Department: Head

Semantic-based Federated Defense for Distributed Malicious Attacks

Farhan Ullah
University of Camerino

Leonardo mostarda
University of Perugia

Diletta Cacciagrano
University of Camerino

Chien-Ming Chen
Nanjing University of Information Science and Technology

Saru Kumari
Chaudhary Charan Singh University

***Abstract*—Consumer electronics, particularly Android, has become the leading mobile ecosystem due to its accessibility and adaptability. However, the constant connectivity of Android apps makes them a target for malicious network attacks, leading to the potential theft of sensitive data and disruptions across various sectors, including the economy and healthcare. Developing a reliable, distributed malware detection system using training data from multiple sources is challenging. This is mostly due to privacy issues and a lack of consistent data. This paper proposes a semantic-driven Federated Learning (FL) approach using transformer-based transfer learning to defend against distributed malicious attacks. The semantic features of malicious scripts are examined using the Bidirectional Encoder Representations from the Transformers (BERT) model. Following that, Deep Neural Network (DNN) uses these semantic features for local training, resulting in local model updates for each client. After merging the local model updates from each client, the global server generates global weights and sends them to distant clients. The proposed approach is evaluated on two standard datasets, including CIC-AndMal2017 and CICMalDroid2020, and it obtains high detection accuracy of 99.38% and 99.14%, respectively. These findings encourage cybersecurity organizations to collaborate and develop a powerful distributed security system using private data.**

CONSUMER TECHNOLOGY has rapidly progressed and become an integral part of our daily life. Consumer electronics are increasingly susceptible to malicious or financial cyberattacks utilizing Android apps. The

Digital Object Identifier 10.1109/MCE.YYYY.Doi Number

Date of publication DD MM YYYY; date of current version DD MM YYYY

widespread use of Android has attracted the attention of hackers, who use the operating system to launch continuous malware attacks. Android is a popular operating system for consumer devices, making it a prime target for hackers. The expanding threat landscape highlights the significance of effective cybersecurity solutions in consumer electronics. Integrating modern FL technologies for malware detection and prevention is an important priority for protecting user data in consumer electronics [1]. Semantic-based FL is crucial for accurately analyzing the content of malicious network traffic and detecting Android malware in consumer technology. This method enhances accuracy while protecting data privacy across various consumer devices.

Deep Learning (DL) has developed as a valuable and widely acknowledged method for decreasing malware threats. It uses large amounts of user data to detect malware attack patterns and trends in a centralized manner. The typical procedure involves sending applications to a cloud server for detection before installation when employing this centralized approach. This technique increases network overhead and costs, especially for large apps, and requires the unsecured collection and transmission of sensitive data. One possible solution is to enable distributed learning, which processes sensitive user data locally rather than transferring private mobile data to a cloud server. FL is a method for enhancing data security and privacy by keeping it locally on devices. The objective is to use privacy-preserving technologies to combine model parameters from several consumer clients to create a globally optimum model without sharing consumer data. FL allows local models to achieve detection abilities similar to centralized learning. It also effectively mitigates any leakage risks by transmitting sensitive data during the procedure [2] – [3].

Identifying malicious network activity involves the recognition of abnormal patterns, such as unusual traffic quantities and repeated small packets indicating scanning. Furthermore, major warning flags include several failed login attempts and payload data that mimics known malware. Network traffic-based techniques aim to identify distinguishing features that may be utilized to characterize harmful apps accurately [4] – [5]. Several malicious scripts may be used by network malware to damage an Android app. Semantic-based feature analysis can uncover potentially hazardous scripts for behavior segmentation. Some malicious software connects directly to an IP address without first attempting to resolve the address. This is also commonly interpreted as a malicious warning indicator. SMS malicious software constantly sends malicious code to the server, demanding authorization and payment. These actions can be investigated by extracting semantic information from network traffic. We used semantic-based criteria to identify possibly malicious scripts that perform harmful acts like storage or resource utilization

