claiming to achieve high results. The "Deep Fed" framework that proposed by Li et al. [17], combines a Convolutional Neural Network (CNN) and a Gated Recurrent Unit (GRU) to create a highly effective IDS for industrial Cyber–Physical Systems. In a 7-client setup, their model achieves performance of 99.20% accuracy, 98.85% precision, and 97.47% recall, that is comparable to an ideal centralized model. Furthermore in [18], Mothukuri et al. utilize GRU models in an FL system to proactively detect intrusions. The proposed framework leverages on-device computational resources for training and employs a multi-layer GRU architecture, resulting in 99.5% accuracy rate.

In simpler model with one DL architecture Rehman et al. [19] design a fog-enabled FL system using CNN. Their framework yield 93.4% accuracy on the Edge-IIoTset and 95.8% accuracy on the CIC-IDS2017 dataset [20], with the global federated model consistently demonstrating superior performance over local models.

From the related works that viewed in this section two main research gaps remain. First, most existing studies rely on DL models such as AEs, CNNs, and recurrent neural network (RNN) variants like GRUs [21], [22]. While effective, these models introduce additional complexity due to the architectural requirements of DL algorithms, which poses challenges for IoT devices with limited memory and processing power. For addressing this gap, in the proposed model we employed XG Boost which is a tree-based method that is more efficient for handling tabular network traffic data and better suited for resource-constrained environments. Second, the application of XG Boost with FL in the cyber security is still not well studied. This work helps fill that gap by proposing a novel FL framework that leverages the XG Boost algorithm specifically for detecting APTs and combines with the privacy-preserving benefits of FL with the proven performance and efficiency of gradient-boosted trees for complex classification tasks.

3. Theoretical Part

This section introduces the main techniques used to develop the IDS within FL approaches based on the XG Boost algorithm.

3.1 Federated learning

FL is a new paradigm that can significantly support IDS targeting IoT devices by offering solutions to the critical challenge of data privacy and security [12]. Traditional ML requires the transfer of sensitive data from IoT devices to centralized servers for model training [23]. This arises as a serious concern in relation to the risk of data breach and violation of regulations regarding data privacy. With FL, in particular, each device trains a shared model with only their private local data, without sharing the raw information with the central server [24]. This decentralized approach improves privacy by keeping sensitive data on the device and allows for the aggregation of diverse data patterns across different IoT environments, leading to a more robust IDS. This is minimizing on the bandwidth and resources of computation, hence highly suitable for devices in IoT that normally have very low resources. In general, the proposed technique enables collaborative learning with respect to security concerns pertaining to the given heterogeneous device categories to increase the general security concepts in IoT networks.

3.2 XG Boost

XG Boost is a powerful ML algorithm widely used for classification tasks, especially when dealing with tabular data. It operates by constructing a series of decision trees in a boosting manner, where each new tree corrects the errors of the previous ones[25]. There are many advantages in choosing XG Boost for developing an IDS specifically with imbalanced class distributions, which is common in cyber security datasets. Where the algorithm's ability to handle missing values, its regularization features and its robustness against over fitting make it suitable for identifying rare attack patterns amidst a larger volume of benign samples. Furthermore the algorithm provides built-in functionalities for adjusting the class weights, which allowing it to give more importance to minority classes, thus improving detection rates for less frequent intrusion types [26], This makes it an effective choice for enhancing the performance and reliability of IDS in the face of class imbalance.

4. Experimental Procedure

This section introduced the main components of the experimental design and describe the dataset that employed in developing the proposed IDS based on FL that capable of detecting APTs at an early stage.

4.1 Dataset

SCVIC-APT-2021 dataset is used for model evaluation process of the proposed model that serving as a comprehensive and realistic benchmark designed for APT detection in network traffic that developed by the Smart Connected Vehicles Innovation Centre (SCVIC) at the University of Ottawa [27]. The I.C. phase in this dataset simulates IoT network environment by incorporating smart sensors and IP cameras, which providing a relevant and challenging scenario for APT detection [28]. We used this dataset on tour proposed model because it covers phases of APT attack, including the I.C. phase, which is the target of research. The dataset includes 84 features that capture various network traffic characteristics essential for detecting APTs, with each data instance labeled with ground truth information for each APT stage. In this study only 4 features are used to develop our proposed model based on feature selection method shown in [29].

