Banking Malware Detection: Leveraging Federated Learning with Conditional Model Updates and Client Data Heterogeneity

Nahid Ferdous Aurna[1], Md Delwar Hossain[1], Hideya Ochiai[2], Yuzo Taenaka[1], Latifur Khan[3] and Youki Kadobayashi[1]

[1]*Division of Information Science, Nara Institute of Science and Technology, Nara, Japan*

[2]*Grad. School of Info. Science and Tech., The University of Tokyo, Tokyo, Japan*

[3]*Computer Science Department, The University of Texas at Dallas, Richardson, U.S.A.*

Keywords: Banking Malware, Federated Learning, Ensemble Learning, Data Heterogeneity.

Abstract: Banking malware remains an ongoing and evolving threat as cybercriminals exploit vulnerabilities to steal sensitive user information in the digital banking landscape. Despite numerous efforts, developing an effective and privacy preserving solution for detecting banking malware remains an ongoing challenge. This paper proposes an effective privacy preserving Federated Learning (FL) based banking malware detection system utilizing network traffic flow. Challenges such as, dealing with data heterogeneity in FL scheme while maintaining robustness of the global shared model are addressed here. In our study, three distinct heterogenous datasets consisting benign and one of the prevalent malicious flows (zeus, emotet, or trickbot) are considered to address the data heterogeneity. To ensure model's robustness, initially, we assess various models, selecting Convolutional Neural Network (CNN) for developing an ensemble model. Subsequently, FL is incorporated to maintain data confidentiality and privacy where ensemble model serves as the global model ensuring the effectiveness of the approach. Moreover, to improve the FL scheme, we introduce conditional update of client models, effectively addressing data heterogeneity among the federated clients. The evaluation results demonstrate the effectiveness of the proposed model, achieving high detection rates of 0.9819, 0.9982, and 0.9997 for client 1, client 2, and client 3, respectively. Overall, this study offers a promising solution to detect banking malware while effectively addressing data privacy and heterogeneity in the FL framework.

1 INTRODUCTION

With the digitization of banking and financial activities, cybercrime has become predominantly financially motivated. Sophisticated banking malwares like zeus, emotet, and trickBot act as the major drivers of malicious activities targeting sensitive information, specifically login credentials. Banking malwares exhibit distinct network packet behaviors, establishing encrypted command-and-control communication, while other malware types may vary in communication patterns and objectives at the network level. These malwares continue to pose significant cybersecurity risks in the financial sector and so it's important to develop a robust detection strategy to encounter these critical malwares.

Malware detection is a persistent research problem, with continuous efforts aimed to develop and improve the detection techniques. Numerous research has been conducted in this regard using Machine Learning (ML) and Deep Learning (DL) techniques. For instance, in (Mohaisen and Alrawi, 2013), classical ML methods such as Support Vector Machine (SVM), Logistic Regression (LR), K-Nearest Neighbor (KNN) have been applied for detecting zeus and benign samples. In another study (Kazi et al., 2019a), Decision Tree (DT), Random Forest (RF) and the SVM algorithm have been considered where RF outperformed other models. Various studies have explored DL based method such as CNN (Wang et al., 2017), Capsule Network (Lu et al., 2023) and CNN with Autoencoder (An et al.,) to classify malware. Besides, there are some existing deep learning based fusion and ensemble learning methods for classifying malwares (Liu et al., 2021). The key limitation of this centralized ML/DL based researches lies in not considering privacy of data that has been addressed in (Jiang et al., 2022), (Rey et al., 2022) applying FL based approach. Though these FL based detection approaches are very intriguing, but challenges remain

Aurna, N., Hossain, M., Ochiai, H., Taenaka, Y., Khan, L. and Kadobayashi, Y.
Banking Malware Detection: Leveraging Federated Learning with Conditional Model Updates and Client Data Heterogeneity.
DOI: 10.5220/0012409700003648

In *Proceedings of the 10th International Conference on Information Systems Security and Privacy (ICISSP 2024)*, pages 309-319
ISBN: 978-989-758-683-5; ISSN: 2184-4356

in improving global model's robustness and handling heterogeneous data as FL based approach is prone to data heterogeneity while maintaining global model's performance efficacy.

In this work, we aim to overcome aforementioned limitations of data privacy, heterogeneity and robustness of global model in existing approaches using Federated Learning based scheme. Handling heterogeneous data poses a significant challenge. To address this issue, we utilize three separate datasets, each containing benign samples and a prevalent banking malware type (zeus, emotet, or trickbot). Another challenge remains in ensuring robustness and efficacy of the global model in FL scheme. To deal with this, we conduct isolated training of the datasets with four different DL models: CNN, MLP, LSTM and TabNet (Arik and Pfister, 2020) where CNN exhibits superior performance, consequently we built an ensemble-based CNN model comprised of three personalized trained CNN models on three distinct datasets. Eventually, this ensemble model served as the global model in the FL approach considering three clients, each having one distinct dataset with varying feature sets and malware types. That's how data heterogeneity among all clients is ensured in our study and ensemble of personalized trained CNN models strengthens the efficacy of global shared model. Moreover, to further enhance the performance of FL based scheme, conditional update is incorporated for each of the client models in every communication round. In this way, local models are updated only when the global model's current weight demonstrates an improvement in performance. As a result, the proposed FL-based approach exhibits significant improvement due to the ensemble of personalized trained CNN models and incorporation of conditional updates for federated clients. Furthermore, Experimental evaluation on an unseen multi-class dataset showcases the generalization capability of the proposed model. The key contributions of this work are featured below:

- We propose a privacy preserving FL based approach with conditional model update to detect sophisticated banking malwares considering data heterogeneity
- We conduct personalized training of each dataset and ensemble the best converged models with a view to ensuring the robustness of global FL model.
- We demonstrate a way of improving FL technique by conditional update of all clients' local model at each communication round.
- We ensure the generalization capability of global federated model by testing it with an unseen multi-class dataset consisting all three banking malwares.