Machine Learning (ML) detects and combats malware assaults by recognizing key attack behaviors and trends from massive amounts of user data, as shown in Figure 1. However, this strategy has drawbacks, most notably data security and integrity. For instance, clients' need to provide data to the central server creates a vulnerability that malevolent users can exploit. Low latency, enough resources, and dependable communication are necessary for real-time operations. One way is to train the DL model using the initial client data and then distribute the trained model among clients. This decreases the requirement for continuous data transfers while boosting prediction accuracy for specific clients, reducing network reliance and transportation costs. However, data must be transferred to the central server during the initial training phase, which raises security and potential data privacy concerns. FL supports distributed learning by allowing clients to view sensitive data locally rather than sending it to a central server. This technique protects data privacy and prevents distributing harmful scripts to other clients, reducing the risk of data breaches and attacks. Its primary goal is to decentralize data analysis and keep user data from being routed to centralized servers. Effective cybersecurity solutions require detailed awareness of risks such as spoofing, infiltration, anomalies, and Denial of Service (DoS).

Our Contributions

FL offers an alternative approach to distributed learning by utilizing local user data instead of transferring it to a cloud server. Decentralizing data analysis maintains user data on the network and prevents malicious scripts from spreading, which improves cybersecurity. This paper uses a semantic-driven FL strategy that employs transformer-based transfer learning to prevent distributed malicious attacks. The global server generates client-specific model updates and merges them for increased security by assessing semantic features using a transformer. The global server then sends aggregated model updates to each client, refreshing the

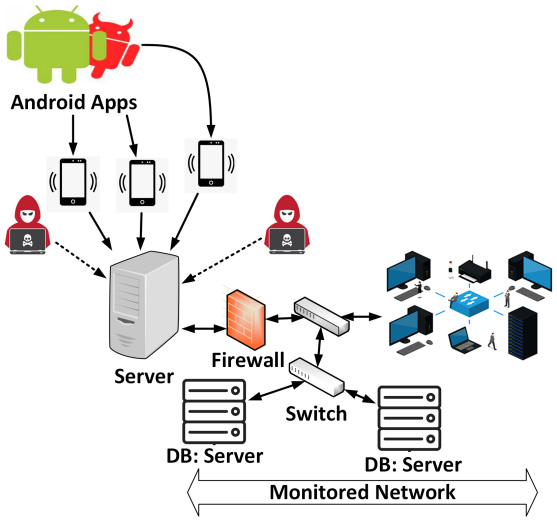

Figure 1. Conventional ML models with cyberattacks

local updates. Semantic-based features aim to identify harmful traffic while protecting the client's security and privacy through FL.

The paper is organized in the following sections: Part 2 explains the related research, and Part 3 describes the proposed approach. Experiments are summarized in Part 4 and concluded in Part 5.

MALICIOUS ATTACKS: RELATED WORK

This section examines the current state-of-the-art malicious attack methodologies and several FL frameworks.

ML and DL techniques

Consumer electronics devices frequently incorporate automatic registration capabilities due to the difficulties of operating compact interfaces. However, it may threaten user privacy, as the stored registration data is susceptible to identity scams and theft. Cho et al. [1] methodically illustrate this threat using two real-world Android app case studies (Starbucks and MOCI) and a prototype. It also investigates five defense techniques to reduce the probability of credential data theft. Consumer Mobile banking employs mobile devices to execute online banking tasks, including transferring funds, monitoring account balances, and paying expenses. Although it is convenient, it is susceptible to a variety of attacks. Wazid et al. [6] investigate the rapid development of mobile banking, including associated threats and current malware attacks, as well as new security measures to improve its security.

Gao et al. [7] develop apps that utilize the API edges by invocation relationships and usage patterns. The graph is loaded into a network to generate embedding for app classification. GDroid uses graph neural networks to classify malware. GDroid identifies 98.99% of Android malware, with less than 1% of false positives. Zhang et al. [8] demonstrated an efficient search strategy for neural architectures in Multi-Target Classification (MTC). It analyzes real-world Android network traffic to determine appropriate model topologies. The problem is a constrained optimization challenge with a differentiable search space and discrete design. The strategy assisted MTC in identifying suitable classification models using USTC-TFC2016 data. An edge server hybrid monitoring system was designed by Sedjelmaci et al. [9]. This system efficiently detects malicious devices by combining signature-based and neural network-based methodologies. In addition to mitigating zero-day threats, the approach consistently identifies malicious actors who exploit wireless communication to inject fabricated data. The experiments comprised various network threats, including DoS and fuzzing attacks.

