The intrinsic convenience of federated learning in malware IoT detection

Chiara Camerota
DINFO
Università degli Studi di Firenze
Florence, ITALY
chiara.camerota@unifi.it

Tommaso Pecorella
DINFO
Università degli Studi di Firenze
Florence, ITALY
tommaso.pecorella@unifi.it

Andrew D. Bagdanov
DINFO
Università degli Studi di Firenze
Florence, ITALY
andrew.bagdanov@unifi.it

***Abstract*—The Internet of Things is emerging as a key concept, defining a network of interconnected devices capable of seamless data collection, exchange, and analysis. However, due to their emphasis on simplicity, these devices are often vulnerable to malware attacks. This study examines the potential of machine learning methods, specifically in the context of Federated Learning, to enhance privacy protection and to benefit from IoT's decentralized nature, such as the low overhead traffic. The proposed approach is a federated machine learning algorithm based on a central aggregator and several clients. The study aims to conduct a comprehensive analysis using the IOT-23 dataset, which contains real and labeled instances of malware infections. The test outcomes demonstrate that the proposed approach outperforms centralized approaches regarding the global area under the precision-recall curve (AUPRC) and variance, with a significance level of 0.05.**

***Index Terms*—AI - ML techniques, anomaly detection, federated learning, Internet of Things**

I. Introduction

The Internet of Things (IoT) refers to a network of interconnected devices and objects that can collect, exchange, and analyze data, allowing them to communicate and interact seamlessly [1]. These devices feature lighter protocols, lower power consumption, and compact shapes, allowing flexibility and adaptability [2]. In addition, this kind of Internet can be implemented in several wireless networks in order to become adaptable in various use cases; for example, a LoRaWAN connection can be chosen in smart cities, while for smart homes, the Z-Wave strategy is more appropriate. Likewise, smart devices have a wide range of applications, many in sensitive areas. However, no standard supports all smart devices, and built-in security mechanisms are not yet standardized. Furthermore, no proven security methods to guarantee the digital security of these infrastructures [3]. In addition, IoT devices prioritize simple operation over robust security measures, making them vulnerable to malicious users who can coordinate attacks through malware designed to cause damage, disrupt, or gain unauthorized access to a system [4]. Therefore, it is necessary to implement an intelligent, light, and flexible solution that can learn various attack patterns and does not interfere with the network's bandwidth or usability.

Furthermore, these methods must be able to forget learned patterns and re-learn when malware evolves [5]. In literature, deterministic techniques are commonly used in conjunction or not with Machine Learning (ML) methods, depending on the main aspect the tool should preserve. A key work that helps us understand this concept is offered by [6], which discusses anomaly detection analyses for different phases of the malware life cycle. Statistical analysis is utilized during the pre-execution phase, dynamic analysis is employed during the execution step, and memory analysis is conducted during the post-execution step. Conversely, malware detection is often considered a stand-alone problem and does not consider the network's topology or distributed nature. In [7], the authors take into account specific IoT challenges, such as privacy, but do not consider the impact of the model on the wireless network. Deeper, centralized models often are not appropriate for IoT cases because they can be accurate but may make the communication channel busy due to data exchange [8].

Federated learning (FL) is a suitable strategy to minimize data exchange. Furthermore, it is a distributed machine learning algorithm that benefits from the topology of IoT networks. FL employs an iterative approach with discrete interactions between the client and server, known as federated learning rounds, to achieve results comparable to those of traditional machine learning models. Therefore, this method has the potential to access a large amount of data while maintaining privacy guarantees.

This paper addresses the critical challenge of malware detection within the Internet of Things paradigm, focusing on the pivotal role of advanced models such as Federated Learning. We explore the adaptability of these models to the unique constraints of IoT environments and thoroughly examine the open challenges associated with deploying machine learning models in real-world scenarios where resources are finite and operational constraints are stringent. Our approach centers on developing a federated machine learning classifier as a practical and deployable solution for identifying malicious traffic within IoT networks. Using data generated from traffic packet analyzers, we outline effective strategies for data definition, preprocessing, and the deployment of ML algorithms. This

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE0000001 - program “RESTART”.