The subsequent sections of the paper are structured as follows: Section II reviews existing works related to the problem domain highlighting potential limitations. Section III thoroughly depicts the proposed methodology. Section IV provides a comprehensive demonstration of our result. Section V includes relevant discussions with future directions. Finally, Section VI denotes the concluding remarks of our study.

2 RELATED WORKS

Malware detection has become an intriguing research area to ensure security of computer systems and networks in financial sector. Numerous studies have been conducted exploring distinct approaches to address this challenge.

2.1 Machine Learning Approaches

Gezer et al. (Gezer et al., 2019) investigated the effectiveness of four ML algorithms to detect TrickBot-related traffic flows where Random Forest classifier achieves superior performance identifying malicious flows. However, their study lacks evaluation on unseen malware samples, limiting practical implication and robustness. Wang et al. (Wang et al., 2022) presented a systematic review of 10 ML-based techniques including RF, XGBoost, K Nearest Neighbor (KNN), MLP for detecting encrypted malicious traffic and proposed a comprehensive dataset from multiple sources. However, this study lacks data heterogeneity (having different feature set for different dataset) which could be a significant consideration in this area. Rawat et al. (Rawat et al., 2022) focused on detecting malware affecting financial sector from network traffic flow using several machine learning models i.e., RF, MLP, Logistic Regression (LR), with RF achieving the highest precision. Yet the study was confined to binary classification, restricting the exploration of more diverse malware classes.

2.2 Deep Learning Approaches

Agrafiotis et al. (Agrafiotis et al., 2022) proposed an image-based approach to classify malicious traffic where the raw traffic is transformed to images using vision transformer (ViT) and CNN. Still this approach needs to be experimented with unseen traffic samples to validate the efficacy of this method. Moreover, this work is restricted to detecting malicious

traffic flow without categorizing the type of malware. Mahdavifar et al. (Mahdavifar et al., 2020) exhibited deep learning-based semi supervised method to classify several malware categories outperforming other ML techniques. However, this study lacks consideration of data heterogeneity and data privacy. Fox et al. (Fox and Boppana, 2022) introduced a malicious network traffic detection method using 2D-CNN by converting data packets into bitmap images. Yet, personalized training on different datasets could eradicate the deterioration of performance for certain datasets.

2.3 Federated Learning Approaches

Jiang et al. (Jiang et al., 2022) proposed an FL based Android malware classification scheme namely FedHGCDroid. They introduced a CNN and Graph Neural Network (GNN) based model called HGCDroid for accurate feature extraction and employed FedAdapt to improve adaptability of model in dealing with non-IID (Non-independent and identically distributed) distributions of data. Similarly, Rey et al. (Rey et al., 2022) explored FL for detecting malware in IoT devices to mitigate the data privacy concern. The proposed framework utilized supervised (MLP) and unsupervised (autoencoder) models, attaining promising outcome while ensuring privacy preservation of data.

Despite the undeniable contributions made by the existing ML, DL and FL-based schemes, there exists notable scope for enhancement that has not yet been thoroughly addressed in this domain. Factors such as data heterogeneity across federated clients, personalized training of models for each client, and adaptive weight updates have not been fully considered. In our study, we have endeavored to explore aforementioned critical issues to construct a more robust FL-based malware detection approach.

3 METHODOLOGY

This study presents a federated learning-based approach for banking malware detection using deep learning techniques that can be summarized in five phases as illustrated in Fig. 1. The methodology involves dataset preprocessing, deploying and selecting the best deep learning model for personalized training, building an ensemble model, and implementing federated learning with the ensemble model as the global model. Comprehensive validation is performed, including testing the global model on an unseen multi-class dataset. The proposed approach represents a promising advancement in banking malware detection, offering improved privacy and robustness through federated learning.

3.1 Data Collection and Preprocessing

The datasets we used in our study are created by the Stratosphere Laboratory, publicly available in (Stratosphere, 2015) where traffics were captured by running malware samples in the controlled environment. Basically, we collected network traffic of three types of banking malwares: Zeus, Emotet and Trickbot along with benign traffic in separate PCAP (Packet Capture) files. For the experimental purpose we created three different datasets from these traffics having three diverse type of malwares. Over the years, Zeus malware has evolved into various strains like Zeus Gameover, SpyEye, ICE IX, and Shylock, all geared towards accessing bank accounts with distinct strategies. Emotet and TrickBot have similarly advanced, gaining abilities to compromise user data and even install additional malware. TrickBot has notably refined its evasion methods to avoid detection. These diverse and sophisticated variations make detecting and countering these malware types increasingly complex.

3.1.1 Dataset-1 (Zeus)

For creating the first dataset we considered zeus and benign traffic data and transformed the raw traffic data (PCAP file) to csv file according to the TCP sessions. For this transformation, each TCP session information were taken for first 30 seconds and the features include the number of upload and download bytes for each second. In this way, each session contains 60 features that is upload and download bytes for 30 seconds.