Federated security solutions

The development of cybersecurity protection frameworks based on FL algorithms has recently evolved [10]. These frameworks manage compute overhead and data privacy concerns throughout the defensive process, from training and monitoring to detection and reaction. Hichem et al. [11] present recent advances in MEC attack detection and defense systems that employ FL and Generative Adversarial Network (GAN) technologies. Additionally, they provide the FedGAN technique and a non-cooperative game methodology to enhance attack detection accuracy in a cyber-defense architecture. Bakopoulou et al. [12] propose an FL approach to mobile packet classification, enabling cooperative global model training without data exchange. This method performs well on real-world datasets regarding transmission costs, computation performance, and classification accuracy. For packet categorization apps, the problems addressed are model selection, feature selection, and federated learning parameter adjustment. Using non-identifiable data, Xiong et al. [13] examined privacy protection in FL scenarios. Their suggested solution, 2DP-FL, protects differential privacy by deliberately introducing noise into both the local and global distribution of models. Experimental results suggest that 2DP-FL

protects privacy, accelerates learning, and increases model efficiency. The FLAME protection paradigm, which Nguyen et al. [14] proposed, aims to ascertain the optimal noise levels that can be utilized to block backdoors efficiently. Model clustering and weight extraction approaches reduce the noise, ensuring trustworthy performance by eliminating potential malicious vulnerabilities from the combined model.

We propose an innovative semantic-driven FL technique that employs transfer learning with a multi-head attention transformer. Before employing a DNN for local training and client model updates, this method uses a transformer to analyze the semantic aspects of harmful scripts. The federated server aggregates these updates, enhancing privacy and security and optimizing each client's FL process. Our method is tested on two benchmark datasets: CIC-AndMal2017 and CICMalDroid2020.

SEMANTIC-FEDERATED DEFENSE FOR DISTRIBUTE ATTACKS

The proposed approach extracts sentiment indicators from malicious Android network data using a semantic-based FL technique to offer protection against malware attacks. This methodology combines the efficacy of network semantic features with the privacy-preserving FL method to create an effective malware detection solution. The pre-trained transformer model extracts salient information from network traffic, reducing the workload on the FL and focusing more on harmful scripts. Attackers may employ a range of Android apps because the clients are connected remotely, which could lead to chaotic behavior as shown in Figure 2).

Network Traffic Analysis

Android app network traffic analysis extracts HTTP/HTTPS requests, TCP flows, API endpoints, network performance indicators, and data use trends. This Android network traffic analysis details app behavior, security, performance, privacy threats, and data compliance concerns. By examining PCAP files, Wireshark is an essential tool for analyzing Android network traffic from Android apps. It allows for extracting data usage patterns, network performance indicators, API endpoints, and HTTP/HTTPS searches. Developers may optimize and strengthen network communication tactics using Wireshark and packet sniffing to understand app behavior, security vulnerabilities, and effectiveness.

HTTP traffic is widely used due to its popularity in worldwide interaction. The information in HTTP headers is useful for detecting malicious attempts. Mobile apps use encoded HTTP to communicate essential data securely. TCP flows contain information on data transit, received data, and total session count. HTTP traces provide important information, including source/destination, host, port, bytes, frame/packet lengths, and TTLs. This contains URLs (e.g. "www.yahoo.com"), GET, and POST queries. Filtering is crucial for maintaining semantic relevance. Short sequences are removed to avoid obscuring critical network activity and redundant features. Standardizing sequence length is critical for effective malware analysis because variances can lead to false classifications. We balance sequence lengths using a specified L and zero-padding to equalize shorter sequences while retaining their original names. First, remove any non-essential data, such as irrelevant network activity, to preprocess network traffic data efficiently. Next, we extract important information from HTTP flows and TCP streams, such as source and destination addresses, request types (GET/POST), and data transfer statistics. Data is standardized using padding techniques and sequence lengths to ensure sample consistency. We undertake quality tests to eliminate residual anomalies and discrepancies, resulting in clean and relevant network data.