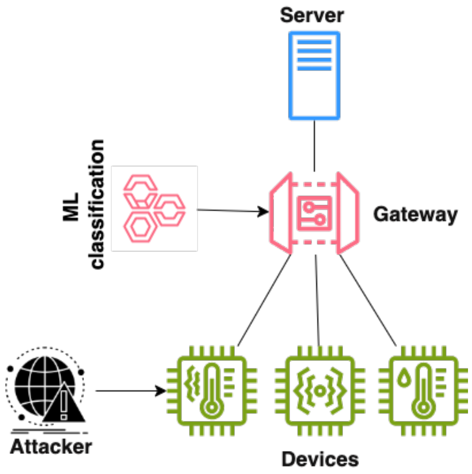

Fig. 1. A simplified representation of IoT system. The devices are low-powered, so the ML classifier is installed on the gateway to enhance the environment's security.

results in a robust model characterized by low variance in accuracy, making it suitable for resource-constrained environments. Additionally, we highlight the advantages of FL in reducing information leakage from devices and enhancing network performance. Our findings underscore the importance of advanced ML techniques in strengthening IoT ecosystems against evolving cyber threats, ensuring the integrity of interconnected devices. For further clarification, please refer to the diagram in Fig. 1. This illustrates the deployment of the ML classifier on the gateway and the execution of the attack on the devices. It should first be noted that the server is not being considered a point of interest in this context. The paper is structured as follows. Section II examines the current state-of-the-art in malware detection, which is the basis for the proposed model and methodology, such as the definition of problems explained in III. Section IV is dedicated to the proposed model and methods, in which the primary aspect and novelty of the proposed model are discussed. The next section discusses the results and their implications. After that, we present the future challenges and the conclusions.

II. State of Art

The field of malware detection is extensive and encompasses numerous case studies. Malware attacks increasingly threaten both the IoT and traditional systems [9].

The literature contains several strategies for addressing this issue, which are collected, explored, and classified due to the heterogeneity of the problem and the available solutions. Machine learning techniques can help to analyze dynamics and memory by generalizing tasks with relevant data. Nascita's study shows that using customized data for IoT to feed an ML model improves performance [10]. This is also supported by [11], who demonstrate how to mitigate damage to IoT devices by intelligently identifying both known and emerging IoT malware. They achieve this by analyzing several device properties, transforming them into images, and applying dynamic analysis. However, these studies often overlook crucial factors such as the high cost of collecting extensive datasets, the scarcity of available data, and, most importantly, the overhead traffic generated by these approaches.

Another important aspect of machine learning models is their ability to be integrated into a cloud environment. However, this presents a complex challenge. A decentralized model should be considered over a centralized model to mitigate data exchange and communication channel availability issues. For this reason, Federated Learning techniques address this challenge and ensure privacy. They enable machine learning model training on the device and exchange only parameters with the server without sharing the user's information.

An example of FL applied to an IoT network is [12]. The authors present a model where the federated approach prioritizes participant privacy while achieving results comparable to those of centralized models. The authors also investigate the safety and robustness of this approach and demonstrate that the baseline model aggregation averaging step is highly vulnerable to attacks, even with a single adversary. However, they do not consider the impact on the network or analyze the comparison regarding the results' variance.

Another reason to consider decentralized methods over centralized ones is the challenge of combining anonymized data from heterogeneous and differently shaped data sources. Anonymized data alone cannot ensure equal client distribution, and a centralized model may face difficulties in handling the non-independent and identically distributed (non-IID) nature of device data. As a result, this can lead to potential bias and reduced accuracy during model training. In this situation, clients may be unreliable and experience higher failure rates or dropouts due to their dependence on less robust communication media, such as Wi-Fi, and power-constrained systems, such as smartphones and IoT devices. The authors of [13] suggest a novel federated learning approach to address privacy, robustness, and model training, concerns when dealing with non-IID data. Still, they focus our work exclusively on the ML model aspects and do not consider the environment. For interested readers, [14] comprehensively analyzes various FL methodologies that apply to IoT systems. Additionally, reference [15] outlines potential research topics.