3.1.2 Dataset-2 (Emotet)

For creating the second dataset, similarly we considered emotet and benign traffic data and transformed them into csv file according to the TCP sessions. For this dataset, besides number of upload and download bytes for each second of initial 30 seconds, we considered additional 2 features those are Mean TCP windows size value and Total payload per session. In this way the total number of features is 62 for this dataset.

3.1.3 Dataset-3 (Trickbot)

In case of the third dataset, we considered trickbot and benign traffic data and like the previous 2 datasets, we extracted the features based on each TCP session in the transformed csv file. In this case, the total number of features is 63 which includes all the features in

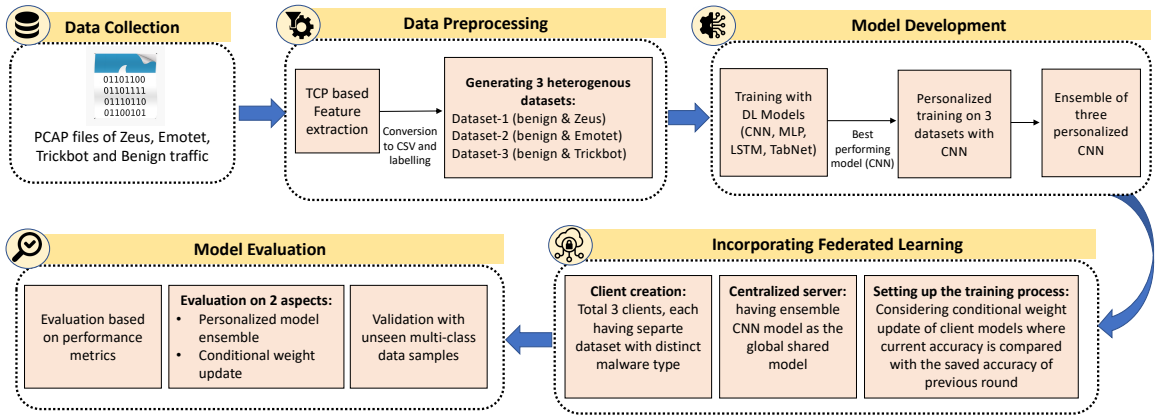

Figure 1: The workflow of the proposed method that encompasses dataset preparation for diverse malware types, ensemble model creation through several DL model evaluation, integration with FL approach incorporating conditional client model updates, and comprehensive validation of the proposed approach.

Dataset-2 with additional one feature that is IP packet length.

For each of the datasets, the malware file and the benign files are labeled, merged and shuffled after transforming into csv format. In this way, we tried to maintain certain amount of heterogeneity among all the datasets considering feature sets, malware types and number of samples. Differences in feature sets across datasets from various clients are common. To showcase proposed FL scheme's ability to handle such heterogeneity, we deliberately introduced slight feature variations. Our aim is to demonstrate the effectiveness of our proposed approach in addressing performance challenges in the FL architecture under such conditions. All these features were selected based on the suggestion of expert in this field and some past studies on the similar problems were considered as well. The data distribution and considered feature sets are mentioned in Table 1 and Table 2 respectively. The discrepancy between real-life data imbalance and the employed dataset is evident from Table 1 as real-world scenarios generally exhibit a higher occurrence of benign traffic. The target is to ensure an unbiased evaluation of our proposed model's performance with the aim to establish a baseline performance under controlled conditions.

The reason behind using TCP based features for our study lies in the fact that TCP is one of the most commonly used protocol to establish the C&C (Command and Control) connections between the compromised host and attacker's server for this kind of malwares. Moreover, we considered the upload and download byte counts as the significant features in our study because by analyzing TCP sessions and their upload/download bytes per second, it is possible to capture the behavior of malware in terms of data transfer, communication frequency, and overall network activity.

Table 1: Sample count of the datasets used in this study.

Dataset	Data Samples	Flow Count
Dataset-1 (Zeus)	Zeus	5100
	Benign	27980
Dataset-2 (Emotet)	Emotet	30000
	Benign	11818
Dataset-3 (Trickbot	Trickbot	22445
	Benign	8145

Table 2: Feature sets of the datasets used in this study.

Features used in this study	Dataset-1 (Zeus))	Dataset-2 (Emotet)	Dataset-3 (Trickbot)
Upload byte count for each second (initial 30 seconds)	✓	✓	✓
Download byte count for each second (initial 30 seconds)	✓	✓	✓
Mean TCP windows size value	–	✓	✓
Total payload per session	–	✓	✓
IP packet length	–	–	✓
Total no of Features	60	62	63

3.2 Model Development

The model development process can be divided into three stages; in the first stage, we did some experimentation on four distinct deep learning models. Secondly, the best models for the three datasets were considered for personalized training and these models are ensembled to create a robust model. Finally, federated learning was incorporated using the ensemble model as the global model.

Table 3: Hyperparameter values for the applied deep learning models.