Multi-head attention of transformer: Semantic Features Extraction

BERT employs transformers, a deep learning model in which the attention mechanism dynamically computes the correlations between input and output attributes. A bidirectional paradigm is employed to identify the left and right contexts of terms, allowing the information to be evaluated holistically instead of chronologically. Unlike non-contextual models, the transformer produces several interpretations for connected terms, which provide a single-word description regardless of context. The multi-head transformer improves capabilities by concurrently focusing attention on various parts of the incoming data. This concurrent processing enables the model to collect several abstractions and comprehend complex relationships in the dataset. A multi-head transformer is necessary for the semantic analysis of malicious scripts. Malicious scripts frequently exhibit complex, context-dependent behaviors that are difficult to identify with conventional methods. The multi-head attention technique

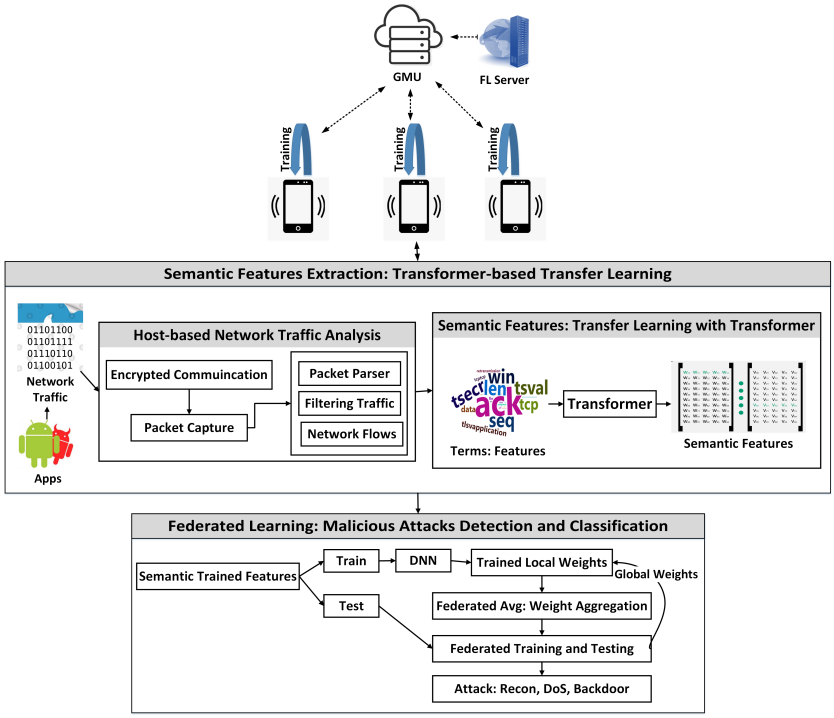

Figure 2. Semanti-based federated defense for distributed attacks in consumer electronics

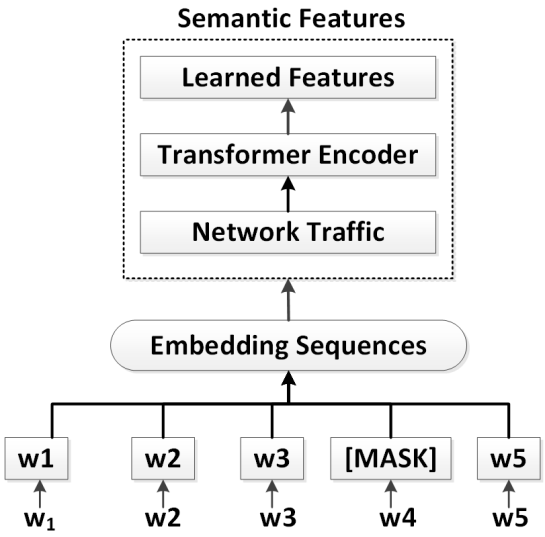

Figure 3. Semantic Features Mapping

examines a script's semantic structure, including syntactic and semantic relationships across network data.