4.2 Experimental setup

Before describing the steps of the FL implementation, the dataset is filtered to include only two of the five classes: normal traffic (307,817 instances) and I.C. traffic (150 instances). The dataset consists of features extracted from network traffic flows—such as packet size and flow duration—using specialized tools like CICF low Meter and Net Flow. However, not all of these features can be directly used with ML algorithms. Therefore, several preprocessing steps are applied, including the removal of biased features, handling of null values, and conversion of categorical attributes into numerical form using the label encoder. Next the preprocess dataset horizontally split among the clients, where each client (IoT device) receives a subset of data instances (rows) with all set of features (columns). This configuration ensures that every client has a representative sample covering the complete feature space. Additionally, stratified sampling method was applied to maintain a consistent proportion of class instances across all clients.

Figure 1 shows the proposed FL system's framework that summarizes the core contribution of this work also is provide a visual summary of the model architecture highlighting the server and client components and the collaborative learning process. The IDS leverages FL with XG Boost, a gradient-boosting framework optimized for structured and tabular data, making it highly effective for identifying APTs in varied distributed environments such as IoT. The training process is initiated by the server side, as follows:

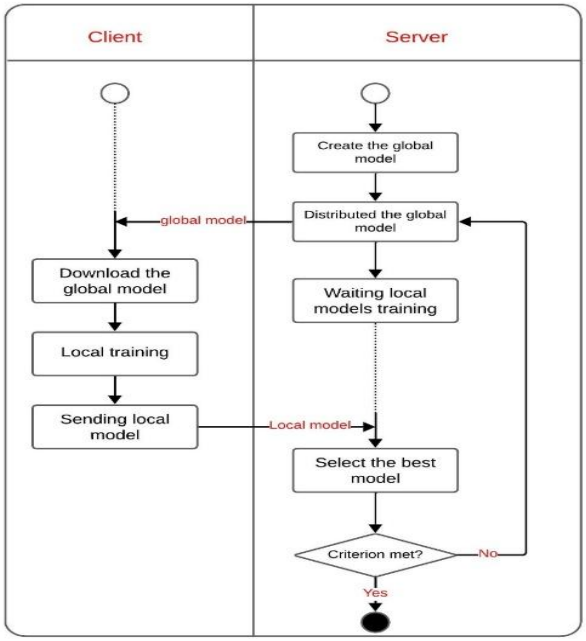

Figure (1): Federated Learning Workflow

- **Creating the Global Model.** The server establishing a global model P^o with customized XGBoost parameters that shown in Table 1 aimed at optimizing performance across diverse client datasets.

This configuration is then compiled into the global model parameters which outline the structure and tuning characteristics of the XG Boost model.

Table 1. XG Boost Hyper parameters

Hyper parameter	Value	Description
n_estimators	100	Number of boosting rounds (iterations).
learning_rate	0.3	Step size shrinkage used in update to prevent over fitting.
max_depth	6	Maximum depth of a tree. Controls model complexity.
min_child_weight	1	Minimum sum of instance weight (hessian) needed in a child.
gamma	0	Minimum loss reduction required to make a further partition on a leaf node.
subsample	1	Fraction of observations to be randomly sampled for each tree.
colsample_bytree	1	Fraction of features to be randomly sampled for each tree.

Distributing the Global Model to Clients. When P^o is defined and all parameters are set and the server distributes the P^o among to all N clients:

$$P_i^o = P^o, \forall\, i \in \{1,2,3,\ldots,N\}$$

Each client i will receiving a copy of the P^o then the client train his local model P_i^o using his own datasets D_i, this will producing trees that are structured and optimized based on P^o but tailored to client model specific data and that will ensures consistency across all client models by maintaining the same core XGBoost configuration which enabling an effective and coordinated FL process. When clients download P^o, several key processes occur:

Model Initialization. Each client initializes its local XG Boost model using the parameters specified in the global model P^o.