Hyperparameters	CNN	MLP	LSTM	TabNet
Input activation function	ReLU	ReLU	ReLU	–
Output activation function	Sigmoid	Sigmoid	Sigmoid	–
Optimizer	Adam	Adam	Adam	Adam
Initial learning rate	0.0001	0.0001	0.001	0.02
Learning rate decay	0.3	0.3	0.3	0.2

3.2.1 Experimentation with Deep Learning Models

Initially we started our experimentation with four deep learning models: CNN, MLP, LSTM and TabNet. The architectures of CNN, MLP and LSTM has been depicted in Fig. 2. These architectures have been chosen after several experimentations with each of the datasets, and for the TabNet model we used the pretrained default architecture which is followed from the original paper (Arik and Pfister, 2020). As mentioned in Fig. 2(a), the CNN model consists of total 9 layers including 2 convolution layers, 2 batchnormalization layers, 2 maxpooling layers, 1 faltten layer followed by 2 dense layers. The MLP and LSTM both models have 3 layers as depicted in Fig. 2(b) and (c). The MLP consists of 3 dense layers whereas the LSTM consists of one LSTM layer followed by 2 dense layers.

These models are used for training the three datasets and the hyperparameter values are provided in Table 3. To choose the optimal hyperparameter values, keras tuner is initially used along with some empirical testing. After experimentation, the result is depicted in Table 4 and the purpose of this investigation was to scrutinize the best performing model. To evaluate deep learning models, we considered several evaluation metrics: precision, recall, F1 score, false positive rate (FPR), false negative rate (FNR) and accuracy. Accuracy is a widely used metric to assess the performance of a classification model, representing the proportion of correct predictions over the total predictions made. Precision is the ratio of true positive predictions to the total positive predictions, while recall is the ratio of true positive predictions to the sum of true positive and false negative predictions. The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's effectiveness in capturing both true positives and minimizing false negatives. Here, the best performing values are highlighted with bold font. According to the result exhibited in Table 4, it is evident that CNN could attain the tradeoff among all the performance metrics considering each of the datasets which makes the CNN model an efficient candidate for the rest of the study.

Table 4: Performance of the applied deep learning models on three datasets.

Dataset	Model	Precision	Recall	F1-score	FPR	FNR	Accuracy (%)
Datset-1 (Zeus)	CNN	**0.9952**	0.9871	**0.9911**	**0.0049**	0.0046	**99.53**
	MLP	0.9916	0.9757	0.9835	0.0078	0.0089	99.12
	LSTM	0.8416	0.6935	0.7058	0.0627	0.2541	77.54
	TabNet	0.9823	**0.9938**	0.988	0.0343	**0.0011**	99.37
Datset-2 (Emotet)	CNN	**0.9943**	**0.9978**	**0.996**	**0.0000**	0.0114	**99.68**
	MLP	0.9701	0.9883	0.9787	0.002	0.0063	0.983
	LSTM	0.9206	0.9377	0.9286	0.0273	0.1316	94.32
	TabNet	0.9798	0.9467	0.9616	0.0403	**0.0000**	96.99
Datset-3 (Trickbot)	CNN	**0.9988**	**0.9996**	**0.9992**	**0.0000**	0.0025	**99.93**
	MLP	0.9088	0.9105	0.9097	0.0468	0.01357	92.96
	LSTM	0.815	0.8231	0.8189	0.0882	0.2818	86.02
	TabNet	**0.9987**	0.9971	0.9979	0.0020	**0.0006**	99.84

3.2.2 Personalized Training and Implementing Ensemble CNN Model

In real life scenarios of data imbalance, model performance can be affected. To enhance model robustness, we pursue personalized training and create an ensemble model based on meticulous model evaluation, aiming for improved detection rates. According to the analysis in first stage, CNN is chosen for personalized training on the three datasets. Then three individual best performing model weights are saved from three separate experiments on the datasets. These models with best performing weights are further ensembled as illustrated in Fig. 3. Here, sequential, sequential_1, and sequential_2 depicts the three individual CNN models with their best weights. Then the features are concatenated followed by 2 dense layers to get the final ensemble model.

3.2.3 Incorporating Federated Learning with Ensemble Model

We incorporate federated learning approach at the final stage of model development where the ensemble CNN model is used as the global shared model that we got from the previous stage. Linguistic variations, technological disparities, and other factors can lead to the prevalence of specific malware types in distinct regions (Newsroom, 2017). In our experimental setup, each client, representing a geographically separated bank, simulates a unique strain of malware, showcasing the potential for localized malware vari-

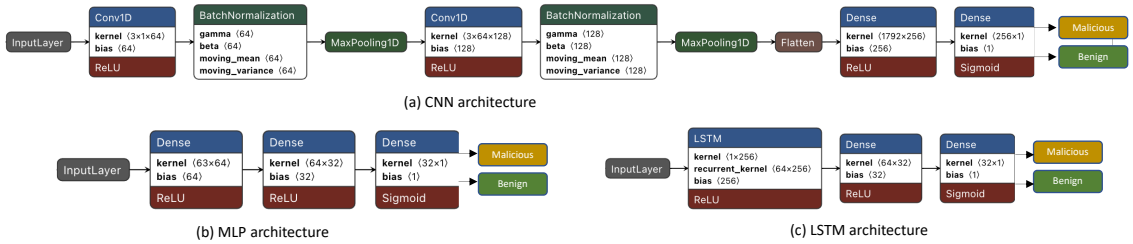

Figure 2: Architecture of the Deep learning models used in this study: (a) CNN model (b) MLP model (c) LSTM model.