We use a transformer to extract training features from network data, as shown in Figure 3. This procedure entails iteratively traversing embedded network features (such as w1, w2, etc.), which allows the BERT-large neural network to generate H-dimensional vectors that connect to these features. A [MASK] token is used to change 15% of the features in each sequence before processing by the transformer. The model then predicts the original values based on the context of the non-masked features. This method helps the model understand the relationships in Android network traffic data. It consequently results in more precise network patterns and anomaly detection and evaluation. Furthermore, the masking method aids the model in learning to predict missing or obscured data. This functionality is crucial for network security and traffic optimization. We used the BERT Large model, a more complex version of the transformer design with 24 layers and 340 million parameters. While it often improves performance, it requires more processing resources. Our strategy involves adding a classification layer to the encoder stream to help improve model results and refine predictions. This additional layer refines the model's expertise and decision-making capabilities, resulting in more nuanced and accurate results in various applications. The following steps are used:

- The encoder output is supplemented with a classification layer.
- Through multiplication with the embedding matrix,

output vectors are converted to the lexical dimension.

- Each feature in the vocabulary is assigned a probability using Softmax.

Sporadic Consumer Clients

This work investigates the issues provided by sporadic behavior in federated clients and suggests techniques to improve malware detection against distributed attacks. The irregularity of remote client interactions can be attributed to various factors, including either the arrival or departure of new clients [7] and [12]. This may lead to the development of imbalanced datasets and a reduction in model accuracy. We suggest two approaches to address this issue. one way is to aggregate the weights received for new clients from the current and following iterations while discarding the weights from previous training cycles when a client stops participating. Alternatively, the most recent client weights may be kept while the model is updated by aggregating current client weights. We emphasize the importance of considering client data volumes and preferences while dealing with the complexities of choosing alternative approaches. A transformer-based DNN is employed to perform local training by utilizing local model updates. Global model updates effectively manage data aggregation and FL training, thereby preserving confidentiality and preventing distributed attacks.

This work aims to improve model weights, handle client-to-client data conflicts, and employ federated weighted averaging techniques to improve malware classification performance against distributed attacks.

EXPERIMENTATION AND RESULTS

Datasets

The proposed work is examined using two standard datasets, such as CIC-AndMal2017 [14]: https://www.unb.ca/cic/datasets/andmal2017.html, and CICMalDroid2020 [15]: https://www.unb.ca/cic/datasets/maldroid-2020.html.

Performance Evaluation

We used semantic-FL to study sporadic behavior with different numbers of clients (2, 7, 10, and 15). We discovered that using fewer clients led to lower results. However, performance is substantially enhanced by increasing the number of clients. Figures 4 and 5 depict the local and global accuracy across epochs using the CIC-AAGM2017 dataset with 2 and 7 clients, respectively. We found that adding more clients to share the task slightly improved accuracy. The efficiency of our method is shown by the inverse relationship between accuracy and loss, which is illustrated in Figure 6 by showing the local and global loss values for all clients. Figures 6 and 7 also present the local and global epoch curves for accuracy and loss values using the CIC-AAGM2017 dataset. Moreover, Figures 8 – 11 show the dynamic local and global accuracy and loss values according to the epochs for clients 10 and 15, respectively. Table 1 shows performance comparisons using the CIC-AAGM2017 dataset for sporadic clients. Similarly, Tables 2 and 3 show the performance comparison using the CICMalDroid2020 and CIC-AAGM2017 datasets. Our results indicate that our approach achieves the best malware detection performance as the number of clients increases. For instance, with the CIC-AAGM2017 dataset, we achieved precision, recall, f-measure, and accuracy rates of 99.38 %, 98.62 %, 98.19 % for malware, and 98.94 %, 99.12%, 98.28 %, and 98.96 % for benign samples, respectively. More clients improve generalization and reduce bias by absorbing more diverse and heterogeneous data, resulting in higher model accuracy. A larger and more diverse dataset compiled from several clients enhances feature extraction and robustness. This aggregation also has a regularizing effect, which reduces noise within individual datasets. However, efficient communication management is necessary to maximize these benefits. Increasing the number of clients boosts model accuracy, flexibility, and robustness.

