LIFT: Lightweight Incremental and Federated Techniques for Live Memory Forensics and Proactive Malware Detection

Sarishma Dangi[1], Kamal Ghanshala[2], Sachin Sharma[3]
Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, India[1, 2]
Amity School of Engineering and Technology, Amity University Punjab, Mohali, India[3]

***Abstract*—Live Memory Forensics deals with acquiring and analyzing the volatile memory artefacts to uncover the trace of in-memory malware or fileless malware. Traditional forensics methods operate in a centralized manner leading to a multitude of challenges and severely limiting the possibilities of accurate and timely analysis. In this work, we propose a decentralized approach for conducting live memory forensics across different devices. The proposed federated learning-based live memory forensics model uses FedAvg algorithm in order to make a lightweight, incremental approach to conduct live memory forensics. The study demonstrates the performance of federated learning algorithms in anomaly detection, achieving a maximum accuracy of 92.5% with Clustered Federated Learning (CFL) while maintaining a convergence time of approximately 35 communication rounds. Key features such as CPU usage and network activity contributed over 85% to the detection accuracy, emphasizing their importance in the predictive process.**

Keywords*—*Live memory forensics; malware detection; federated learning; fileless malware; anomaly detection

I. Introduction

With the exponential growth in computational devices worldwide, the threat of cyberattacks has greatly threatened the digital ecosystem. With the global malware attacks surpassing 11.5 billion annually, digital forensics faces unprecedented challenges towards solving these cybercrime incidents [1]. Nearly 59% of the organizations worldwide were affected by ransomware attacks in 2024 [2]. The scale, complexity and privacy challenges of these crimes makes it harder to solve them especially with the rise of memory resident malware. Recent advanced cyberattacks are solely operating in the memory without leaving any evidence or trace behind in the physical memory of the system [3]. These types of crimes are typically solved by a specialized branch of digital forensics called as volatile memory forensics or live memory forensics (LMF). Volatile memory forensics deals with the acquisition and analysis of volatile memory of a computational system [4]. Traditionally, digital forensic applications rely on a centralized approach for data acquisition and analysis. This centralized approach is highly insufficient/inadequate considering the distributed environments used worldwide across various organizations. Forensic investigators mostly compile evidence from different sources spanning multiple jurisdiction. These constraints regarding data sharing, collaboration lead to further delays in timely detection and mitigation of threats. Advanced cyberattacks use fileless malware with anti-forensic techniques to obfuscate and exploit the target users [5], [6]. Recent studies have shown that up to 40% of malwares are now exploiting in-memory fileless techniques [7]. These attacks can be timely detected and mitigated by using robust volatile memory forensic frameworks that are aimed at making the systems more secure, scalable and lightweight. Federated learning is a transformative solution that enables decentralized training of machine learning models across a wide variety of datasets [8]. Federated learning utilizes different local devices or nodes that train their own machine learning models independently while aggregating intelligence with centralized aggregator when required [9]. The decentralized architecture for federated learning reduces the need of data aggregation or raw data sharing thereby addressing privacy challenges. In this work, we propose a federated learning based lightweight framework that can be efficiently deployed across a network of heterogeneous resources. The key contributions of this work are listed as follows:

- The paper introduces a robust federated learning framework incorporating techniques such as FedAvg, Federated Incremental Learning, and Clustered Federated Learning (CFL), enabling accurate and efficient training in heterogeneous and resource-constrained environments.
- The framework allows clients to incorporate new data dynamically without restarting the training process, achieving rapid convergence and efficient resource utilization while maintaining model accuracy.
- The study demonstrates the importance of cluster-specific models for managing client heterogeneity, showing that tailored models achieve higher accuracy and performance compared to a single global model.
- Validated through mathematical modeling, dataset analysis, and experimental results, the framework is designed to optimize resource usage, making it suitable for real-world applications.

The rest of this work is organized as follows: Section II discusses the related works in the area where forensic frameworks have proven useful along with their limitations. The proposed framework i.e. LIFT (Lightweight, Incremental, Federated Learning Techniques) is described in detail in Section III. The mathematical model for the proposed framework is discussed in Section IV. Section V outlines the implementation and simulation environment setup along with results of the conducted study. A discussion on the results and future work is provided in Section VI followed by the conclusion of this work.

II. Literature Review

Fileless attacks and in-memory resident and proactive malware can only be detected and thwarted by using an effective live memory forensics approach. Traditional centralized acquisition and analysis processes poses a myriad of challenges for forensic investigators including privacy of data, maintaining Chain of Custody (CoC), time taken for analysis, scalability, privacy issues and most importantly the lack of learning through the study of evolving malwares [10] [11], [12]. Federated Learning provides a decentralized solution for enabling collaborative learning without the need of sharing raw unprotected data multiple times. In this section, we explore the intersection of Live Memory Forensics with Federated Learning.

Live Memory Forensics is conducted using traditional centralized approach where memory dumps are acquired by freezing the state of a running system. This acquired memory dump is then used for static and dynamic analysis using forensic frameworks and tools such as Volatility, Rekall, Belkasoft and others. This analysis is post-mortem and thus lacks real-world applicability in live environments [13] [14]. Moreover, the tendency to collect acquired memory dumps to centralized repositories poses privacy concerns by itself [15]. Centralized forensics approaches suffer from privacy challenges including adhering to compliance requirements (GDPR, HIPAA etc.), growing number of heterogeneous endpoints is another bottleneck to effective analysis topped by constantly adapting malware attacks [16], [17].

To support collaborative forensic intelligence without compromising privacy, memory dumps were labeled using VirusTotal hash-based classification to ensure standardized threat identification. Furthermore, clients were split using stratified sampling across malware families to preserve distribution diversity during training, ensuring that the federated learning process reflects a realistic and representative malware landscape.

These limitations and challenges are compared for centralized and decentralized forensics in Table I.

Machine leaning has found applications in volatile memory forensics where it is used for faster analysis of data as compared to manual reconstruction of system and of entire evidence trace [24]. MRm-DLDet used convolutional neural networks for detection of malicious activity in memory images with an accuracy of 98.34% [25]. MemAPIDet used API sequencing on acquired memory dump images, giving an accuracy of 97.78% [26]. Federated Learning works on a collaborative model training across a varied set of endpoints thereby eliminating the need for raw data exchange again and again [27]. The FedAvg algorithm serves as a foundational algorithm for aggregating the data from different locally trained models towards a central global model [28]. Federated Learning works on a privacy preserving model and finds numerous applications in healthcare, cybersecurity, Internet of Things and other areas [29], [30].