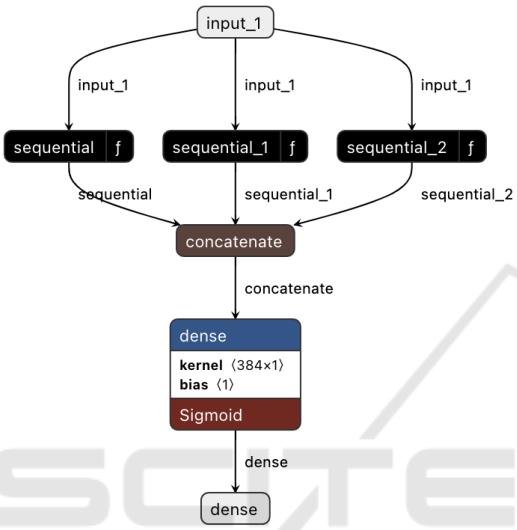

Figure 3: The global shared Ensemble CNN model constructed by stacking features from the three personalized-trained CNN models.

Table 5: Hyperparameter values for federated learning.

Hyperparameters	Value
Communication round	20
Number of local epoch	3
Initial Learning rate	0.001
Learning rate decay	0.1
Early stopping patienc	2
Optimizer	SGD
Train/test split	90%-10%
Federated averaging weighting	Client data size

ations. Fig. 4 depicts the whole functionality of this approach. The three datasets are distributed in three different clients and the central server holds the ensemble CNN model as the global shared model. The initial individual model weights (W_t^{c1}, W_t^{c2}, W_t^{c3}) are sent from the clients to the central server and there the server aggregates the model weights using FedAvg (McMahan et al., 2017) method. Then the updated global weights (W_t^g) are sent back to the clients. In this way, the model gets trained at every communication round between server and the clients. Additionally, in our approach, we put the condition that at every round when each client updates the model weights, it checks the previous round's accuracy (A_t^{cn}) that is stored locally. If the current accuracy (A_{t+1}^{cn}) is higher, the model weights are updated and stored, otherwise it keeps the previous weight. In this way, the model is always guided towards a better result, resulting in early convergence. Moreover, all the clients are trained with their own heterogeneous private data without any kind of sharing. The hyperparameter values for the proposed FL approach is depicted in Table 5 and these values have been chosen upon several experiments and insightful analysis.

4 RESULT ANALYSIS

The result analysis of this work is done considering the following aspects: firstly, we investigate the performance of the proposed method considering different performance metrics. Additionally, some experiments are done to inspect the impact of model ensemble and conditional weight update in the federated learning. Finally, the model is validated by testing with unseen multi-class data followed by a comparative analysis with respect to some state-of-the-art methods.

4.1 Evaluation of Proposed FL Based Scheme

To assess the effectiveness of the proposed Federated Learning (FL) based approach, we utilized various evaluation metrics, including precision, recall, F1 score, false positive rate (FPR), false negative rate (FNR), and accuracy. The definition and effectiveness of these metrics have already been mentioned in subsection 3.2, and Table 6 shows the result of all the clients based on these evaluation metrics using proposed FL approach. Here, we can observe that for each of the clients, the proposed approach attained quite promising result for both benign and mal-

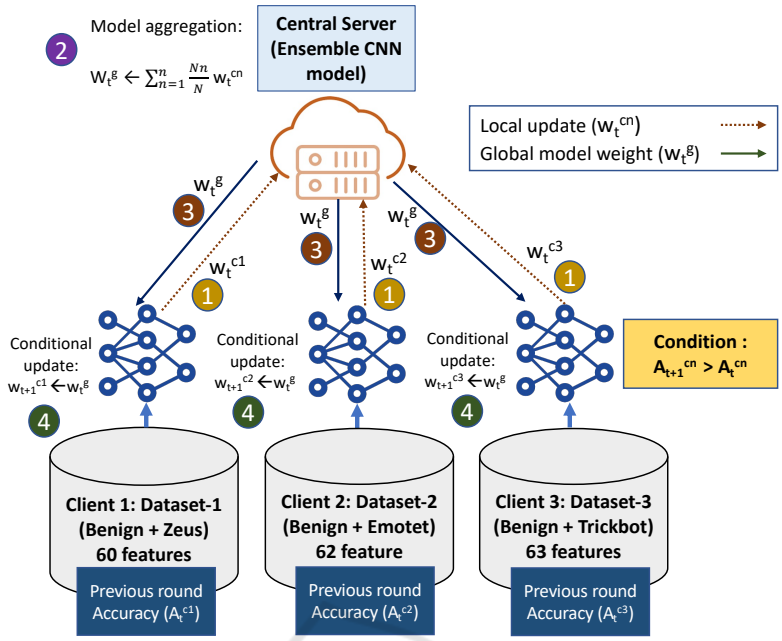

Figure 4: The proposed FL-based scheme utilizing the ensemble CNN as the global shared model for 3 clients with distinct datasets of varying malware types and features, alongside conditional model updates for clients.

ware class. The attained accuracy is 99.30%, 99.74%, 99.95% for client 1, client 2 and client 3 respectively. The confusion matrix for for client 1, client 2 and client 3 are depicted in Fig. 5(a), (b), and (c) respectively.

Table 6: Result obtained from the proposed federated learning approach.