Local Training. Clients begin the local training process using their own dataset D_i , where the local data for each one splitting to 70% for training, 20% for testing and 10% for validation process, then they train their models for building their decision trees according to the global parameters that received from the server. This local training allows clients to tailor the model to their specific data distributions which enhancing its relevance and accuracy.

Performance Evaluation. After training on their local data each client evaluate the performance of their models using the **macro average F1-score** which helps clients assess how well the model has learned from their local data.

Model Update Generation. Each client generates an updated model based on its training and rather than sending the entire model back to the server they typically prepare a summary of their model updates or local gradients, which captures the learning outcomes from their training process.

Communication with the Server. Clients send their model updates back to the server for evaluation then the server assesses each client model based on the macro-averaged F1-score and selects the best-performing local model. This selected model is then distributed to the remaining clients, enabling them to continue training with the most effective model from the previous round.

Iterative Process. After each round t, the server collects updates from clients, refines or updates the global model based on those updates, and produces a new version of the model, $\boldsymbol{P^{t+1}}$, for the next iteration. This iterative process continues as the model P evolves through rounds $t = 1,2,3, \ldots$ until the model meets a convergence.

In summary, when clients download the global model, they initialize and train their local versions, evaluate their performance, and send back updates, contributing to the collaborative learning process without compromising data privacy. This allows the global model to adapt and improve over multiple rounds of training.

5. Results and Discussion

This section presents the results of applying IDS based on FL to detect APTs at the I.C. phase with only 4 features listed in Table 2, that selected based on the study in [29], the number of FL rounds was set to 10, with the number of clients ranging from 2 to 5 for training local models. At the end of each round, the best-performing model was selected and generalized as the global model. Furthermore, early stopping was employed to avoid over fitting.

Table 2. Description of selected features

Feature	Description
Idle Max	The maximum idle time observed between two packets in a flow, indicating periods of inactivity.
Fwd Seg Size Min	The minimum size of segments in the forward direction, reflecting the smallest packet size sent by the source.
Flow IAT Std	The standard deviation of the inter-arrival time (IAT) between packets in a flow, measuring the variability in packet timing.
Bwd Init Win Bytes	The initial window size in bytes set by the receiver in the backward direction, indicating the TCP flow control mechanism.

Table 3 shows the relationship between the number of clients and the number of FL rounds required to reach the point of no further improvement in the Macro Average F1 score. These values were obtained by applying an early stopping technique that returned the best values during the training process.

Table 3. Performance Evaluation for Detect I.C. Phase

No. of Client	No. of Round	Performance metrics		
		Precision	Recall	F1-Score
2	4	94%	100%	97%
3	**6**	**97%**	**100%**	**98%**
4	**6**	**97%**	**100%**	**98%**
5	4	91%	97%	94%

The model requires only four rounds to reach optimal performance when two clients participated in the training process because of when the number of clients decrease that lead to the number of training samples per client increases, allowing the model to converge more quickly to a stable solution.

With 3 or 4 clients, the model required six training rounds to achieve better performance, particularly in terms of precision. This improvement was due to the greater diversity of data across clients that enables XG Boost to more effectively differentiate true positives from false positives. Also and as shown in Fig. 2, the training process successfully converged to an optimal solution while with only 2 clients, the smaller and less varied dataset made the model more susceptible to false positives, resulting in lower precision. Figure.2 shows the converges to the best solution during the training process.