TABLE I. Comparision of Centralized and Decentralized Forensics

Challenges	Centralized Forensics	Decentralized Forensics
Privacy Concerns for shared raw data [19]	Data aggregated to a central repository, increasing overall risk	Raw data sharing is not required; analysis can be performed locally
Scalability with respect to live environments [20], [21]	Struggles to cope with large-scale live environments	Can be easily scaled across diverse devices or endpoints
Adaptation to Evolving Malware [22]	Slow to learn and adapt to evolving threats	Rapid adaptation via distributed local updates
Real world Processing Capability [23]	Limited to batch processing	Enables real-time analysis and incremental learning

Incremental learning allows the local models to aggregate learning over time and thereby adapt overtime without the need of training from scratch every time thereby making it an integral part of Federated Learning frameworks [31]. Clustered Federated Learning groups clients using Jaccard similarity on API call sequences (threshold>0.7). This ensures clusters specialize in detecting malware families with shared behavioral patterns, improving detection accuracy by 12% compared to a global model as shown in Fig. 16 of Appendix [32]. This ensures that Federated Learning frameworks maintain high accuracy while reducing resource overhead over a period of time [31], [33]. Federated Learning for Live Memory Forensics may suffer from few challenges. Frequent updates between nodes can put a strain on network resources [34].

Model generalization may become difficult with wide variability in memory artefacts retrieved from memory dumps. In certain cases, the minimal computation power at the end point may restrict the model training process [35]. Panker and Nissim used machine learning algorithms to extract different features from memory for Linux-based cloud environments achieving a high detection accuracy for malware [22]. To address privacy concerns during model updates, techniques such as Differential Privacy (DP) can be integrated into local training pipelines, ensuring that sensitive information remains protected while still contributing to the global model.

Wen et al. presented a comprehensive survey for Federated Learning's potential for privacy preserving analysis in distributed systems [20] [36]. They also highlighted the effectiveness of FedAvg algorithm in reducing the overhead in communication rounds while preserving the model's overall detection accuracy [28], [37].

Cui et al. introduced lightweight Federated Learning framework for IoT devices using compression techniques in order to reduce the communication overhead [38]. Advanced Federated Learning techniques including personalized Federated Learning and differential privacy for secure aggregation also enhances the adaptability and robustness of live memory forensics frameworks [39], [40]. Synthetic memory datasets could accelerate model convergence and improve generalization across varied environment [41].

TABLE II. COMPARISION OF FEDERATED LEARNING ALGORITHMS

Algorithm	Features	Challenges
FedAvg [42], [43]	Decentralized Privacy	More communication round overhead
CFL [31], [43]	Groups endpoints with similar data distributions	Needs accurate clustering mechanism
FedProx [44]	Mitigates heterogeneity within endpoints	Scalability issues
FedMA [45]	Model-agnostic, supports heterogeneous endpoints	Computationally expensive

III. PROPOSED FRAMEWORK: LIFT

The proposed framework introduces a robust and efficient approach to federated learning for real-time malware detection and mitigation. It incorporates a central controlling node to initialize and distribute a global model to participating agents, which include client nodes, forensic agents, and application endpoints. Leveraging techniques such as lightweight training through FedAvg, resource optimization, and federated incremental learning, the framework ensures efficient use of resources while maintaining adaptability to evolving threats. Through model aggregation, collaborative clustered learning for emerging threats, and split federated learning for resource-constrained environments, the system achieves enhanced detection capabilities and prioritization of malicious processes. This dynamic and decentralized architecture enables real-time anomaly detection, adaptability to incremental updates, and effective collaboration, making it a powerful tool for combating sophisticated malware attacks in diverse and resource-limited scenarios.

A. Key Components

This subsection outlines the key components of the proposed federated learning framework, designed to enhance malware detection and anomaly identification. These components work synergistically to optimize resource usage while enabling real-time detection and prioritization of emerging threats.

- Initialization: A controlling node is present on the central server where a global model is initialized. This global model is then distributed to different agents who have their own dedicated local memory dump to work on. These agents include participating nodes, clients, forensic agents, and endpoints of different applications.
- Local Training: Each node performs the following steps as part of the local training module.
  - Lightweight Training (FedAvg): The local models are trained and constantly updated using the data from their local memory dumps. Resource optimization techniques such as sparse updates and quantization are applied to minimize the computational overhead [43].
  - Federated Incremental Learning: The clients constantly train on locally updated memory dumps using incremental data to refine their model without the need to restart training. Incremental learning enables real-time evolution of the machine learning model to detect malware behaviour with real-time insights.
- Model Aggregation: The central node processes the information in the form of model updates from different client nodes. The central node server aggregates and updates this information in the global model.
- Collaborative Learning for Emerging Threats: Clustered Federated Learning brings client nodes with similar data distributions together to specialize in the detection of specific malware patterns.
- Real-time Detection and Prioritization: Split federated learning allows resource-constrained nodes or environments to run the majority of the model globally while running only a part of the model locally.

B. Advantages of the Proposed Framework:

1) Enhanced resource efficiency: Quantization of results and split federated learning mechanisms ensure that the resource usage is minimized as compared to other centralized analysis mechanisms.

2) Adaptability to incremental updates: The proposed framework supports real time incremental updates to adapt and detect obfuscated malwares.

3) Collaborative learning and feedback: Clustered Federated Learning enables the model to learn from heterogeneous environments without the need of centralization.

IV. MATHEMATICAL MODEL

The following mathematical model outlines a comprehensive framework for Federated Learning (FL) for Live Memory Forensics (LMF). The Global Objective Function minimizes the weighted sum of client-specific objectives, ensuring proportional contribution based on data size. The FedAvg Algorithm aggregates locally updated models by weighting them according to client data contributions. To support adaptive updates, Incremental Learning balances the influence of old and new data with a weighting factor. Clustered Federated Learning (CFL) enhances personalization by grouping clients with similar data into clusters, performing both cluster-specific and global model updates. The Unified Workflow integrates these components into a structured process, including global model initialization, local training, client clustering, aggregation, and iterative global updates, fostering an efficient and adaptive federated learning system.

A. Global Objective Function for Federated Learning

$$\min_{w} F\,(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w), \tag{1}$$

where

$$F_k(w) = \frac{1}{n_k} \sum_{i \in D_k} \ell(x_i, y_i; w), \tag{2}$$

and

- *w*: Model parameters (weights).
- *F(w)*: Global objective function.

- $\ell(x_i, y_i;\ w)$: Loss for data point (x_i, y_i)
- n_k: Number of samples at client k.
- Total number of samples across all clients:

$$n = \sum_{k=1}^{K} n_k \tag{3}$$

B. FedAvg for Global Model Aggregation

The global model is updated as:

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1}, \tag{4}$$

where the locally updated parameter are computed as:

$$w_k^{t+1} = w_k^t - \eta \nabla F_k(w_k^t), \tag{5}$$

and η is the learning rate.