Client	Malware	Precision	Recall	F1-score	FPR	FNR	Accuracy (%)
Client 1	Benign	0.9936	0.9982	0.9959	0.0098	0.0064	99.30
	Zeus	0.9902	0.9656	0.9777			
Client 2	Benign	0.9907	1.0000	0.9953	0.0000	0.0093	99.74
	Emotet	1.0000	0.9963	0.9982			
Client 3	Benign	0.9982	1.0000	0.9991	0.0000	0.0018	99.95
	Trickbot	1.0000	0.9993	0.9997			

4.2 Impact of Model Ensemble

In our study, the personalized trained three CNN ensemble models are used as the global shared model in the FL architecture. The performance of individual CNN models and the ensemble model on the FL approach is depicted in Fig. 6(a). In this figure, the first, second and third graph represent the impact of model ensemble on client 1, client 2, and client 3 respectively. From these graphs we can observe that ensemble model can maintain a good trade off among all performance metrics and attains a balanced performance considering each of the clients, whereas the individual models fail to maintain a balanced performance. For instance, CNN1 performed well for client 1 and client 3 but in case of client 2, the performance degraded. Similarly, CNN2 performed well for client 1 and the performance dropped in client 2 and client 3. Likewise, performance of CNN3 degraded in client 2, whereas the ensemble CNN outperformed all the models in this regard. This clearly depicts how the ensemble model contributes to a consistent performance across all clients.

4.3 Impact of Conditional Weight Update of Clients

In FL architecture, we consider conditional weight update for each of the clients as explained earlier. Fig. 6(b) illustrates the impact of this conditional weight update on each client. Here, in the first graph we observe that, due to conditional update the precision, recall, F1 score, and accuracy improved drastically for client 1. Similarly, the second and third client also attained significant improvement in the performance metrics. It is observed from the experimental result that the conditional weight update has significantly enhanced the overall detection rate of the proposed model (44.52% for client 1, 1.84% for client 2, and 10.84% for client 3).

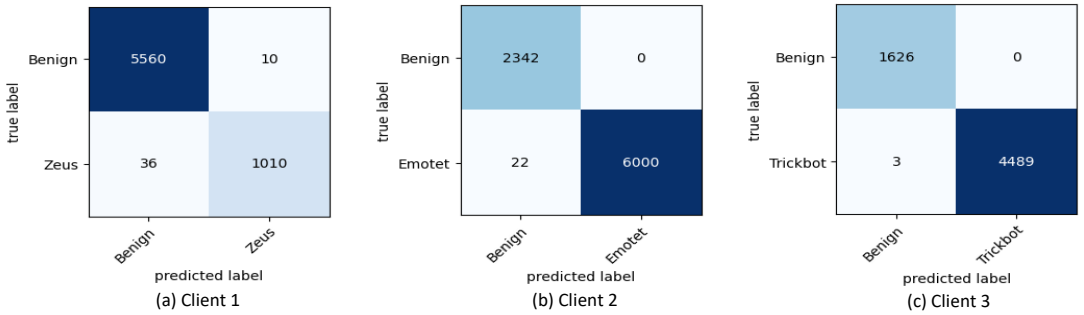

Figure 5: Confusion matrix for the proposed FL scheme: (a) Client 1 (b) Client 2 (c) Client 3.

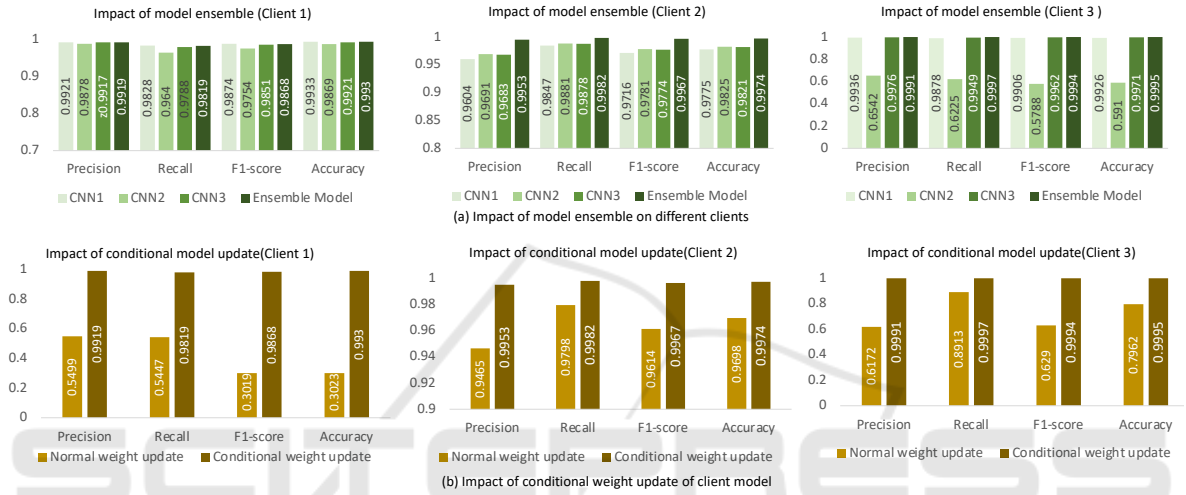

Figure 6: Evaluation of the proposed ensemble based deep FL approach on two aspect: (a) Impact of personalized CNN model ensemble (b) Impact of conditional updated of federated client models.