III. Problem definition: How can we minimize the variability in the results?

In the Internet of Things networks, achieving efficient malware detection while balancing key factors such as privacy, low computational overhead, and effective data exchange presents a complex challenge. The objective is to develop a robust malware detection method that provides high accuracy and maintains stability over time, with minimal variation in accuracy across different scenarios. This involves a state-of-the-art analysis to identify an optimal detection model and deploy it in a manner that balances the critical points mentioned above. The investigation analyzes a subset of

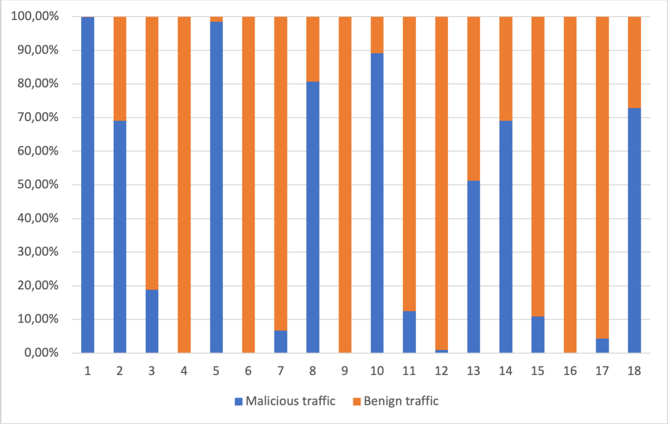

Fig. 2. Distribution of Malicious and Benign across the clients (horizontal axis).

overhead features from the IoT-23 dataset developed by Avast Software in Prague. The label distribution across the dataset is illustrated in Fig. 2, while a deep description is presented in the appendix. The challenges encountered included varying data set shapes, a significantly unbalanced label distribution, and a limited number of correlated features. A detailed data pre-processing investigation was also conducted. The objective of the research is to transform the limitations of IoT networks into strengths within a generalized model. Distributed approaches enhance scalability, whereas centralized models that exchange substantial data volumes between servers and clients can strain bandwidth. In federated models, only model parameters are shared with all clients, eliminating the necessity for each clients to transmit their own data and the relative parameters, thus reducing bandwidth usage and enhancing efficiency [16].

IV. Methodology and solution proposed

In this section, we present the pre-processing data methods and models. We briefly explain the standardization methods and the federated models used in the classical and modified versions. In conclusion, the evaluation methods are presented to describe the analysis in detail.

A. Pre-processing data approach

The pre-processing methodology became crucial in this work due to the variety of data and the high correlation between the features. For this reason, we introduce the core ideas considered and their implications. The following paragraphs present local and global standardization and distributed principal component analysis. These techniques address the high variance present in the data and make them homogeneous to simplify classification.

a) Local and global standardization: Standardization is a statistical technique that reshapes the data to make them more homogeneous and easy to classify. Given a client s with i examples in the dataset D_s, the local standardized data z_{si} is a transformation of the original data $d_{si} \in D_s$ as follows:

$$z_{si} = \frac{d_{si} - \mu_s}{\sigma_s} \tag{1}$$

where μ_s is the average vector of D_s and σ_s is the vector of standard deviation based on D_s. Otherwise, the global standardization replaces the μ_s and σ_s arrays with the μ and σ vectors, which are the mean and the standard deviation arrays calculated across all clients. These approaches offer valuable techniques for standardizing data, minimizing shared information, and enhancing model robustness.

The local standardization approach improves client confidentiality by modifying data within each client's domain. This method avoids transmitting sensitive information, such as the client's mean vector and variance, to the central server. Normalizing data locally helps address non-identically and independently distributed (non-IID) data across the network, making input data more uniform across clients. In contrast, global standardization involves estimating the overall data distribution on the server side. During the initial round of Federated Learning, the server aggregates mean and variance vectors from all participating clients, providing a comprehensive view of the network's data distribution. This global insight enables more robust model training by addressing disparities in data across clients. Both approaches offer distinct benefits: local standardization prioritizes privacy, while global standardization focuses on improving model performance by leveraging more comprehensive data distribution information. The choice between these methods depends on the federated learning application's specific needs and constraints.

b) Principal Component Analysis: The distributed version of Principal Component Analysis (PCA) [17] is presented. A classical PCA aims to find a smaller dimensional subspace that captures as much of the variance of the data as possible. The method involves a collaborative approach where the variance matrix is constructed using data from multiple clients or nodes in a distributed system. In this framework, each client processes and summarizes its local data, which is then exchanged with a central server or coordinator responsible for aggregating this information into a global variance matrix. Efficient computation of principal components necessitates the exchange of summarized indices or statistical measures between clients and the server. Nevertheless, this communication process may result in overhead due to data transmission and coordination. Despite the potential trade-off in increased communication complexity, the advantages of distributed PCA are consequential.