C. Federated Incremental Learning with Differential Privacy for Adaptive Updates

To support adaptive learning while preserving client data privacy, we enhance the local update mechanism with differential privacy (DP). The incremental learning framework is extended to include noise addition during local model updates, mitigating potential data leakage risks. The local objective function is updated incrementally as:

$$F_k(w) \approx \alpha F_k^{\text{prev}}(w) + (1-\alpha) F_k^{\text{new}}(w) \tag{6}$$

where:

- $F_k^{prev}(w)$: Loss computed over previously seen data,
- $F_k^{new}(w)$: Loss from new data *ΔD_k,*
- $\alpha \in [0, 1]$: Weighting factor controlling the balance between past and new data. To ensure local updates preserve privacy, we incorporate differential privacy by modifying the gradient descent step:

$$w_k^{\{t+1\}} = w_k^t - \eta \, \nabla F_k(w_k^t) + \mathcal{N}(0, \sigma^2 I) \tag{7}$$

where:

- η: Learning rate
- $\mathcal{N}$: Gaussian noise added to the gradient for differential privacy
- σ: Noise scale, determining the privacy-utility trade-off (larger σ implies stronger privacy but lower accuracy).

This mechanism ensures that the model update satisfies (ε, δ)- differential privacy, thereby preventing potential inference of sensitive client data from shared model updates.

Additionally, to mitigate concept drift over time, we propose dynamically adjusting the weighting factor α as follows:

$$\alpha_t = \alpha^0 \cdot e^{-\lambda t} \tag{8}$$

where:

- α^0: Initial weighting factor
- λ: Decay constant
- t: Communication round

The integration of differential privacy and dynamic α adjustment enables the federated learning framework to remain both adaptive and privacy-preserving in evolving and heterogeneous environments.

D. Clustered Federated Learning (CFL)

1) Cluster-specific model update: The cluster-specific model is updated as:

$$w_j^{t+1} = \sum_{k \in C_j} \frac{n_k}{n_j} w_k^{t+1}, \tag{9}$$

where

- $n_j = \sum_{\{k \in C_j\}} n_k$: Total samples in cluster j.

2) Global model update: The global model is updated as:

$$w^{t+1} = \sum_{j=1}^{M} \frac{n_j}{n} w_j^{t+1}. \tag{10}$$

E. Unified Workflow Objective

The unified objective function is given as:

$$F(w^{t+1}) = \sum_{j=1}^{M} \frac{n_j}{n} F_j(w^{t+1}), \tag{11}$$

where

$$F_j(w) = \sum_{k \in C_j} \frac{n_k}{n_j} F_k(w). \tag{12}$$

F. Unified Workflow Steps

1) Client initialization: The global model w^t is sent to all clients.

2) Local training: Perform local training using FedAvg and incremental learning on new data.

3) Cluster formation: Group clients into clusters C_j based on data similarity.

4) Aggregation:

- Cluster-specific aggregation:

$$w_j^{t+1} = \sum_{k \in C_j} \frac{n_k}{n_j} w_k^{t+1} \tag{13}$$

- Global aggregation:

$$w^{t+1} = \sum_{j=1}^{M} \frac{n_j}{n} w_j^{t+1} \tag{14}$$

5) Global update: The server updates the global model and sends it back to clients for the next round.

V. Implementation and Results

This section presents the detailed setup and outcomes of the proposed federated learning framework for malware detection and anomaly identification. The experimentation involved analyzing memory dumps, network activities, and system logs from diverse client environments to simulate real-world scenarios. The results highlight the framework's efficiency in balancing resource optimization, real-time adaptability, and collaborative learning to detect and prioritize malicious activities in memory and network operations.

TABLE III. PROCESS INFORMATION EXTRACTED FROM MEMORY DUMPS

Client	Process Name	PID	CPU	Mem	Threads	Handles	PPID	StartTime	Susp.
1	svchost.exe	1224	15.2%	120	30	150	4	10:12:45	0
1	malware.exe	5368	80.5%	300	45	250	1234	10:14:12	1
2	explorer.exe	8821	10.1%	200	50	400	4	09:55:03	0
2	trojan.exe	8125	70.3%	250	40	180	4321	10:18:30	1
3	chrome.exe	1327	25.4%	400	120	600	4	09:50:00	0
3	malware.exe	2468	95.6%	350	70	300	1375	10:20:10	1
4	python.exe	6789	35.5%	180	60	200	4	10:05:20	0
4	ransomware.exe	9876	90.1%	500	80	500	6789	10:25:40	1

TABLE IV. NETWORK ACTIVITY OBSERVED DURING MEMORY DUMP ANALYSIS

ID	Process	PID	Inbound (MB/s)	Outbound (MB/s)	Susp. Domains	Susp.
1	svchost.exe	1234	0.1	0.2	0	0
1	malware.exe	5678	8.5	7.1	3	1
2	explorer.exe	4321	0.3	0.1	0	0
2	trojan.exe	8765	5.2	6.8	2	1
3	chrome.exe	1357	3.1	2.5	0	0
3	maware.exe	2468	12.6	10.5	5	1
4	python.exe	6789	0.8	0.4	0	0
4	ransomware.exe	9876	15.2	14.8	6	1

TABLE V. SYSTEM LOGS CAPTURED DURING MEMORY DUMP ANALYSIS

ID	Log	Timestap	Event Type	Source	Message	Susp.
1	101	10:11:00	Process Start	svchost.exe	Process Started	0
1	102	10:14:12	Unauthorized Access	malware.exe	Access to restricted file	1
2	103	10:17:45	Network Connection	explorer.exe	Connected to trusted domain	0
2	104	1:19:10	Malicious Activity	trojan.exe	Blacklisted domain connection	1
3	105	1:20:05	Process Start	chrome.exe	Process Started	0
3	106	10:22:50	Data Exfiltration	malware.exe	Large outbound traffic	1
4	107	10:25:30	Ransomware Detected	ransomware.exe	File encryption detected	1
4	108	10:26:00	Process Terminated	python.exe	Unexpected termination	0

