

A Horizontal Federated-Learning Model for Detecting Abnormal Traffic Generated by Malware in IoT Networks

Phuc Hao Do^{*,***}, Tran Duc Le^{**}, Vladimir Vishnevsky^{****}, Aleksandr Berezkin^{*} and Ruslan Kirichek^{*,****}

^{*}The Bonch-Bruевич Saint-Petersburg State University of Telecommunications, Saint-Petersburg, Russia

^{**}University of Science and Technology – The University of Danang, Da Nang, Viet Nam

^{***}Danang Architecture University, Da Nang, Viet Nam

^{****}V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia

haodp@dau.edu.vn, letranduc@dut.udn.vn, vishn@inbox.ru, pcdreams@mail.ru, kirichek@sut.ru

Abstract— To detect botnet attack traffics in the Internet of Things (IoT) networks, Deep Learning (DL) has received much attention. However, the conventional centralized deep learning technique cannot be utilized to identify a botnet attack and ensure user data privacy simultaneously due to the centralized dataset characteristic. This study addresses the IoT malware traffic detection opportunities afforded by Federated Learning and examines the inherent security concerns of this new learning paradigm. The framework has been evaluated using N-BaIoT, a dataset that collects the network traffic of many real IoT devices infected by malware. In this paper, we compare the effectiveness of the Federated Learning model using famous models such as Convolutional Neural Network, Long Short-Term Memory, and Gated Recurrent Unit with a centralized approach. The results show that Federated Learning achieves high performance and can be used to detect abnormal traffic in IoT. In addition, we experimentally point out that CNN is the model that gives the best results among the three evaluated models.

Keywords— IoT, abnormal traffics, malware detection, federated learning, AI model

I. INTRODUCTION

As a result of the rapid growth of the Internet of Things (IoT) technology, IoTs devices have become an integral part of people's daily life. However, it inevitably introduces some network security challenges [1][2]. IoT devices are easy targets for malicious attacks such as malware attacks due to their heterogeneity and vulnerability. The prevalence of privacy and security concerns is rising due to the continual expansion of IoT devices and the resulting exposure of more and more private data online. It is essential to monitor IoT networks to prevent malicious cyberattacks on IoT devices [3]. By studying the traffic of IoT devices, network intrusion detection in the IoT ecosystem may be enhanced, and cyberspace security can be guaranteed [4][5][6].

In recent years, the fast expansion of deep learning has significantly aided the advancement of Internet of Things intrusion detection research. Deep neural networks are used in this approach to automatically identify the features in the dataset without the need for human selection. Thereby it

reduces the feature engineering workload and improves the data processing performance. The Internet anomaly detection approach cannot be immediately and easily applied to the IoT due to the requirement for large amounts of data. Most existing machine learning or artificial intelligence solutions rely on a single server responsible for gathering data from various IoT devices and building global models. This strategy is ineffective, nevertheless, in situations when device actions involve sensitive or private information (as in IoT networks) that might seriously impact environmental security and privacy in the event of disclosure to unauthorized parties.

Federated Learning (FL) [7] is becoming increasingly relevant in this context where information privacy and integrity are essential. In FL, the training model is decentralized by several nodes or clients using local data. Each decentralized node trains a distinct model on its data in this paradigm before distributing the model parameters (not the private data) to the others. Through a central entity known as a server or by using a peer-to-peer methodology, the interchange of the model parameters and their aggregation to produce a singular and global model may be carried out [8]. Following several iterations, each client obtains a global model by aggregating their unique models. Because data is not shared with any external identities, this strategy naturally supports data privacy.

In this paper, we build a horizontal Federated-Learning model based on some famous models such as Convolutional Neural Network (CNN), Long short-term memory (LSTM), and Gated Recurrent Unit (GRU), to detect abnormal traffics, especially DDoS attack traffic generating by malware such as Botnet in IoT networks.

We use the N-BaIoT [9] dataset for model training because it is collected from real devices instead of simulating network traffic like other datasets. The objective of this study is to evaluate the performance of FL compared with the traditional centralized artificial intelligence model. Some research questions: Can FL detect malware or malicious traffic in IoT networks? What training models are appropriate when applied to this decentralized approach? Therefore, our main contributions are:

- Using feature selection and feature engineering techniques to clean the N-BaIoT dataset and select features that have the most significant impact on model accuracy and performance;
- Comparing the effectiveness of the FL model with traditional deep learning model with the same dataset, thereby evaluating the applicability of FL to the goal of identifying abnormal traffics in IoT networks;
- Deploying and comparing deep learning models such as CNN, LSTM, and GRU in the FL model to assess compatibility and select the suitable model for the FL approach.