One advantage of PCA is its enhanced scalability to larger datasets that exceed the capacity of a single computing node or device. Distributing the computational load across multiple nodes reduces the overall processing time, leading to faster computation and analysis. Additionally, the distributed PCA model provides improved fault tolerance and resilience. Distributing data and computations across multiple nodes makes the system less susceptible to failures or disruptions in individual components. This redundancy can enhance the reliability and robustness of PCA computations in large-scale distributed environments. Furthermore, these distributed approaches also enable data analysis that preserves privacy. Clients can maintain control over their raw data while contributing only aggregated statistics or transformed representations to the central server. These strategies demonstrate proactive steps toward effective information management and collaborative model development within the context of FL. Each approach tackles unique challenges posed by distributed data settings, ultimately contributing to the advancement of secure and efficient machine learning practices.

B. Models

This section outlines our adopted models, introduces the base model, and describes our modifications. Specifically, the federated models discussed here are adaptations of Federated Averaging (FeAvg) and Federated Knowledge Distillation (FD). For a more detailed explanation of these models, please refer to [18], [19] and [20].

a) Classical version: The federated models utilize an iterative approach that entails discrete interactions between clients, who possess a local model, and the server, which establishes the global model. The server chooses a subset of clients to train the local model in each iteration and updates the global model based on the parameters received. The logit function is the main distinguishing factor between the two approaches.

For the FedAvg algorithm is shown in Fig. 3, while the loss function of a generic s device is given by:

$$L_s(D_s, \mathbf{w}) = \frac{1}{N_s}\sum_{i=1}^{N_s} l(d_{si}; \mathbf{w}) \tag{2}$$

where D_s corresponds to the dataset of device s, N_s corresponds to the amount of examples list in D_s, l is an entropy function such as cross-entropy function and d_{si} is the i-th example of D_s. It is important to note that the weight $\boldsymbol{w}$ is usually a simple averaging of the selected clients' weights.

On the other hand, Federated Knowledge Distillation involves a more complex loss function and requires selected clients to store the mean logit vector per label during local training. A simplification of the algorithm is shown in Fig. 4. The logit is a combination of the previous logit and the distance between the client and server logit arrays, as defined by the formula:

$$L_{KD}(D_s, \mathbf{w}) = L_s(D_s, \mathbf{w}) + \lambda \cdot KL_{dist,y}(Y_{log}, Y_{log,s}) \tag{3}$$

where $L_s(D_s, \mathbf{w})$ was defined previously and $KL_{dist,y}(Y_{log}, Y_{log,s})$ is the Kullback-Leibler metrics distance used to compare the logit distribution of the client and the server. The pre-trained phase, typical of this method, was excluded from the experiment to create a lightweight and fast tool.

b) Our modified version: The difference from the usual methods is that the update of the local weights of device s at step $t+1$ is calculated using the following formula:

$$\mathbf{w}_{s,t+1} = \beta \cdot \mathbf{w}_{s,t-1} + (1-\beta) \cdot \mathbf{w}_t \tag{4}$$

where $\mathbf{w}_{s,t-1}$ is the weights vector of the device s at step $t-1$, and $\mathbf{w}_t$ is the average weights vector of the selected device at step t. β is a hyperparameter that can take values from 0 to 1. In our experiment, we used $\beta = 0.75$. This approach guarantees stable weights and robust model performance, enabling the model to learn effectively without being influenced by variations in scale, such as alternating between high-value and low-value clients across training rounds.

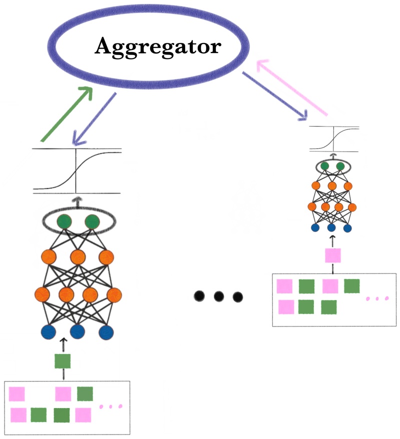

Fig. 3. Representation of a Federated Average method. The blue arrows communicate the average weights vector, while the other arrows communicate the client weights vectors. The boxes indicate the devices.