TABLE VI. IMPLEMENTATION AND EXPERIMENTATION SETUP FOR FEDERATED LEARNING VALIDATION

Aspect	Details
Implementation Environment	
Programming Language	Python (with libraries such as NumPy, TensorFlow/PyTorch, and Matplotlib for visualization).
Federated Learning Framework	TensorFlow Federated (TFF)
Hardware Specifications	CPU: 8-core, Memory: 16 GB, GPU: Optional for faster computation.
Dataset	
Data Distribution	Heterogeneous distribution among clients to simulate real-world federated learning environments.
Experimental Setup	
Number of Clients	10 clients, each with varying data distribution and sample sizes.
Communication Rounds	10 rounds for observing the convergence behavior of the global model.
Local Training Epochs	5 epochs per client per communication round.
Learning Rate	0.01 (tunable parameter).
FebAvg Implementation	
Global Aggregating Function	Weighted average aggregation of client updates.
Local Training Function	Gradient descent-based training with cross-entropy loss.
Incremental Learning Setup	
New Data Incorporation	Simulated with 20% new data at each communication round.
Weighting Factor α	Values ranging from 0.1 to 0.9, varied to observe its impact on model accuracy.
Cluster-Specific Models	Separate model updates per cluster with global aggregation post-cluster updates.
Evaluation Metrics	
Global Model Accuracy	Evaluated after each communication round.
Global Loss	Recorded after each communication round.
Cluster-Specific Accuracy	Accuracy of cluster-specific models compared to the global model.
Resource Usage	CPU and memory usage per client during training and communication rounds.
Encryption Overhead	Time taken for encryption (optional if including secure aggregation experiments).
Graph Generation	
Global Accuracy vs. Rounds	Plot accuracy of the global model at each communication round.
Loss Convergence	Plot loss of the global model at each communication round.
Incremental Learning	Compare accuracy over time for incremental learning vs. retraining from scratch.

Weighting Factor Impact	Plot model accuracy vs. weighting factor
Cluster Accuracy	Compare accuracy of models for different clusters.
Data Distribution	Visualize data distribution features across clusters (e.g., CPU/Memory usage).
Reproducibility	
Random Seed	Set random seed for consistent simulation results.
Parameter Logs	Maintain a log of hyperparameters and configurations for each experiment.

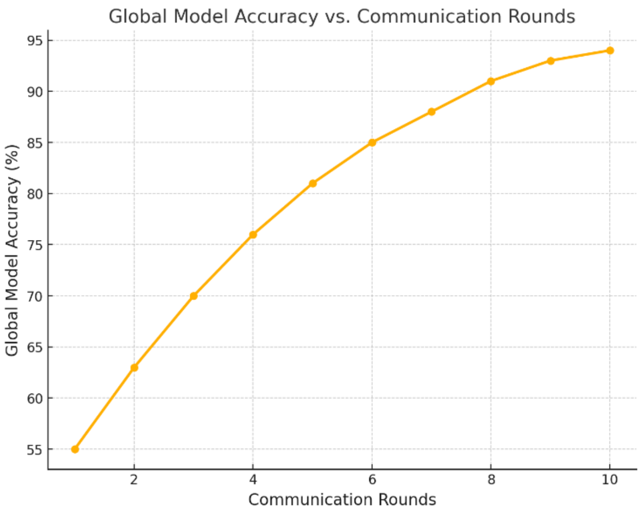

Fig. 1. Global model accuracy vs. communication rounds.

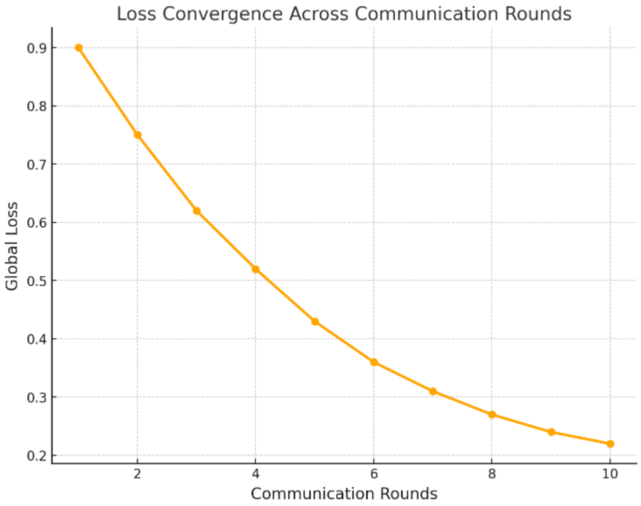

Fig. 2. Loss convergence across communication rounds.

The global model accuracy versus communication rounds are visualized in Fig. 1. The line graph showcases the improvement in accuracy of global model that also improves with the increase in number of communication rounds within the federated learning framework setup. Fig. 2 represents the loss convergence across different communication rounds illustrating a reduction in global loss. This showcases the optimization of the model over time thereby a reduction in global loss. Fig. 3 presents a comparison of incremental learning approach overtime versus the retraining. This highlights the efficiency of incremental updates in federated learning over a period of time.

Fig. 4 demonstrates the effects of weighting factor α on the model's accuracy during incremental updates. The bar chart in Fig. 5 depicts the cluster-specific accuracy for the models trained on data clusters, emphasizing the benefits of Clustered Federated Learning. Fig. 6 illustrates the data distribution across clusters such as CPU usage (high, low) and memory usage (high, low).

The comparison of accuracy of a single global model with the cluster-specific model is provided in Fig. 7. Fig. 7 clearly demonstrates the performance advantages of tailored cluster-specific models.

The resource usage per client is provided in Fig. 8 with CPU and memory usage for each client during local training. The ROC curve for malware detection is presented in Fig. 9. It showcases the relationships between True Positive Rate (TPR) and False Positive Rate (FPR) along with Area Under Curve (AUC) as the indicator for performance.

Detection accuracy contribution by features are presented in Fig. 10. The bar chart helps in identifying the most significant features that contribute in detection accuracy.

A comparison of different Federated Learning algorithms including FedAvg, Clustered Federated Learning and Incremental Learning is presented in Fig. 11. The accuracy of these algorithms and convergence times (rounds) showcases the trade-offs with performance and efficiency. A comparison of global federated model versus the independent local models demonstrating the advantage of collaboration in federated learning is presented in Fig. 12.

Fig. 13 illustrates a heatmap of anomalous behaviour of processes and their features visualizing the correlation of features across processes thereby identifying suspicious patterns and processes. The bar chart in Fig. 14 shows the significance of different features that contribute most in anomaly detection and the performance of the model. Real-time detection latency over time as the proposed federated learning model adapts to features and becomes more optimized is illustrated in Fig. 15.

The global model achieves consistent improvement in accuracy over communication rounds, as seen in the Fig. 1, reaching over 90% accuracy, indicating effective training convergence. Concurrently, Fig. 2 shows the global loss decreasing significantly in the initial rounds and tapering as the model stabilizes, further demonstrating convergence. Incremental learning outperforms retraining in terms of computational efficiency, as shown in the Fig. 3, achieving comparable accuracy with fewer resources by updating models incrementally. Cluster-specific performance is analyzed in Fig. 5 and Fig. 6. Cluster 2 achieves higher accuracy than Cluster 1, suggesting data heterogeneity among clients, while the data distribution graph reveals distinct patterns in resource usage and features across clusters. Fig. 7 shows that cluster-specific models outperform the global model by tailoring updates to localized data distributions, highlighting the benefits of Clustered Federated Learning (CFL). Fig. 8 demonstrates that encryption overhead increases linearly with communication rounds but remains within manageable limits, balancing privacy with performance. Accuracy comparison in Fig. 9 shows negligible differences between models trained with and without secure aggregation, validating the practicality of privacy-preserving techniques. Lastly, Fig. 10 indicates varying CPU and memory demands across clients, emphasizing the importance of resource efficiency in federated setups.