The results of this study will be included in the Draft Recommendation ITU-T Q.TSRT_IoT “Test specifications for remote testing of Internet of Things using the probes”.

II. RELATED WORKS

Several researchers have conducted a thorough investigation to find a more trustworthy anomaly detection strategy for IoT networks. To solve the issue of constrained device resources, the study in [10] created a unique, lightweight attack detection method that uses the Support Vector Machines (SVM) algorithm to recognize hazardous assaults in the IoT space. To offer a distributed intrusion detection system, Ferdowsi et al. recommended anomaly detection utilizing distributed Generative Adversarial Networks (GANs) [11]. Each device may protect user privacy while applying a detection model to its data since it is not maintained centrally. A centralized IoT intrusion detection system based on fog computing was suggested in the publication [12]. Two cascading recurrent neural networks (RNNs) are used in the design, each with its hyperparameters and attack-specific tuning. The system administrator is notified if any of the two RNNs determines that an input instance is malicious. Time-series data from IIoT sensors were utilized by the authors of [13] to identify abnormalities using an attention-based convolutional neural network with LSTM [14].

In recent years, FL has risen to prominence in cybersecurity. This IoT security paradigm has already been used in several works. In this context, the study proposal [15] underlined the problem with traditional AI-based solutions' lack of data privacy. However, a private dataset was used for the examination. Despite sharing similar objectives, the research mentioned in [16, 17] mainly focused on industrial IoT devices. They looked at application samples and sensor readings rather than network data.

The authors introduced the Federated Averaging (FedAVG) approach in [18], which is a strong basis for many FL-based investigations. In this strategy, a central server coordinates the training of a global model across several clients using their datasets. The server is responsible for averaging the client-sent models' parameters and sending the resulting global model back to the clients. Until a termination condition is met, this process is repeated. FL has significantly advanced since several

researchers [19] examined the most current advancements in this area.

Two more robust model aggregation functions were suggested in [20]. They are based explicitly on the coordinate-wise mean and median of the clients' models supplied to the server. The authors of [21] suggested resampling as a way to lessen the variability in the distribution of the models that the clients supplied. It should be used before employing a powerful aggregating function. When used with models trained on non-IID datasets, it seeks to lessen any negative consequences that such a function could have.

III. THE HORIZONTAL FEDERATED-LEARNING

Horizontal Federated Learning is called Homogenous Federated Learning or Sample-based Federated Learning. It is introduced in the scenarios where data sets share the same feature space but differ in the sample (other rows). The use of this model is suitable for the fact that we use a single dataset, which is N-BaIoT. Thus the features are the same.

Figure 1 displays the approach used in this study, which consists of the dataset, data processing, feature selection, and classifier. Standardization and minimum-maximum normalization (min-max normalization) are also used as data preparation techniques.

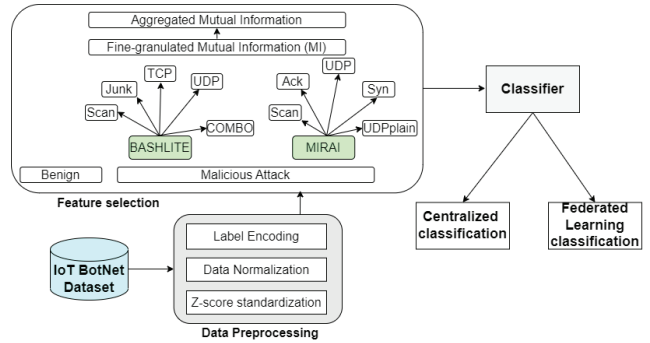


Figure 1. The research flow

A. Dataset

There are many public datasets related to IoT security as follows:

- *N-BaIoT*¹ [9]: This dataset tackles the shortage of available botnet datasets, particularly for the Internet of Things. It implies actual traffic data, collected from nine commercial IoT devices that were actually infected with *Bashlite* and *Mirai* malwares.
- *MedBIoT*²: It is collected from actual and simulated Internet of Things devices in a medium-sized network (i.e., 83 devices). The data collection is divided into categories based on the kind of traffic (malware or normal traffic), making it simple to label the data and extract characteristics from the raw pcap files.