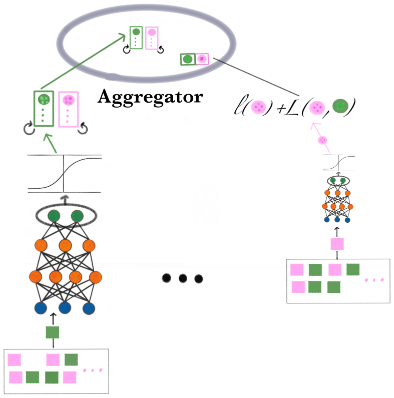

Fig. 4. Representation of a Federated Distillation method. The circle represents the different logit vectors: the full-colored ones are the global logit, the circle with the x in the center is the contribution for the regularization, and the others are the client logits. The boxes indicate the devices.

Our methods assume that the system is stationary, meaning that the statistical properties of the data, such as the mean and variance, remain constant over time. In a stationary system, the data distribution does not change significantly, allowing the model to rely on consistent patterns. However, these estimates may become inaccurate if the global mean and variance shift over time. To address potential non-stationarity, we trigger an index update when there is a significant change in average loss. This mechanism helps maintain the accuracy of the estimates and allows for dynamic adjustments. Specifically, the update is initiated when the loss change exceeds a threshold value of 0.1 ($\textit{threshold} > 0.1$). As previously highlighted, minimizing information exchange and efficiently preprocessing data are crucial for enabling rapid analysis and easy deployment in practical scenarios.

C. Evaluation approach

To facilitate the comparison of results, the area under the precision-recall curve (AUPRC) is chosen as an index of fitness. The AUPRC is a widely recognized machine learning metric for evaluating classification algorithms. Consider the unbalanced nature of the data and provide a suitable summary of the fitness of the model on the data [21]. This index helps to identify the balance between precision and recall at various thresholds. Precision is the ratio of correctly classified examples to all positively classified examples, while recall is the ratio of correctly classified positive examples to all positive examples.

When comparing different models, their AUPRC scores will be compared by ratio. For instance, let's consider a generic Federated Model (AUPRC(Fed)) and a Centralized one (AUPRC(Centr)), the ratio of their AUPRC scores are calculated as follows:

$$\text{AUPRC}_{\text{Ratio}} = \frac{\text{AUPRC(Fed)}}{\text{AUPRC(Centr)}} \tag{5}$$

A ratio greater than 1 indicates that the Federated Model outperforms the Centralized one regarding AUPRC, while a ratio less than 1 indicates the opposite.

In conclusion, a Chi-square test can be performed to verify the variance reduction and evaluate whether the variances of two populations are equal. By applying this test, we can establish whether the variance reduction is meaningful and quantify the effectiveness of the proposed methods. This evaluation offers robust metrics that assure improvement.

V. Evaluation

Our experiments want to highlight how the federated models overcome the centralized model, which is used as a baseline. That happens because the proposed model is more suitable for the nature of the IoT environment and allows some key concepts. Also, the exchange data can be manipulated depending on bandwidth availability.

A summary of the results can be found in the box plots presented in Figure 5. These box plots illustrate the distributions of the AUPRC values for each data configuration and operation method. The PCA and standardized transformation results were aggregated for the centralized approach as they produced identical outcomes. In contrast, the federated approach without standardized data led to the presentation of both models' outcomes. Upon examination of the figure, it becomes evident that the federated model with globally standardized or PCA data exhibits minimal variance and demonstrates superior global AUPRC performance. A chi-square test is performed to verify this reduction in variance, and the results are reported in Table I. The test results demonstrate that the Federated Learning approach significantly reduces variance in the Distributed PCA AUPR index and the Global Standardized AUPR index compared to the local Standardized Data Index, with a significance level of 0.05.

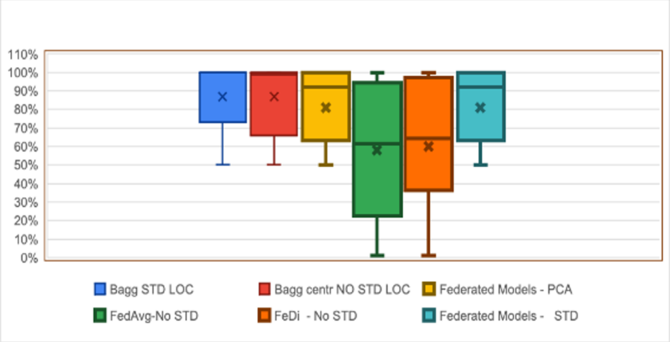

Fig. 5. Box plots of the AUPRC of the clients. The cross in each box represents the mean. As we can see, the low variance is achieved by the Federated models.