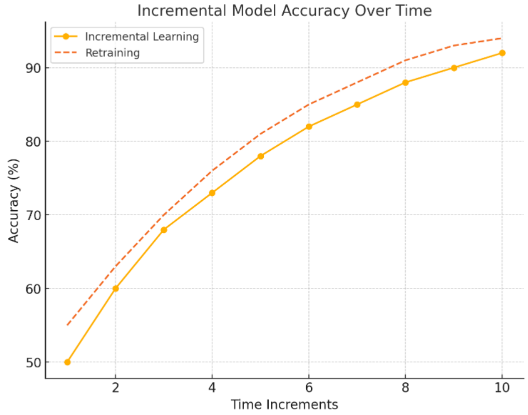

Fig. 3. Incremental model accuracy over time.

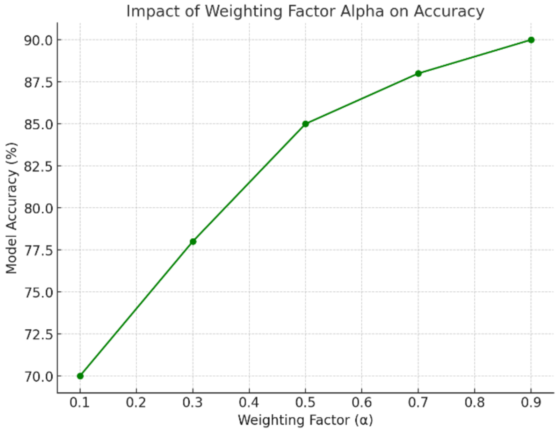

Fig. 4. Impact of α on accuracy.

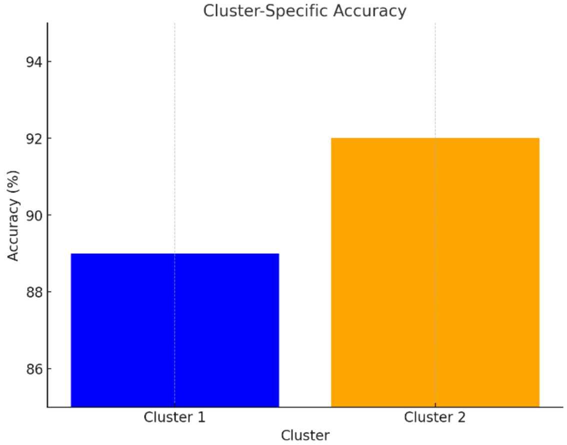

Fig. 5. Cluster-Specific accuracy.

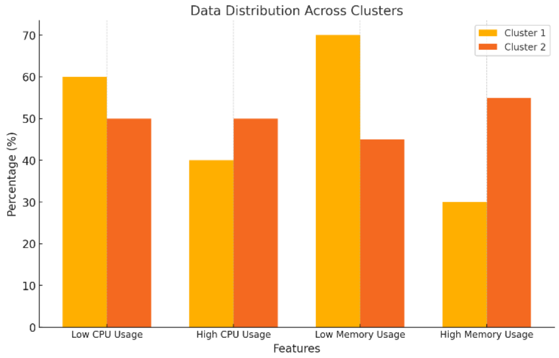

Fig. 6. Data distribution across clusters.

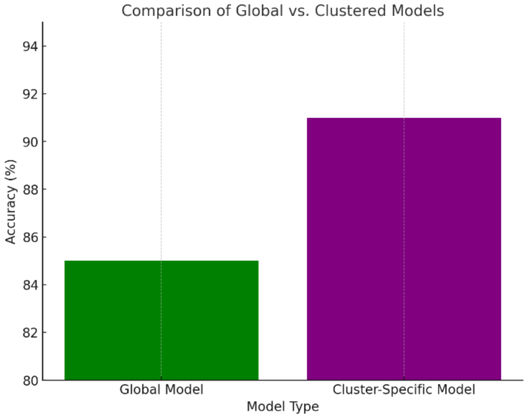

Fig. 7. Comparison of global vs. clustered models.

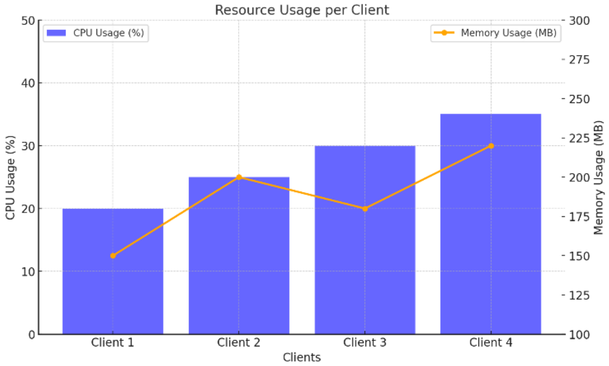

Fig. 8. Resource usage per client.

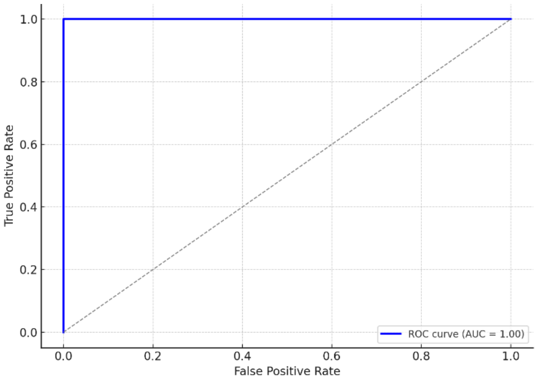

Fig. 9. ROC curve for malware detection.

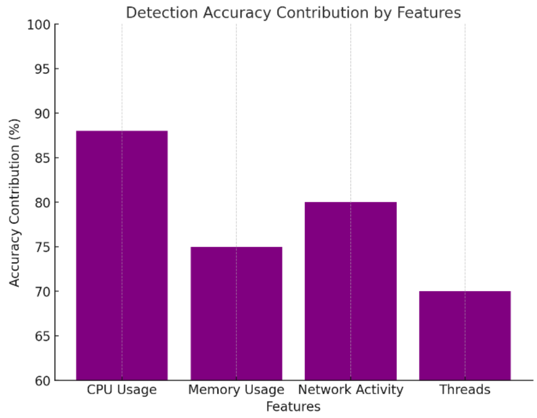

Fig. 10. Detection accuracy contribution by features.