4.4 Model Validation with Unseen Multi-Class Data

To validate the robustness and generalization capability of the ensemble CNN model, we assess it with a multi-class dataset of unseen samples. This is different from the datasets used in the federated clients where each of the clients contains one type of malicious samples (zeus, emotet, or trickbot) along with the benign samples. For validation purpose, a dataset is considered that includes all three types of malicious samples along with benign samples. Here, initially Dataset-1, Dataset-2, and Dataset-3 are merged and used as training set and the new unseen dataset is used as the test set. This new dataset is prepared as the same way the other three datasets are prepared and collected from the Stratosphere website (Stratosphere, 2015). This includes total 63 features same as the Dataset-3 and has total 50,000 samples including 16256 benign samples, 3028 zeus samples, 17631 emotet samples, and 13085 trickbot samples. To depict the testing result, the confusion matrix is provided in Fig. 7 and also the result based on various performance metrics are presented in TABLE 7 from which we observe that the proposed model could attain an accuracy of 97.19% which validates its capability of detecting new unseen malware samples as well. Yet, due to the diversity in malware variants, we see degradation of precision and F1-score of zeus malware for the unseen dataset. The purpose of this experiment is to assess our model's performance if we use a new multi-class data with the same global shared model in the FL approach which demonstrates the future prospect of this model.

Table 7: Performance of the proposed ensemble model on unseen multi-class dataset.

Class label	Precision	Recall	F1-score	Accuracy (%)
Benign	0.9816	0.9751	0.9784	97.19
Zeus	0.8791	0.9129	0.8957	
Emotet	0.9802	0.9809	0.9806	
Trickbot	0.9701	0.9689	0.9695	

Table 8: Comparison with state-of-the-art malware detection techniques.

Ref.	Model	Dataset	Training Method	Privacy Preservation	Heterogeneity in Features	Personalized Training	Detection Rate (%)
(Kazi et al., 2019b)	Random Forest	Zeustracker	Centralized	✗	✗	✗	95.00
(Rawat et al., 2022)	Random Forest	Private data	Centralized	✗	✗	✗	99.89
(Agrafiotis et al., 2022)	CNN	CIC-IDS2017	Centralized	✗	✗	✗	99.70
(Fox and Boppana, 2022)	CNN	CIC17	Centralized	✗	✗	✗	90.80
		USTC16	Centralized	✗	✗	✗	99.90
		UTSA21	Centralized	✗	✗	✗	**1.00**
(Jiang et al., 2022)	GNN, CNN	AndroZoo	FL	✓	✗	✗	92.79
(Rey et al., 2022)	MLP, Autoencoder	N-BaIoT	FL	✓	✗	✗	–
(Hsu et al., 2020)	SVM	Private data	FL	✓	✓	✗	95.31
Proposed Method	Ensemble CNN	Stratosphere (2015)	FL	✓	✓	✓	99.32

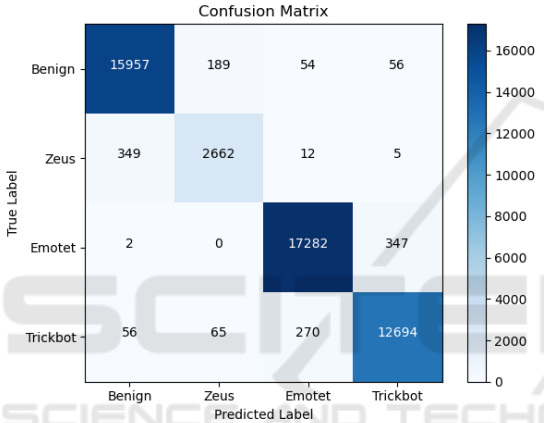

Figure 7: Confusion matrix for classifying unseen multi-class network flows.

4.5 Comparison with State-of-the-Art Methods

A comprehensive comparison of our proposed method denoted as 'Proposed Method' with recent state-of-the-art techniques on malware detection has been depicted in Table 8. The evaluation is represented across several critical dimensions such as training method, privacy preservation, handling feature heterogeneity, detection rate etc. The comparison showcases the strength of the proposed method in terms of the aforementioned criteria. Unlike classical DL and ML methods, a privacy preserving FL based approach with conditional weight update has been introduced in our work. Another fundamental challenge in real life applications remains in handling data heterogeneity in feature space that has been considered in our approach. To improve model's performance for all the clients, personalized training is conducted that consequently helped our model to converge across all the clients. Though the proposed method didn't attain the best detection rate as per Table 8, it consistently outperformed the state-of-the-art methods across all other evaluated dimensions.

5 DISCUSSION AND FUTURE DIRECTION

This study is dedicated to implementing a robust and data privacy-preserving banking malware detection system. The comprehensive analysis focuses on three main aspects: developing a robust detection method through model ensemble, ensuring data privacy using federated learning (FL), and enhancing the conventional FL-based scheme by addressing data heterogeneity among clients while maintaining strong detection performance. The proposed study demonstrates an effective approach to deal with data heterogeneity using an ensemble model as the global shared model. Furthermore, an enhanced performance is achieved by introducing conditional model update in federated clients. This analysis exhibits the impact of model ensemble and conditional weight update in overall performance of FL based approach while dealing with data heterogeneity.