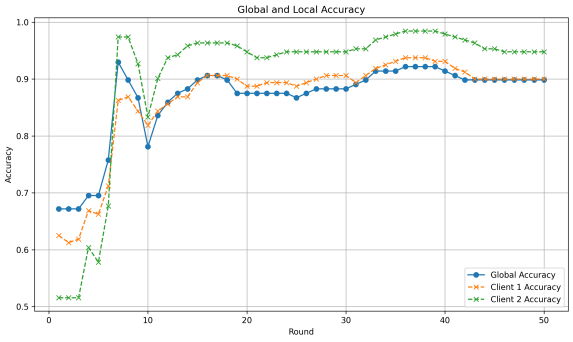

Figure 4. Local and Global accuracy for 2 clients

CONCLUSION

Android apps are vulnerable to suspicious network attacks due to their ongoing connectivity, increasing the risk of data theft. Inconsistent data and privacy concerns impede the development of a reliable distributed

This article has been accepted for publication in IEEE Consumer Electronics Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MCE.2024.3431792

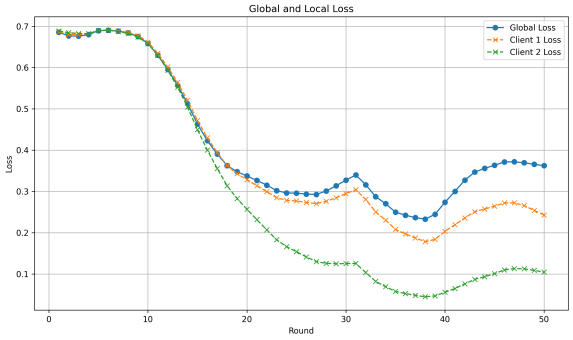

Figure 5. Local and global loss for 2 clients

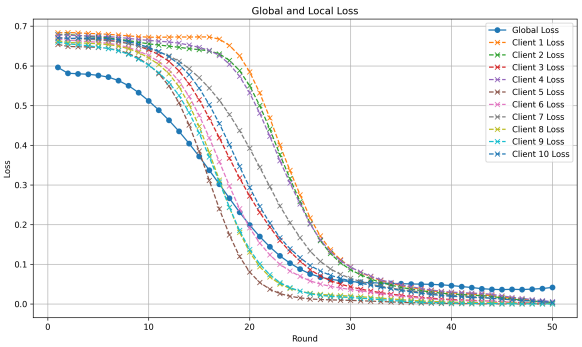

Figure 9. Local and global loss for 10 clients

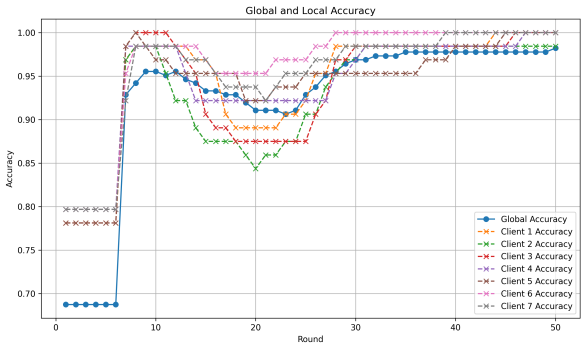

Figure 6. Local and global accuracy for 7 clients

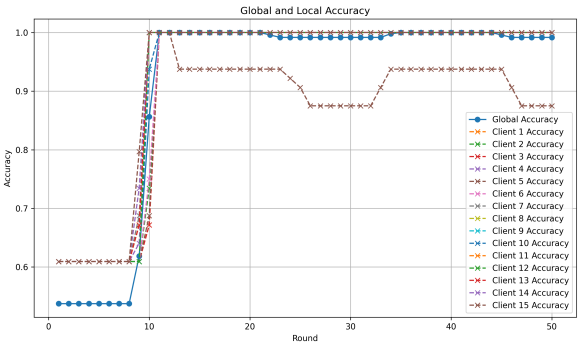

Figure 10. Local and global accuracy for 15 clients

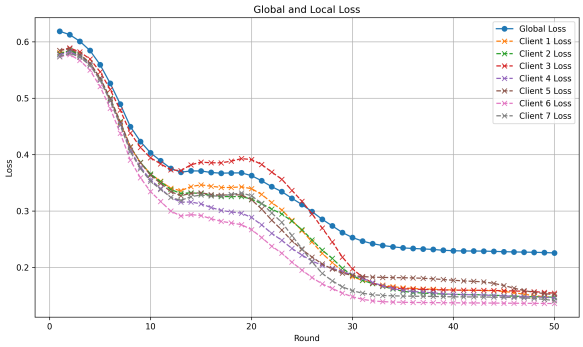

Figure 7. Local and global loss for 7 clients

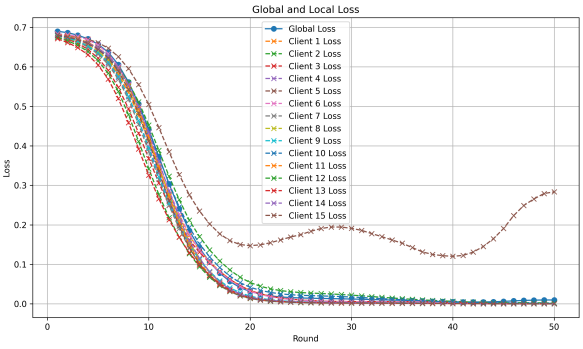

Figure 11. Local and global loss for 15 clients

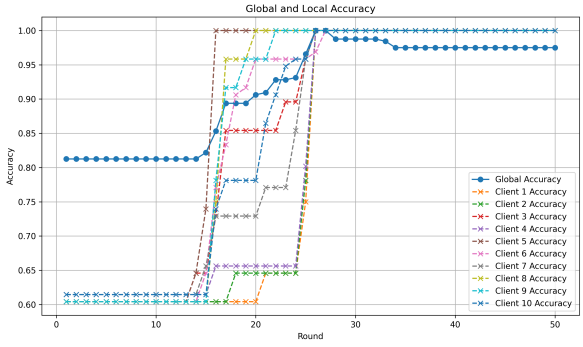

Figure 8. Local and global accuracy for 10 clients

Table 1. Performance Comparison using CIC-AAGM2017

Class	Clients	Precision	Recall	F-measure	Accuracy
Benign	2	93.21	93.82	92.96	93.62
Malware		94.34	93.81	93.02	
Benign	7	99.38	98.62	98.19	98.96
Malware		98.94	99.12	98.28	
Benign	10	99.82	99.10	98.74	99.11
Malware		98.91	98.82	99.08	
Benign	15	100	99.02	99.28	99.38
Malware		99.20	99.21	99.32	

Table 2. Performance Comparison using CICMalDroid2020

Class	Clients	Precision	Recall	F-measure	Accuracy
Benign	2	92.96	92.45	92.84	93.89
Malware		93.29	93.23	93.92	
Benign	7	96.48	96.69	96.29	98.12