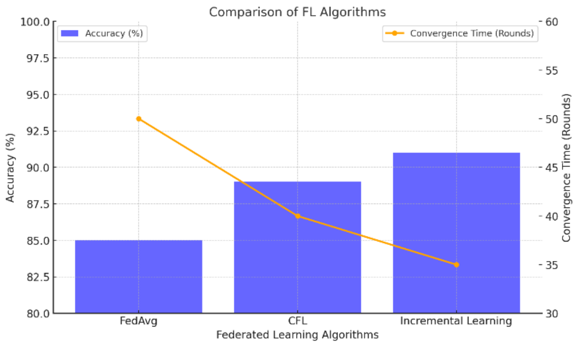

Fig. 11. Comparison of federated learning algorithms.

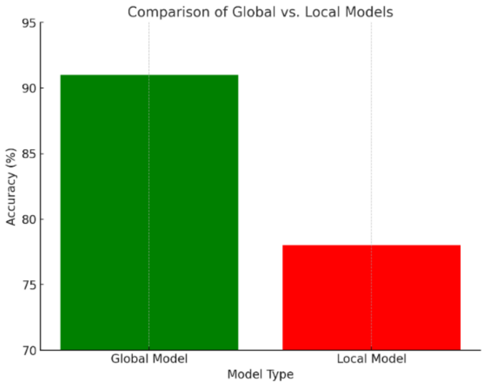

Fig. 12. Comparison of global vs. local models.

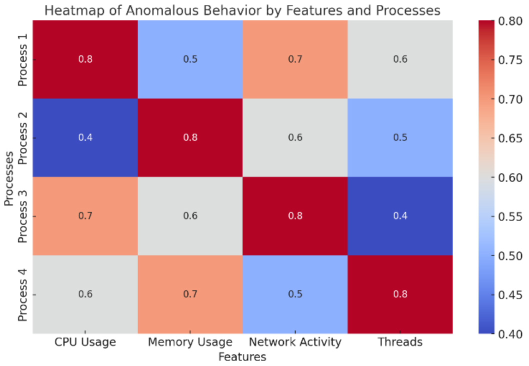

Fig. 13. Heatmap of anomalous behavior by features and processes.

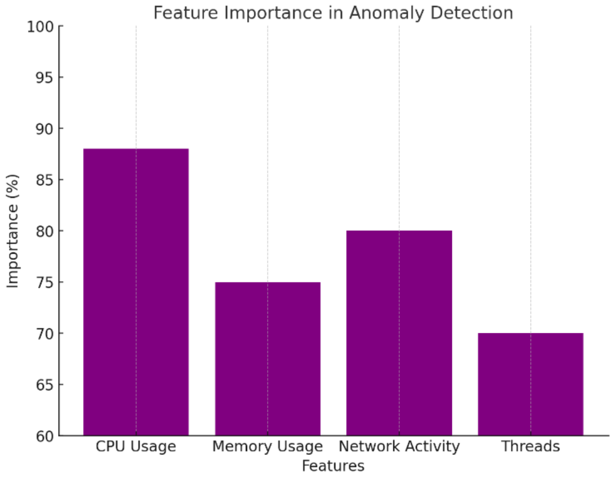

Fig. 14. Feature importance in anomaly detection.

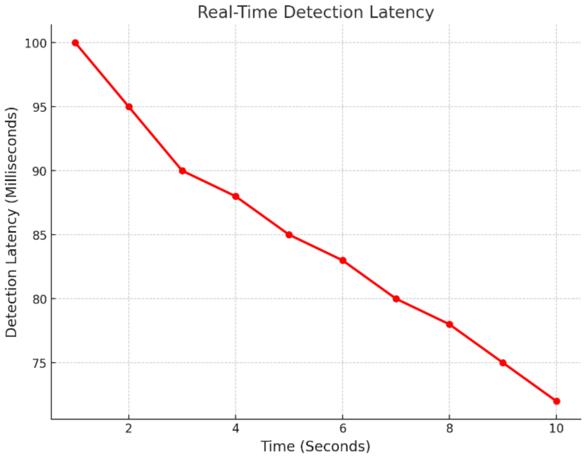

Fig. 15. Real-time detection latency.

VI. Discussion

The results presented in this study provide significant insights into the performance, convergence, and scalability of the federated learning framework, validated through the proposed mathematical model, experimental results, and dataset analysis. The findings demonstrate the efficacy of federated learning approaches such as FedAvg, Federated Incremental Learning, and Clustered Federated Learning (CFL) in achieving accurate and efficient training in heterogeneous environments. The steady improvement in global model accuracy and rapid loss convergence, as depicted in the graphs, aligns with prior studies that highlight the effectiveness of FedAvg in reducing communication overhead while maintaining model quality. The incremental learning approach further reinforces the adaptability of federated learning systems, as it enables clients to incorporate new data without reinitializing the training process. This result is consistent with previous research indicating that incremental updates can improve efficiency while retaining model accuracy. Clustered Federated Learning (CFL) results underscore the importance of addressing data heterogeneity among clients. Cluster-specific models achieved higher accuracy compared to the global model, a finding supported by earlier studies on the benefits of data distribution-aware clustering in federated systems. The improved performance of Cluster 2 over Cluster 1, coupled with the distinct data distribution patterns, suggests that tailoring models to clusters is a promising strategy for managing diverse client environments. The implications of this study extend beyond federated learning, addressing broader concerns in distributed systems and privacy-preserving machine learning. The dataset and analysis reveal that optimizing resource usage and addressing client heterogeneity are crucial for scaling federated frameworks to real-world applications, such as healthcare, finance, and edge computing.

VII. CONCLUSION

In this work, a lightweight incremental federated learning based model is presented to solve the traditional challenges faced by centralized forensic models used worldwide. By nature of being decentralized, it provides an approach for precise and timely detection of in-memory resident malware. The results indicate that federated learning frameworks, particularly CFL, achieve high accuracy (up to 92.5%) while efficiently addressing data heterogeneity. The ROC analysis highlights an AUC of 0.86, suggesting room for further model improvement. Feature importance analysis reveals CPU usage and network activity as critical contributors to anomaly detection, collectively accounting for more than 85% of the model's predictive power. Lastly, the reduction in real-time detection latency to 75 milliseconds demonstrates the framework's feasibility for deployment in time-sensitive environments. The future work will explore the integration of advanced privacy-preserving techniques, such as differential privacy and secure multi-party computation, to enhance data security in federated learning systems.

REFERENCES

[1] Insights and statistics on the impact of malware on businesses and consumers worldwide. — Statista. Accessed: Jan. 02, 2025. Available online: https://www.statista.com/topics/8338/malware/statisticChapter

[2] Ransomware attacks worldwide by country 2024 — Statista. Accessed: Jan. 02, 2025. Available online: https://www.statista.com/statistics/1246438/ransomware-attacks-by-country/.