¹ <https://www.kaggle.com/datasets/mkashifn/nbaiot-dataset>

² <https://cs.taltech.ee/research/data/medbiot/>

- *Bot-IoT*³: The BoT-IoT dataset was developed using a realistic network environment constructed at the Cyber Range Lab of the UNSW Canberra Cyber Center. The dataset consists of Service Scan, DDoS, DoS, and Keylogging, and Data exfiltration attacks. In addition, DDoS and DoS attacks are categorized based on the protocol employed.
- *TON-IoT*⁴: This dataset contains heterogeneous data sources gathered from Telemetry datasets of IoT and IIoT sensors, Windows OS, Ubuntu, and network traffic datasets. The datasets were acquired from the Cyber Range and IoT Labs' realistic and wide network.
- *IoT-23*⁵: IoT-23 contains 20 malware captures (scenarios) conducted on IoT devices, and 3 benign IoT traffic captures. The authors ran a particular malware sample on a Raspberry Pi for each malicious scenario, which utilized multiple protocols and carried out distinct tasks.

The N-BaIoT dataset is collected separately for each device, so it is very suitable to implement the FL empirical model. Therefore, we will use it for our research. IoT devices in the N-BaIoT dataset were targeted by the Bashlite and Mirai botnet attack families. Each file has 115 features as well as a class label. The dataset has also been built to support binary and multi-class classification. The target class labels are: "TCP attack," "benign" for detection, and "Mirai" or "Bashlite" attack types for multi-class classification.

In this study, we will focus on multi-classification. We use part of the dataset to proceed with the experiment. The multi-class classification will be evaluated.

B. Preprocessing

Although data preparation is difficult and time-consuming [22], its importance has been demonstrated for speeding the training process and enhancing its effectiveness. Consequently, this study employs label encoding, min-max normalization, and standardization as pre-preprocessing procedures.

1) Label Encoding

The class label has 11 different category values (one "Benign" class and ten subclasses of attack type). As a result, before these attributes are used with the models, they are converted into numerical values. There are several methods for converting categorical values, including one-hot encoding [23], ordinal encoding [24], similarity encoding [25], entity embedding [26], and multi-hot encoding [27]. Among them, one-hot encoding and ordinal encoding are the most widely employed. This study uses the one-hot encoding technique for encoding categorical values.

2) Normalization and Standardization

Suppose the columns in a dataset contain values with varying ranges. In that case, the performance of both regression and classification models is negatively impacted. Mahfouz et al. in [28] demonstrated how this issue causes

the performance of the models to decrease when uneven scales of features are observed in a dataset. It is required to determine the acceptable range for the insignificant and dominating values to address such problems. The two most used methods are min-max normalizing and z-score standardization:

- The following equation is used to apply min-max normalization to change the values of the dataset's feature values into the range [0, 1]:

$$X_{normalized} = \frac{X - X_{min_value}}{X_{max_value} - X_{min_value}} \quad (1)$$

where $X_{normalized}$ represents the normalized value, X_{min_value} and X_{max_value} are the intended interval's boundary range, which is [0, 1], and X is the initial value that would be altered inside those ranges.

- Z-score standardization is used to rescale dataset features, reflecting the characteristics of a normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$.

$$X_{normalized} = \frac{X - \mu}{\sigma} \quad (2)$$

3) Feature Selection Techniques

After encoding the target class, the "benign" class was added to the N-BaIoT dataset, which has 115 characteristics and 10 class labels. To improve the performance of the classification process, we can implement some feature selection methods to choose the best features as input for the model training process instead of using all the features. There are many typical selection methods [31], such as PCA Principal Component Analysis (PCA), Mutual Information (MI), or Analysis Of Variance (ANOVA). In the scope of the paper, we choose the MI method for the feature selection process of the input data.

An aggregated MI with multiple rank aggregation functions is developed and evaluated for the multiclass dataset. The concept of aggregated MI is explained as follows:

- Calculate the information gain score for each feature, f_i , in dataset D relative to class type $c \in C$. The features are then ranked based on the aggregator functions listed below. Calculate the information gain score for each feature, f_i , in dataset D relative to class type $c \in C$. After that, the features are ranked based on the aggregation methods (Min, Max, Mean). Only part of preserved features are supplied to the classifiers, and the total performance is assessed.
- List of aggregators:
 - *Min*: Chooses the class type c_i as the target class and takes the relevance score with the lowest value.