TABLE I
Summary of the average AUPRC ratio between Federated models (FedAvg and FD) and Centralized models

Model	Federated No STD	Federated Global STD	Federated PCA
Centralised No STD	0.94 (FedAvg) 1.07 (FD)	1.64	1.65
Centralised Data Transformation	0.82 (FedAvg) 0.97 (FD)	4.9	4.91

As mentioned above, analyzing model performances based on their AUPRC ratios provides valuable insight into different configurations. In the case of using only transformed data, FL methods exhibit significantly higher performance, approximately three times on average (with $\text{AUPRC}_{Ratio} \simeq 4.9$), whether employing PCA-transformed data or standardized data. This outcome illustrates these methods' efficacy in reducing the variance across the results obtained. It can be seen that FedAvg demonstrates superior performance in this regard, as it minimizes the information exchange between nodes. In comparison to a centralized model, excluding standardized data during federated learning resulted in an average performance improvement of approximately 7% ($\text{AUPRC}_{Ratio} = 1.07$). Despite this improvement, the significant variance observed suggests that while this federated model shows promise, it may not offer the most consistent results.

Further comparison between standardized data and transformed data within the FL framework revealed a substantial performance gain, with an average improvement of around 60% compared to traditional centralized methods

TABLE II
Result of Chi test on AUPRC index performed on the client AUPRC distribution

Chi test	p-value
PCA data	0.04
No STD data - FD	0.99
No STD data - FedAvg	0.99
Glob STD data	0.04

($\text{AUPRC}_{\text{Ratio}} \simeq 1.6$). This highlights the critical importance of federated participation and data transformation in optimizing model outcomes.

A Chi-square test was conducted to validate these findings, confirming significant differences in the mean AUPRC distributions between standardized and transformed data (p-values between 0.1 and 0.15); see Table II. Additionally, data exchanged between nodes and the aggregator was analyzed. The results show that overhead traffic is directly influenced by the volume of data exchanged. Among the exchanged data types, model weights had the smallest size, averaging 224 bytes. In contrast, centralized models require the transmission of the entire device dataset, resulting in a significant overhead burden. However, the exact amount varies depending on device traffic and is not specified here. Other data transmitted during FL include feature sum and deviation vectors, each 56 bytes in size, which are essential for global standardization and principal component analysis (PCA). Basic traffic information is also communicated during the setup phase. Finally, the federated dropout (FD) algorithm requires additional data transfer, including the labeled logit vector (8 bytes) and model weights, further increasing the data exchange requirements.

The average data size for each client is around 140 MB. The average GPU utilization per example is 3.51 for centralized models and 2.15 for Federated approaches. The execution time for 1 MB is 5 seconds on average for the centralized model and 4.83 seconds on average for the Federated models. These values indicate that higher computational effort is required for validation on individual devices, but this enables a more distributed workload across the network, as previously demonstrated. Additionally, as depicted in Figure 6, when the example size increases, device GPU usage is lower for the Federated models than the centralized model, as previously mentioned.

VI. Future challenges

The following section outlines future research goals and presents emerging challenges in network security. As network threats continue to evolve, it is imperative to develop methods for accurately classifying raw data, understanding the inner workings of these methods, and addressing critical security issues. A novel approach that combines computer vision techniques with Explainable AI (XAI) methods, such as saliency maps, offers promising advancements in this domain.

By converting raw data into visual representations, such as grayscale images constructed from hexadecimal arrays [22], and applying saliency maps to highlight key features [23], this method enhances the precision of malicious activity detection while clarifying the rationale behind machine learning decisions. Despite these advancements, research on integrating XAI and computer vision for cybersecurity analysis remains limited, presenting a significant research opportunity. Future work should investigate the effectiveness of different XAI methodologies combined with computer vision techniques to improve network anomaly detection and interpretability. This approach holds great potential for enhancing malware detection and security awareness as explainable AI develops.