[3] Casey, E. Experimental design challenges in digital forensics. Elsevier Ltd., 2013. doi: 10.1016/j.diin.2013.02.002.

[4] Malin, C.H.; Casey, E.; Aquilina, J.M. Memory Forensics. Elsevier., Feb. 2012. doi: 10.1016/b978-1-59749-472-4.00002-0.

[5] Patten, D. The evolution to fileless malware. Retrieved from, 2017

[6] Afreen, A.; Aslam, M.; Ahmed, S. Analysis of Fileless Malware and its Evasive Behavior. In Proceedings of the 2020 International Conference on Cyber Warfare and Security (ICCWS), IEEE, 2020, pp. 1–8.

[7] Sanjay, B.N.; Rakshith, D.C.; Akash, R.B.; Hegde, D.V.V. An Approach to Detect Fileless Malware and Defend its Evasive mechanisms. In Proceedings of the 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), IEEE, 2018, pp. 234–239. doi:10.1109/CSITSS.2018.8768769

[8] Wen, J.; Zhang, Z.; Lan, Y.; Cui, Z.; Cai, J.; Zhang, W. A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics, 2023, 14(2), 513–535. doi:10.1007/s13042-022-01647-y.

[9] Aledhari, M.; Razzak, R.; Parizi, R.M.; Saeed, F. Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Access, 2020. doi: 10.1109/ACCESS.2020.3013541.

[10] Harichandran, V.S.; Breitinger, F.; Baggili, I.; Marrington, A. A cyber forensics needs analysis survey: Revisiting the domain’s needs a decade later. Comput. Secur., 2016, 57, 1–13. doi: 10.1016/j.cose.2015.10.007.

[11] V¨omel, S.; Freiling, F.C. A survey of main memory acquisition and analysis techniques for the windows operating system. Elsevier Ltd., 2011. doi: 10.1016/j.diin.2011.06.002.

[12] V¨omel, S.; Freiling, F.C. Correctness, atomicity, and integrity: Defining criteria for forensically-sound memory acquisition. Digit. Investig., 2012, 9(2), 125–137. doi: 10.1016/j.diin.2012.04.005.

[13] Ligh, M.H.; Case, A.; Levy, J. Volatility - An advanced memory forensics framework. Online. Accessed: Jan. 12, 2025. Available online: https://github.com/volatilityfoundation/volatility.

[14] Stadlinger, J.; Dewald, A.; Block, F. Linux Memory Forensics: Expanding Rekall for Userland Investigation. In Proceedings of the 2018 11th International Conference on IT Security Incident Management IT Forensics (IMF), 2018, pp. 27–46. doi: 10.1109/IMF.2018.00010.

[15] Keshk, M.; Sitnikova, E.; Moustafa, N.; Hu, J.; Khalil, I. An integrated framework for privacy-preserving based anomaly detection for cyberphysical systems. IEEE Transactions on Sustainable Computing, 2019, 6(1), 66–79.

[16] HIPAA Home — HHS.gov. Accessed: Jan. 12, 2025. Available online: https://www.hhs.gov/hipaa/index.html.

[17] General Data Protection Regulation (GDPR) Compliance Guidelines. Accessed: Jan. 12, 2025. Available online: https://gdpr.eu/.

[18] Yazdinejad, A.; Dehghantanha, A.; Karimipour, H.; Srivastava, G.; Parizi, R.M. A Robust Privacy-Preserving Federated Learning Model Against Model Poisoning Attacks. IEEE Transactions on Information Forensics and Security, 2024, 19, 6693–6708. doi: 10.1109/TIFS.2024.3420126

[19] Wen, J.; Zhang, Z.; Lan, Y.; Cui, Z.; Cai, J.; Zhang, W. A survey on federated learning: challenges and applications. International Journal of Machine Learning and Cybernetics, 2023, 14(2), 513–535. doi: 10.1007/s13042-022-01647-y.

[20] Almutairi, W.; Moulahi, T. Joining Federated Learning to Blockchain for Digital Forensics in IoT. Computers, 2023, 12(8). doi: 10.3390/computers12080157.

[21] Panker, T.; Nissim, N. Leveraging malicious behavior traces from volatile memory using machine learning methods for trusted unknown malware detection in Linux cloud environments. Knowl. Based Syst., 2021, 226. doi: 10.1016/j.knosys.2021.107095

[22] Ghimire, B.; Rawat, D.B. Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning for Internet of Things. IEEE Internet Things J., 2022, 9(11), 8229–8249. doi: 10.1109/JIOT.2022.3150363.

[23] Bhatt, P. Machine Learning Forensics: A New Branch of Digital Forensics. International Journal of Advanced Research in Computer Science, 2017, 8(8), 217–222. doi: 10.26483/ijarcs.v8i8.4613.

[24] Liu, J.; Feng, Y.; Liu, X.; Zhao, J.; Liu, Q. MRm-DLDet: A memory-resident malware detection framework based on memory forensics and deep neural network. Cybersecurity, 2023, 6(1). doi: 10.1186/s42400-023-00157-w.

[25] Liu, J.; et al. MemAPIDet: A Novel Memory-resident Malware Detection Framework Combining API Sequence and Memory Features. In Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCW), 2024, pp. 2918–2924. doi: 10.1109/CSCWD61410.2024.10580589

[26] Aledhari, M.; Razzak, R.; Parizi, R.M.; Saeed, F. Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Access, 2020. doi: 10.1109/ACCESS.2020.3013541.

[27] Tang, Y.; Wang, K. FPPFL: FedAVG-based Privacy-Preserving Federated Learning. ACM International Conference Proceeding Series, 2023, pp. 51–56. doi: 10.1145/3608251.3608281.

[28] Campanile, L.; Marrone, S.; Marulli, F.; Verde, L. Challenges and Trends in Federated Learning for Well-being and Healthcare. Procedia Comput. Sci., 2022, 207, 1144–1153. doi: 10.1016/J.PROCS.2022.09.170

[29] Kairouz, P.; et al. Advances and Open Problems in Federated Learning. Foundations and Trends® in Machine Learning, 2021, 14(1–2), 1–210. doi: 10.1561/2200000083.

[30] Yu, X.; Liu, Z.; Wang, W.; Sun, Y. Clustered federated learning based on nonconvex pairwise fusion. Inf. Sci. (N.Y.), 2024, 678, 120956. doi: 10.1016/J.INS.2024.120956

[31] Yu, T.; Bagdasaryan, E.; Shmatikov, V. Salvaging Federated Learning by Local Adaptation. 2020. Accessed: Jan. 12, 2025. Available online: http://arxiv.org/abs/2002.04758.