³ <https://iecc-dataport.org/documents/bot-iot-dataset>

⁴ <https://research.unsw.edu.au/projects/toniot-datasets>

⁵ <https://www.stratosphereips.org/datasets-iot23>

- *Max*: Chooses the class type c_i as the target class and takes the relevance score with the highest value.
- *Mean*: Chooses the class type c_i as the target class and takes the relevance score with the mean value.

C. Deep Learning Techniques Applied to Abnormal Traffic Detection

1) Convolutional Neural Networks (CNN)

A CNN algorithm for deep learning is utilized to construct an effective system for picture categorization. The CNN model can also support the creation of effective security-related strategies. CNN has four layers: input, convolutional, pooling, and fully connected.

- **Convolutional Layer**: The training sample is examined, sized, and filtered using the convolutional layer, which applies various convolution kernel filters. The weight matrix for the input sample is created by the convolutional layer, which also recodes the weighted summation kernel layer. The system's performance may be enhanced by selecting the optimal values for crucial hyperparameters to reduce the complexity of the neural network.
- **Pooling Layer**: A pooling layer is utilized to minimize the number of parameters in a map of features by picking the maximum values in each region to create a fit matrix average pooling. This matrix is transformed into the subsequent layer.
- **Fully Connected Layer**: it is the final layer of the CNN networks.

2) Long Short-Term Memory (LSTM)

LSTM [29] is a recurrent neural network (RNN) that uses feedback to remember portions of the input and make predictions. RNNs are intended to process sequential input and have thus found widespread use in speech recognition and machine translation. The well-known problem of vanishing gradients affects the long-term memory of conventional RNNs. It restricts their ability to make predictions based on the most recent data in the sequence. LSTM (Figure 2) overcomes the problem of vanishing gradients and can thus handle longer sequences (long-term memory). LSTM can extract context from a succession of features. Using a gate function, it can add or delete information from the hidden state vector, preserving vital information in the hidden layer vectors. There are three gate functions in LSTM: the forget, the input, and the output. The forget gate is used to regulate how much information from C_{t-1} is maintained throughout the computation of C_t , and i_t the forget vector may be written as follows:

$$f_t = \sigma(U^f x_t + W^f h_{t-1} + b_f) \quad (3)$$

where U^f , W^f , and b_f are the forget gate's parameters, x_t is the input vector in step t , and h_{t-1} is the hidden state vector

in the previous step. The input gate determines the amount of x_t information added to C_t and may be stated as follows:

$$f_t = \sigma(U^f x_t + W^f h_{t-1} + b_f) \quad (4)$$

where U^i , W^i , and b_i are the input gate's parameters and C_t may be determined by using both the input gate's vector i_t and the forget gate's vector f_t as shown below:

$$f_t = \sigma(U^f x_t + W^f h_{t-1} + b_f) \quad (5)$$

Where $\tilde{C}_t = \tanh(U^c x_t + W^c h_{t-1} + b_c)$ reflected the information represented by the vector of the hidden layer. Note that $*$ represents the Hadamard product (element-wise). The output gate regulates the output in C_t , and:

$$o_t = \sigma(U^o x_t + W^o h_{t-1} + b_o), h_t = o_t * \tanh(C_t) \quad (6)$$

where U^o , W^o , and b_o are the output gate parameters and C_t the internal state at time step t .

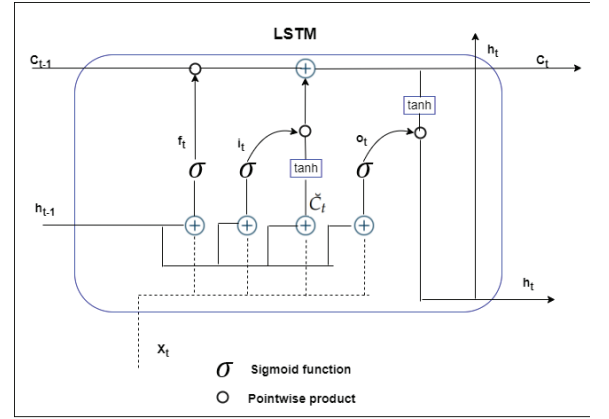


Figure 2. LSTM model

3) Gated Recurrent Unit (GRU)