However, while this study successfully addresses some existing challenges in FL-based malware detection, there are still several aspects that require future investigation. This includes exploring non-IID scenarios, optimizing communication overhead between server and clients, and considering real-time implementation for practical applications. Non-IID scenarios includes considering multiple varieties of malware types in different clients those are not necessarily identical. Communication overhead can be reduced in FL scheme by applying effective client selection

methods. Moreover, in practical cases, imbalanced datasets can affect model performance and such situations can be encountered by technique like Synthetic Minority Oversampling Technique (SMOTE). Additional sensitivity analysis can measure the extent of this impact and gain insights into model's behavior in real-world conditions. Zeus, emoted, and trickbot are few of the most prevalent malwares in banking sector those basically represent the overall malware attack scenario in this domain, which is the main reason for considering these strains, however, in future research, some other significant malware types could be considered. Additional future work may include analyzing the computational cost of our approach with newer datasets to enhance its scalability and efficiency. We intend to work on the aforementioned areas for further improvement of this study in future.

6 CONCLUSIONS

Detecting banking malware is of paramount importance in safeguarding financial systems and protecting user accounts. This paper presents an empirical analysis to construct a robust and privacy-preserving malware detection system specifically tailored for the financial sector. The approach combines deep ensemble learning and federated learning, utilizing three diverse datasets with distinct features. The experimentation involves selecting the best deep learning model (CNN) from four candidates (CNN, MLP, LSTM, and TabNet), followed by ensemble model construction and implementation in the federated learning approach. Notably, the proposed conditional update of client models effectively handles the heterogeneity of datasets, achieving a promising accuracy of 99.30%, 99.74%, and 99.95% for client 1, client 2, and client 3, respectively. The integration of ensemble CNN with the proposed FL architecture offers a promising solution for an effective banking malware detection system.

ACKNOWLEDGEMENTS

Part of this study was funded by the ICSCoE Core Human Resources Development Program and MEXT Scholarship, Japan.

REFERENCES

Agrafiotis, G., Makri, E., Flionis, I., Lalas, A., Votis, K., and Tzovaras, D. (2022). Image-based neural network models for malware traffic classification using pcap to picture conversion. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–7.

An, W., Han, Y., Liu, S., An, B., Tao, T., and Liu, J. Malware https traffic identification based on convolutional neural network and autoencoder. *Available at SSRN 4302957*.

Arik, S. O. and Pfister, T. (2020). Tabnet: Attentive interpretable tabular learning.

Fox, G. and Boppana, R. V. (2022). Detection of malicious network flows with low preprocessing overhead. *Network*, 2(4):628–642.

Gezer, A., Warner, G., Wilson, C., and Shrestha, P. (2019). A flow-based approach for trickbot banking trojan detection. *Computers & Security*, 84:179–192.

Hsu, R.-H., Wang, Y.-C., Fan, C.-I., Sun, B., Ban, T., Takahashi, T., Wu, T.-W., and Kao, S.-W. (2020). A privacy-preserving federated learning system for android malware detection based on edge computing. In *2020 15th Asia Joint Conference on Information Security (AsiaJCIS)*, pages 128–136. IEEE.

Jiang, C., Yin, K., Xia, C., and Huang, W. (2022). Fedhgcdroid: An adaptive multi-dimensional federated learning for privacy-preserving android malware classification. *Entropy*, 24(7):919.

Kazi, M. A., Woodhead, S., and Gan, D. (2019a). Comparing and analysing binary classification algorithms when used to detect the zeus malware. In *2019 Sixth HCT Information Technology Trends (ITT)*, pages 6–11. IEEE.

Kazi, M. A., Woodhead, S., and Gan, D. (2019b). Detecting the zeus banking malware using the random forest binary classification algorithm and a manual feature selection process. In *International Symposium on Security in Computing and Communication*, pages 286–297. Springer.

Liu, S., Han, Y., Hu, Y., and Tan, Q. (2021). Fa-net: Attention-based fusion network for malware https traffic classification. In *2021 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7. IEEE.

Lu, K., Cheng, J., and Yan, A. (2023). Malware detection based on the feature selection of a correlation information decision matrix. *Mathematics*, 11(4):961.

Mahdavifar, S., Kadir, A. F. A., Fatemi, R., Alhadidi, D., and Ghorbani, A. A. (2020). 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)*, pages 515–522. IEEE.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Mohaisen, A. and Alrawi, O. (2013). Unveiling zeus. *arXiv preprint arXiv:1303.7012*.

Newsroom, P. (2017). How Geography Affects Malware Threats Differently. https://www.psafe.com/en/blog/geography-affects-malware-threats-differently/. [Online; accessed 15-August-2023].

Rawat, R., Rimal, Y. N., William, P., Dahima, S., Gupta, S., and Sankaran, K. S. (2022). Malware threat affecting financial organization analysis using machine learning approach. *International Journal of Information Technology and Web Engineering (IJITWE)*, 17(1):1–20.

Rey, V., Sánchez, P. M. S., Celdrán, A. H., and Bovet, G. (2022). Federated learning for malware detection in iot devices. *Computer Networks*, 204:108693.

Stratosphere (2015). Stratosphere laboratory datasets. Retrieved June 13, 2023, from https://www.stratosphereips.org/datasets-overview.

Wang, W., Zhu, M., Zeng, X., Ye, X., and Sheng, Y. (2017). Malware traffic classification using convolutional neural network for representation learning. In *2017 International conference on information networking (ICOIN)*, pages 712–717. IEEE.

Wang, Z., Fok, K. W., and Thing, V. L. (2022). Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study. *Computers & Security*, 113:102542.