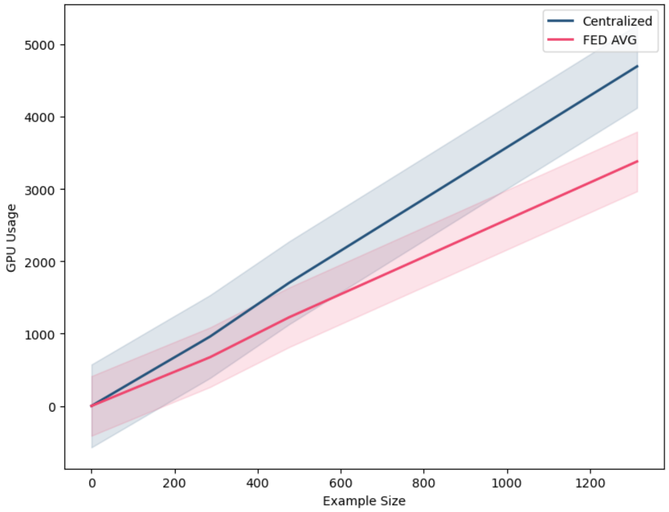

Fig. 6. The graph shows the GPU usage curve based on the test set size (number of examples). The blue curve represents the centralized model, while the pink curve represents Federated models. The pink curve demonstrates the lower GPU usage of the Federated models compared to the centralized model. The bars indicate the confidence interval at level 95%.

In addition to these advancements, our future research aims to expand the current framework to include multinomial cases, which we believe will significantly improve the model's diagnostic capabilities. This expansion will identify a broader range of malware categories, further strengthening cybersecurity defenses. We also plan to explore how federated learning can benefit from scenarios where only one or a small subset of clients encounters new malware. This research will focus on collaborative efforts within the federation to enhance security and learning outcomes, particularly through data transformation techniques that leverage computer vision approaches.

Finally, an important aspect not yet addressed in our framework is the *forgetting problem*, where models must continuously adapt to evolving data distributions and emerging patterns. We will investigate mechanisms to prevent, mitigate, or strategically use forgetting in machine learning models. This exploration aims to improve the model's resilience and versatility, enhancing its efficacy across various real-world applications.

VII. Conclusion

In conclusion, the IoT ecosystem, characterized by a vast network of interconnected devices that collect, exchange, and analyze data, is inherently vulnerable to malware attacks due to its simplicity and limited security features. To address this challenge, a federated approach for binary classification has been investigated, leveraging the decentralized nature of IoT devices and their computational capabilities to optimize learning while ensuring data security.

This study performed a comprehensive analysis using the IoT-23 dataset, which includes labeled instances of IoT malware infections. The results demonstrate that federated models, particularly those employing global standardization or principal component analysis (PCA), outperform traditional centralized approaches in the global area under the precision-recall curve (AUPRC) and exhibit lower variance.

The Federated Average approach, trained on globally standardized data, emerges as the most effective among the methods evaluated. It achieves a crucial balance between enhancing model performance and minimizing overhead traffic, making it a highly promising solution for federated learning applications in IoT environments. This approach improves security and supports efficient and scalable deployment across diverse IoT networks.

Acknowledgement

This work has been partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, a partnership on "Telecommunications of the Future" (PE0000001 - program "RESTART").

References

[1] K. K. Patel, S. M. Patel, and P. Scholar, "Internet of things-iot: definition, characteristics, architecture, enabling technologies, application & future challenges," *International journal of engineering science and computing*, vol. 6, no. 5, 2016.

[2] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Service composition approaches in iot: A systematic review," *Journal of Network and Computer Applications*, vol. 120, pp. 61–77, 2018.

[3] M. K. Hasan, A. A. Habib, Z. Shukur, F. Ibrahim, S. Islam, and M. A. Razzaque, "Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations," *Journal of Network and Computer Applications*, vol. 209, p. 103540, 2023.

[4] A. A. Elngar, R. Chowdhury, M. Elhoseny, and V. E. Balas, *Applications of Computational Intelligence in Multi-Disciplinary Research*. Academic Press, 2022.

[5] F. Deldar and M. Abadi, "Deep learning for zero-day malware detection and classification: A survey," *ACM Computing Surveys*, 2023.

[6] N. Pachhala, S. Jothilakshmi, and B. P. Battula, "A comprehensive survey on identification of malware types and malware classification using machine learning techniques," in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*. IEEE, 2021, pp. 1207–1214.