A GRU model [30] (Figure 3) is also an RNN and LSTM variant. However, GRU has only two gates: the update and the reset gates. Due to its simplicity and ease of training, GRU is superior to LSTM, which data transmitted to the output is decided by the gates, which are two vectors. The update gate helps the model to calculate how much information from the past must be transmitted to the future. For step t , the update gate Z_t is computed using the following formula:

$$z_t = \sigma(U^z x_t + W^z h_{t-1}) \quad (7)$$

where U^z , W^z are the weights of the update gate and h_{t-1} stores data for the previous $(t-1)$ units. The reset gate is used to determine how much of the past data to forget, which is computed using the following formulas:

$$r_t = \sigma(U^r x_t + W^r h_{t-1}) \quad (8)$$

where U^r , W^r are the reset gate's weights and h_{t-1} holds information for the previous $(t-1)$ units. Using the reset gate, the current memory content will preserve the essential information from the past:

$$c_t = \tanh(U^c x_t + r_t * W^c h_{t-1}) \quad (9)$$

where U^c , W^c are the weights. Note that $*$ represents the Hadamard product (elementwise). The final step involves calculating the vector h_t that contains the information for the current unit and transmits it along the network.:

$$h_t = z_t * h_{t-1} + (1 - z_t) * c_t \quad (10)$$

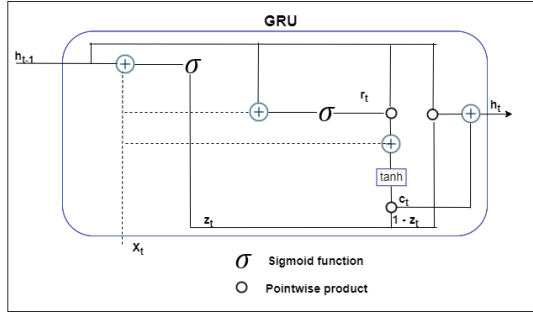


Figure 3. GRU model

A GRU network acquires a long spatial (or temporal) sequence with less computational complexity than a conventional encoder-decoder. GRU can handle longer sequences than regular RNNs because it can solve the vanishing gradient problem with gating techniques. To build context between features spread out across a large area or assess the degree of interconnections between features, GRU and LSTM may be used for sequences of spatial data.

4) Experiment Flow for Centralized Approach

We should recall that we use the N-BaIoT dataset, in which abnormal traffic is caused by malware, especially Botnets. Therefore, here we use “Botnet” as a general concept to indicate the source of abnormal traffic.

Figure 4 shows the operational flow of the experimental process of Botnet detection in the centralized approach. Data is centralized in one place. First, we perform data processing and feature selection from the dataset data to process the data and select the attributes that are good enough to include in the training model. The dataset is divided into 70% for training, 10% for validation, and 20% for testing.

After that, we will apply three models, CNN, LSTM, and GRU, to detect abnormal traffic generated by Botnet. The results are obtained, evaluated, and compared between these models.

D. Federated learning Model

This section describes the architecture of the FL approach. We will present the components and how they interact during the training and testing of models. In addition, it illustrates how the framework is implemented for our validation use case, which uses the N-BaIoT dataset. Figure 5 depicts the framework architecture, which consists of K clients, each of which owns data from a single device and a server that coordinates the FL process. Figure 6 shows the back-and-forth interaction between the client-server and the preprocessing and feature selection process.

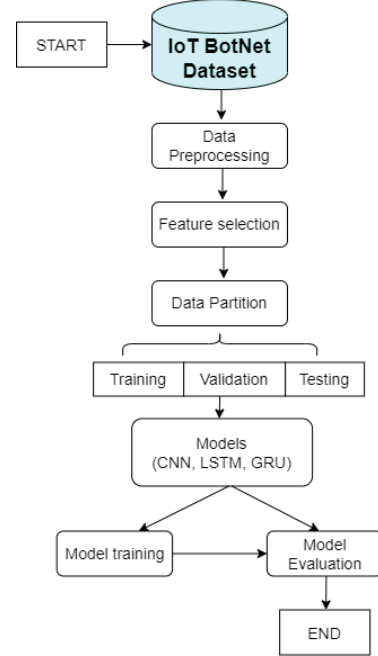


Figure 4. Experiment Flow for Centralized Approach

1) Model training component

The component aims to train the federated learning model that will be utilized for abnormal traffics detection.

Two FL techniques derived from the well-known FedAVG, Mini-batch aggregation and Multi-epoch aggregation, are considered. The primary distinction between FedAVG and the other algorithms is that FedAVG considers the aggregation function as a parameter. As a result, the server can experiment with aggregating methods other than average.