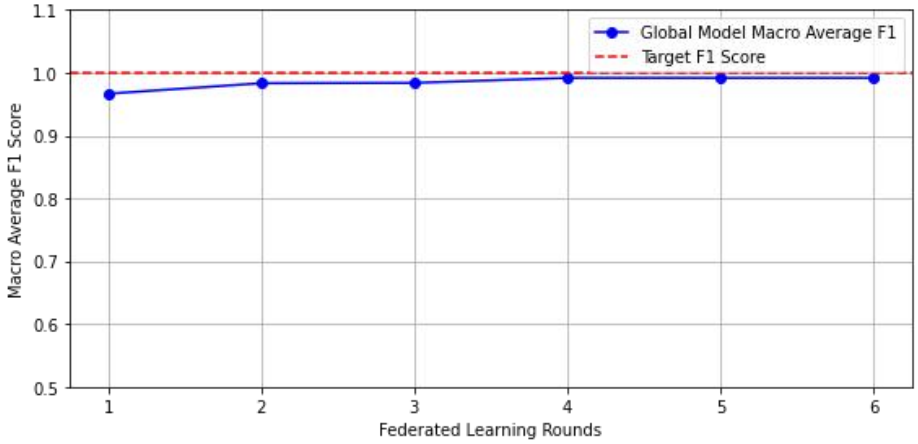

Figure2: Global Model Macro Average F1 Score Across FL Rounds

When the number of clients increases to 5, the model needs only four rounds to converge, but this came with a drop in performance metrics. With more clients, each had access to a smaller portion of the data which lead in reducing the data availability for each client and impacts overall performance. As a result, further rounds were unnecessary since additional training did not lead to improvements in the model's metrics.

6. Conclusions

This paper introduces a novel IDS that leverages FL and the XG Boost algorithm to detect the I.C. phase of APTs in IoT environments. The proposed method harnesses the benefits of FL by integrating it with XG Boost; a global model is constructed and then distributed to multiple local models, each trained on its local data. Early stopping was employed during the XG Boost training phase to predict types of network traffic. The best results achieved were a precision of 97%, a recall of 100%, and an F1-score of 98% when the model was trained with 3 to 4 clients. Future work could expand on this model by evaluating its performance against other phases of the APT lifecycle, testing its scalability with larger number of clients, and exploring its resilience to adversarial attacks within the federated framework.

References

[1] F. John Dian, R. Vahidnia, and A. Rahmati, "Wearables and the Internet of Things (IoT), Applications, Opportunities, and Challenges: A Survey," *IEEE Access*, vol. 8, pp. 69200–69211, 2020, doi: 10.1109/ACCESS.2020.2986329.

[2] T. Steffens, *Attribution of Advanced Persistent Threats*. Springer, 2020.

[3] O. Access, K. M. Khudhair, B. M. Khudhair, and R. R. Hadi, "Cognitive Honeypots AI-Enhanced Deception for Proactive Threat," vol. 3, no. 3, pp. 55–70, 2025.

[4] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 8, p. 1, 2019.

[5] F. Pereira, R. Correia, P. Pinho, S. I. Lopes, and N. B. Carvalho, "Challenges in resource-constrained iot devices: Energy and communication as critical success factors for future iot deployment," *Sensors*, vol. 20, no. 22, pp. 1–30, 2020.

[6] P. M. Chanal and M. S. Kakkasageri, "Preserving Data Confidentiality in Internet of Things," *SN Computer Science*, vol. 2, no. 1, pp. 1–12, 2021, doi: 10.1007/s42979-020-00429-z.

[7] A. Radovici, C. Rusu, and R. Serban, "A Survey of IoT Security Threats and Solutions," *Proceedings - 17th RoEduNet IEEE International Conference: Networking in Education and Research, RoEduNet 2018*, vol. 9, no. 45, 2018, doi: 10.1109/ROEDUNET.2018.8514146.

[8] G. Eric and A. Jurcut, "Intrusion Detection in Internet of Things Systems : A Review on Design Approaches Leveraging Multi-Access Edge," *Sensors*, vol. 22, pp. 1–33, 2022.

[9] O. Access and I. Technology, "A Comprehensive Review of Intrusion Detection Systems in IoT Networks Using ML and DL Techniques," vol. 3, no. 2, 2025.

[10] H. Huang, H. Al-Azzawi, and H. Brani, *Network Traffic Anomaly Detection*. 2014. [Online]. Available: http://arxiv.org/abs/1402.0856

[11] X. Jaw, Ebrima and Wang, "Feature selection and ensemble-based intrusion detection system: an efficient and comprehensive approach," *Symmetry*, vol. 13, p. 1764, 2021.

[12] N. A. Al-Athba Al-Marri, B. S. Ciftler, and M. M. Abdallah, "Federated Mimic Learning for Privacy Preserving Intrusion Detection," *2020 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2020*, 2020, doi: 10.1109/BlackSeaCom48709.2020.9234959.