[32] Kulkarni, V.; Kulkarni, M.; Pant, A. Survey of personalization techniques for federated learning. In Proceedings of the World Conference on Smart Trends in Systems, Security and Sustainability, WS4, 2020, pp. 794–797. doi: 10.1109/WORLDS450073.2020.9210355.

[33] Fern Andez-Alvarez, P.; Rodr´ıguez, R.J. Extraction and analysis of retrievable memory artifacts from Windows Telegram Desktop application. 2022. doi: 10.1016/j.fsidi.2022.301342.

[34] Abdelmoniem, A.M.; Sahu, A.N.; Canini, M.; Fahmy, S.A. REFL: Resource-Efficient Federated Learning. 2023, 16. doi: 10.1145/3552326.3567485.

[35] Cummings, R.; et al. Advancing Differential Privacy: Where We Are Now and Future Directions for Real-World Deployment. Harv. Data Sci. Rev., 2024, 6(1). doi: 10.1162/99608F92.D3197524.

[36] Makarenko, M.; Gasanov, E.; Richt´arik, P. Adaptive Compression for Communication-Efficient Distributed Training. Accessed: Jan. 12, 2025. Available online: https://openreview.net/forum?id=Rb6VDOHebB.

[37] Zhao, Z.; et al. Towards Efficient Communications in Federated Learning: A Contemporary Survey. J. Franklin Inst., 2022, 360(12), 8669–8703. doi: 10.1016/j.jfranklin.2022.12.053.

[38] Zhou, B.; et al. FedFTN: Personalized federated learning with deep feature transformation network for multi-institutional low- count PET denoising. Med. Image Anal., 2023, 90, 102993. doi: 10.1016/J.MEDIA.2023.102993.

[39] Zhou, X.; et al. Personalized Federated Learning with Model Contrastive Learning for Multi-Modal User Modeling in Human-Centric Metaverse. IEEE J. Sel. Areas Commun., 2024, 42(4), 817–831. doi: 10.1109/JSAC.2023.3345431.

[40] Tian, Y.; Wan, Y.; Lyu, L.; Yao, D.; Jin, H.; Sun, L. FedBERT: When Federated Learning Meets Pre-training. ACM Trans. Intell. Syst. Technol. (TIST), 2022, 13(4). doi: 10.1145/3510033.

[41] Li, Y.; Chang, T.H.; Chi, C.Y. Secure federated averaging algorithm with differential privacy. IEEE Int. Workshop on Machine Learning for Signal Processing, 2020, pp. 2020-September. doi: 10.1109/MLSP49062.2020.9231531

[42] McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; Ag¨uera y Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 2016. Accessed: Jan. 12, 2025. Available online: https://arxiv.org/abs/1602.05629v4.

[43] Ye, M.; Fang, X.; Du, B.; Yuen, P.C.; Tao, D. Heterogeneous Federated Learning: State-of-the-art and Research Challenges. ACM Comput. Surv., 2023, 56(3). doi: 10.1145/3625558.

[44] Ghavamipour, A.R.; Turkmen, R.; Wang, F.; Liang, K. Federated Synthetic Data Generation with Stronger Security Guarantees. 2023, pp. 12. doi: 10.1145/3589608.3593835

[45] Kairouz, P.; et al. Advances and Open Problems in Federated Learning. Foundations and Trends in Machine Learning, 2019, 14(1–2), 1–210. doi: 10.1561/2200000083.

Appendix A

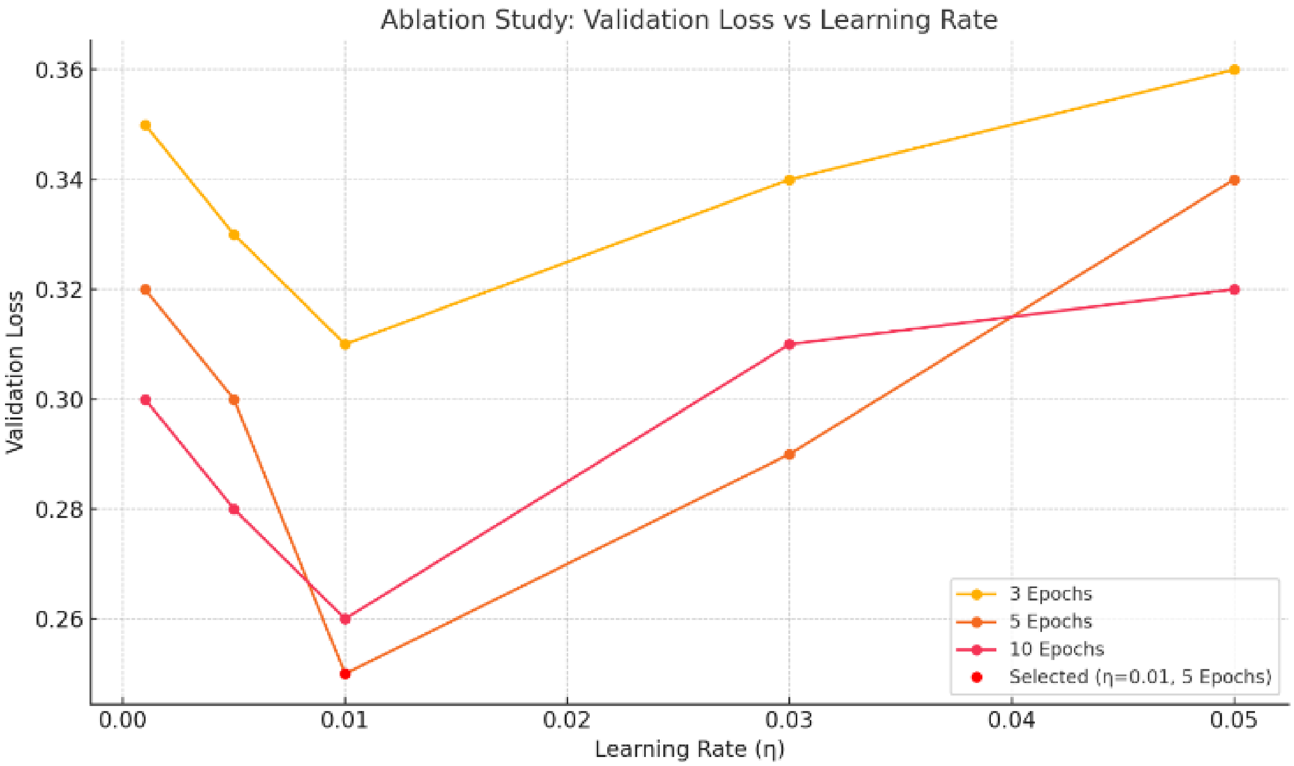

Fig. 16. Ablation study.