Malware		98.72	98.67	98.61	
Benign	10	98.33	98.78	98.53	98.62
Malware		98.60	98.61	98.46	
Benign	15	99.08	98.77	98.91	98.92
Malware		99.02	98.68	98.92	

Table 3. Performance Comparison using CIC-AAGM2017

Class	Clients	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
Adware	2	93.48	92.88	92.85	93.85
General Apps		94.86	94.40	93.62	
Adware	7	99.12	99.14	98.86	99.14
General Apps		99.11	99.11	99.08	

malware detection system. To protect against distributed attacks, we present a semantic-driven FL strategy that employs transformer-based transfer learning. Transformers use advanced sequence modeling and contextual intelligence to evaluate complex network traffic patterns. These multi-head attention mechanisms enable efficient sequential data processing, making them suitable for anomaly detection, feature extraction, and network security. The transformer examines the semantic features of harmful scripts, followed by a DNN for local training and model updates for each client. The federated server aggregates incoming local model updates into global updates and transmits them back to clients. This technique enhances privacy and security by running a semantic-based FL process for each client. The proposed approach is analyzed using two standard datasets, CIC-AndMal2017, and CICMalDroid202.

FL offers security and privacy benefits, but it also faces challenges. In practical settings, data aggregation can be challenging due to the variety of client data types and sources. This diversity increases the risk of inaccurate and biased predictions. The following sections address research constraints and future directions.