[13] O. Access, B. S. Zynal, A. T. Lateef, S. A. Jebur, and H. A. Naser, "Improving Communication Performance Through Fiber Amplifier," vol. 2, no. 2, pp. 1–9, 2024.

[14] V. Rey, P. M. Sánchez Sánchez, A. Huertas Celdrán, and G. Bovet, "Federated learning for malware detection in IoT devices," *Computer Networks*, vol. 204, no. December 2021, p. 108693, 2022, doi: 10.1016/j.comnet.2021.108693.

[15] B. Olanrewaju-George and B. Pranggono, "Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models," *Cyber Security and Applications*, vol. 3, no. October 2023, p. 100068, 2025, doi: 10.1016/j.csa.2024.100068.

[16] C. Regan, M. Nasajpour, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and K.-K. R. Choo, "Federated IoT attack detection using decentralized edge data," *Machine Learning with Applications*, vol. 8, no. November 2021, p. 100263, 2022, doi: 10.1016/j.mlwa.2022.100263.

[17] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber-Physical Systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5615–5624, 2021, doi: 10.1109/TII.2020.3023430.

[18] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated-Learning-Based Anomaly Detection for IoT Security Attacks," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2545–2554, 2022, doi: 10.1109/JIOT.2021.3077803.

[19] T. Rehman, N. Tariq, F. A. Khan, and S. U. Rehman, "FFL-IDS: A Fog-Enabled Federated Learning-Based Intrusion Detection System to Counter Jamming and Spoofing Attacks for the Industrial Internet of Things," *Sensors*, vol. 25, no. 1, pp. 1–34, 2025, doi: 10.3390/s25010010.

[20] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection

Systems," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 3.24 Special Issue 24, pp. 479–482, 2018.

[21] S. A. Jebur, M. A. Mohammed, L. R. Ali, and D. H. Abd, "MIX - Hybrid Convolutional Neural Network Framework with Explainable Artificial Intelligence for Fig Leaves Disease Detection," *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 4, pp. 881–895, 2025, doi: 10.22266/ijies2025.0531.57.

[22] L. R. Al-Khazraji, A. R. Abbas, and A. S. Jamil, "Generating Various Deep Dream Images Through Maximizing the Loss Function of Particular Layers Using Inception-v3 and Inception-ResNet-V2 Models," *Iraqi Journal of Science*, vol. 65, no. 6, pp. 3468–3483, 2024, doi: 10.24996/ijs.2024.65.6.39.

[23] A. Kumar, K. Abhishek, M. R. Ghalib, A. Shankar, and X. Cheng, "Intrusion detection and prevention system for an IoT environment," *Digital Communications and Networks*, vol. 8, no. 4, pp. 540–551, 2022, doi: 10.1016/j.dcan.2022.05.027.

[24] G. Drainakis, "Federated vs. centralized machine learning under privacy-elastic users: A comparative analysis," *19th International Symposium on Network Computing and Applications (NCA).*, 2020.

[25] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-Augu, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[26] K. Ghosh, C. Bellinger, R. Corizzo, P. Branco, B. Krawczyk, and N. Japkowicz, *The class imbalance problem in deep learning*, vol. 113, no. 7. Springer US, 2024. doi: 10.1007/s10994-022-06268-8.

[27] J. Liu *et al.*, "A new realistic benchmark for advanced persistent threats in network traffic," *IEEE Networking Letters*, vol. 4, no. 3, pp. 162–166, 2022.

[28] Y. Shen, M. Simsek, B. Kantarci, H. T. Mouftah, M. Bagheri, and P. Djukic, "Prior Knowledge based Advanced Persistent Threats Detection for IoT in a Realistic Benchmark," in *GLOBECOM 2022 IEEE Global Communications Conference*, 2022, pp. 3551–3556.

[29] B. N. Shaker, B. Al-Musawi, and M. F. Hassan, "Explainable AI for enhancing IDS against advanced persistent kill chain," *Cluster Computing*, vol. 28, no. 7, p. 459, 2025, doi: 10.1007/s10586-025-05219-x.