[7] A. A. Almazroi and N. Ayub, "Deep learning hybridization for improved malware detection in smart internet of things," *Scientific Reports*, vol. 14, no. 1, p. 7838, 2024.

[8] A. Gaurav, B. B. Gupta, and P. K. Panigrahi, "A comprehensive survey on machine learning approaches for malware detection in iot-based enterprise information system," *Enterprise Information Systems*, vol. 17, no. 3, p. 2023764, 2023.

[9] Ö. A. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE access*, vol. 8, pp. 6249–6271, 2020.

[10] A. Nascita, F. Cerasuolo, D. Di Monda, J. T. A. Garcia, A. Montieri, and A. Pescapè, "Machine and deep learning approaches for iot attack classification," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2022, pp. 1–6.

[11] J. Jeon, J. H. Park, and Y.-S. Jeong, "Dynamic analysis for iot malware detection with convolution neural network model," *IEEE Access*, vol. 8, pp. 96 899–96 911, 2020.

[12] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in iot devices," *Computer Networks*, vol. 204, p. 108693, 2022.

[13] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "Fed-iiot: A robust federated malware detection architecture in industrial iot," *IEEE transactions on industrial informatics*, vol. 17, no. 12, pp. 8442–8452, 2020.

[14] Q.-V. Pham, K. Dev, P. K. R. Maddikunta, T. R. Gadekallu, T. Huynh-The *et al.*, "Fusion of federated learning and industrial internet of things: A survey," *arXiv preprint arXiv:2101.00798*, 2021.

[15] M. Venkatasubramanian, A. H. Lashkari, and S. Hakak, "Iot malware analysis using federated learning: A comprehensive survey," *IEEE Access*, vol. 11, pp. 5004–5018, 2023.

[16] J. S.-P. Díaz and Á. L. García, "Study of the performance and scalability of federated learning for medical imaging with intermittent clients," *Neurocomputing*, vol. 518, pp. 142–154, 2023.

[17] E. Oja, H. Ogawa, and J. Wangviwattana, "Principal component analysis by homogeneous neural networks, part¡ cd02d35. gif¿: The weighted subspace criterion," *IEICE Transactions on Information and Systems*, vol. 75, no. 3, pp. 366–375, 1992.

[18] M. Venkatasubramanian, A. H. Lashkari, and S. Hakak, "Iot malware analysis using federated learning: A comprehensive survey," *IEEE Access*, 2023.

[19] H. Seo, J. Park, S. Oh, M. Bennis, and S.-L. Kim, "16 federated knowledge distillation," *Machine Learning and Wireless Communications*, p. 457, 2022.

[20] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-based federated learning for securing internet of things: A comprehensive survey," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–43, 2023.

[21] H. R. Sofaer, J. A. Hoeting, and C. S. Jarnevich, "The area under the precision-recall curve as a performance metric for rare binary events," *Methods in Ecology and Evolution*, vol. 10, no. 4, pp. 565–577, 2019.

[22] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[23] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

Appendix

The IoT-23 dataset, collected by Avast Software, Prague, consists of twenty-three captures of IoT network traffic from 2018 to 2019. Each scenario involves executing specific malware samples on a Raspberry Pi using various protocols. The dataset includes original .pcap files, corresponding Zeek *conn.log* files, and*conn.log.labelled* files with additional labeling columns. Due to large traffic volumes, .pcap files are rotated every 24 hours, sometimes resulting in varying capture durations.

These labels are one for the binary classes (Malicious, Benign), the other for the category of malware. In this work, only the binary labels are considered, and only the following features are considered:

- *duration*: How long the connection lasted
- *origin bytes*: The number of payload bytes sent by the originator. This is taken from sequence numbers for TCP and may be inaccurate (e.g., due to large connections).
- *missed bytes*: Indicates the number of bytes missed in content gaps, representing packet loss. A value other than zero will normally cause protocol analysis to fail but some analysis may have been completed before the packet loss.
- *originator packets*: Number of packets that the originator sent.
- *originator IP bytes*: Number of IP level bytes sent by the originator
- *responder packets*: Number of packets sent by the responder
- *responder IP bytes*: Number of IP level bytes sent by the responder (as seen on the wire, taken from the IP *total length* header field).

Of course, because of the nature of the experiment, dataset sizes vary significantly, with smaller sets comprising approximately four examples while larger sets contain over 100,000 samples. For this reason, the datasets are considered if they have at least 100 examples.