- We employed federated averaging to update the GMU on the global server. Various aggregating functions can be chosen and studied to demonstrate the efficacy of the proposed approach. For instance, the cluster federated averaging operator employs k-means to generate new weights from centroids. Furthermore, when presenting various aggregation functions, ML models are an interesting option for optimizing the aggregate.
- Analyze potential attacks on the operation and develop ways to minimize or mitigate their impact. For instance, attackers could corrupt the model by modifying it with their own corrupted data. Additionally, attacks on data extraction, weight transmission, etc. One proposed solution to this problem is to use homomorphic encryption to communicate weights between clients and servers.
- Performance and privacy are traded off when comparing models trained on distributed data using the FL algorithm to central data. FL prioritizes privacy over efficiency while protecting ML models. Finding a balance between the privacy and performance of the FL technique is one of the primary concerns with distributed malware attacks.

REFERENCES

1. J. Cho, D. Kim, and H. Kim, "User credential cloning attacks in android applications: exploiting automatic login on android apps and mitigating strategies," *IEEE Consumer Electronics Magazine*, vol. 7, no. 3, pp. 48–55, 2018.
2. X. Pei, X. Deng, S. Tian, L. Zhang, and K. Xue, "A knowledge transfer-based semi-supervised federated learning for iot malware detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 3, pp. 2127–2143, May-June 2023.
3. Z. Li, V. Sharma, and S. P. Mohanty, "Preserving data privacy via federated learning: Challenges and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 3, pp. 8–16, 2020.
4. W. Fang *et al.*, "Comprehensive android malware detection based on federated learning architecture," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 3977–3990, 2023.
5. V. Pourahmadi, H. A. Alameddine, M. A. Salahuddin, and R. Boutaba, "Spotting anomalies at the edge: Outlier exposure-based cross-silo federated learning for ddos detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 4002–4015, 2022.
6. M. Wazid, S. Zeadally, and A. K. Das, "Mobile banking: evolution and threats: malware threats and security solutions," *IEEE Consumer Electronics Magazine*, vol. 8, no. 2, pp. 56–60, 2019.
7. H. Gao, S. Cheng, and W. Zhang, "Gdroid: Android malware detection and classification with graph

convolutional network," *Computers & Security*, vol. 106, p. 102264, 2021.

8. M. Zhang, Y. Duan, H. Yin, and Z. Zhao, "Semantics-aware android malware classification using weighted contextual api dependency graphs," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 1105–1116.
9. H. Sedjelmaci and N. Ansari, "On cooperative federated defense to secure multiaccess edge computing," *IEEE Consumer Electronics Magazine*, vol. 13, no. 4, pp. 24–31, July 2024.
10. H. Sedjelmaci, S.-M. Senouci, N. Ansari, and A. Boualouache, "A trusted hybrid learning approach to secure edge computing," *IEEE Consumer Electronics Magazine*, vol. 11, no. 3, pp. 30–37, 2021.
11. E. Bakopoulou, B. Tillman, and A. Markopoulou, "Fedpacket: A federated learning approach to mobile packet classification," *IEEE Transactions on Mobile Computing*, vol. 21, no. 10, pp. 3609–3628, 2021.
12. Z. Xiong, Z. Cai, D. Takabi, and W. Li, "Privacy threat and defense for federated learning with non-iid data in aiot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1310–1321, 2021.
13. T. D. Nguyen, P. Rieger, R. De Viti, H. Chen, B. B. Brandenburg, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen *et al.*, "{FLAME}: Taming backdoors in federated learning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1415–1432.
14. S. Mahdavifar, A. F. A. Kadir, R. Fatemi, D. Alhadidi, and A. A. Ghorbani, "Dynamic android malware category classification using semi-supervised deep learning," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. IEEE, 2020, pp. 515–522.
15. A. H. Lashkari, A. F. A. Kadir, L. Taheri, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark android malware datasets and classification," in *2018 International Carnahan conference on security technology (ICCST)*. IEEE, 2018, pp. 1–7.

Farhan Ullah Division of Computer Science, University of Camerino, Camerino 62032, Italy.

Leonardo Mostarda Department of Mathematics and Computer Science, University of Perugia, Perugia 06123, Italy.

Diletta Cacciagrano Division of Computer Science, University of Camerino, Camerino 62032, Italy.

Chien-Ming Chen School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China

Saru Kumari Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, Uttar Pradesh, India. (Corresponding Author)