In this paper, we will implement the multi-epoch aggregation method. The model is trained for all E epochs simultaneously before being transmitted to the server for multi-epoch aggregation. According to [7], averaging models may produce arbitrarily poor results due to the objective's non-convexity. This issue is significantly more likely to arise with multi-epoch aggregation since the models are trained individually for a much more extended time before being combined. To minimize this issue, multi-epoch aggregation training is performed for $T = 30$ rounds with a decreasing learning rate.

2) Server component

The server is responsible for organizing the clients' training activities. In particular, it initializes the model from scratch. It compiles the models that the clients send into a so-called global model. The server is responsible for initializing the initial model's weights. All clients will have access to the first model when it is finished, and training may begin. It is important to note that every client starts with the same model. The server must combine the updated parameters from each client after receiving them to create the new global model parameters.

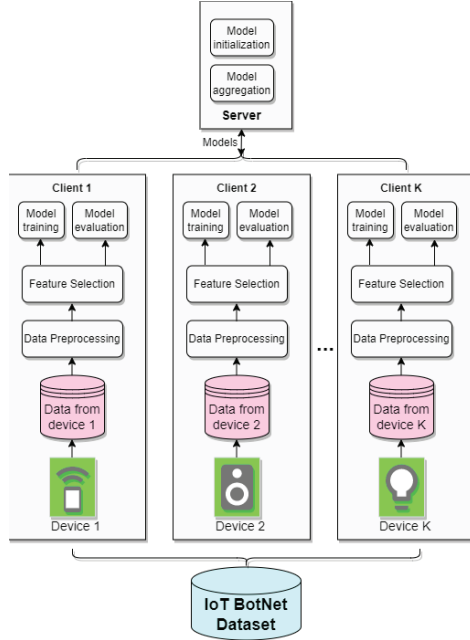


Figure 5. FL framework architecture and its components

E. Evaluations

Using the N-BaIoT dataset, this experiment aims to evaluate how well our approach discovers abnormal IoT traffic caused by malware. It is crucial to compare the federated learning technique with conventional solutions to ensure that it is appropriate for IoT networks [32]. More specifically:

- For a centralized approach: the training data is centralized on the server. The server then conducts the training and testing of the model with that centralized dataset. It does not preserve privacy. We compare the results obtained from training and testing three well-known models: CNN, LSTM, and GRU.
- For Federated Learning with Multi-epoch aggregation: we also apply three models, CNN, LSTM, and GRU, for this approach.

Figure 7 shows the accuracy of the centralized model. Figures 7a, 7b, and 7c correspond to the CNN, GRU, and LSTM algorithms. For CNN, in Fig. 7a, we see that from epoch = 5, this algorithm starts to converge and achieves accuracy = 90.72%. When using the GRU algorithm, the results in Fig. 7b

show that the training model only reaches convergence from epoch = 20 onwards and accuracy = 89.35%. Fig. 7c demonstrates the accuracy of the LSTM model with centralized data. We see that the convergence of the LSTM algorithm for the N-BaIoT dataset is not good, and the accuracy is relatively low. This shows that for this N-BaIoT dataset, the CNN algorithm has the best results compared to the rest of the algorithms.

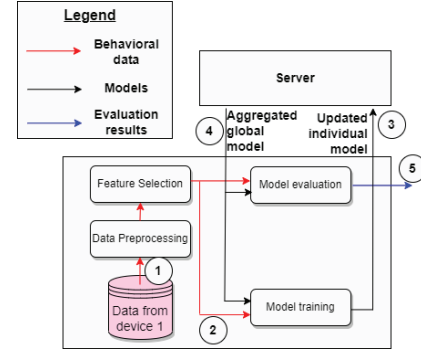


Figure 6. Detailed view of the training process in the FL approach

Figure 8 shows the loss value when training the model with centralized data corresponding to CNN, GRU, and LSTM algorithms. The results show that the CNN model has better results than the remaining models.

Fig. 8a represents the loss value when applying the CNN algorithm. From the 5th epoch, the loss value gradually converges and stabilizes, $\text{loss_val} = 0.14$. For GRU, the results are shown in Figure 8b with $\text{loss_val} = 0.2$. When deployed with the LSTM algorithm, the results of the loss value have not converged, and the fluctuation is very high when experimenting with the number of epochs = 30.

The Federated Learning approach's results of model training with distributed data on each client are shown in Figure 9.

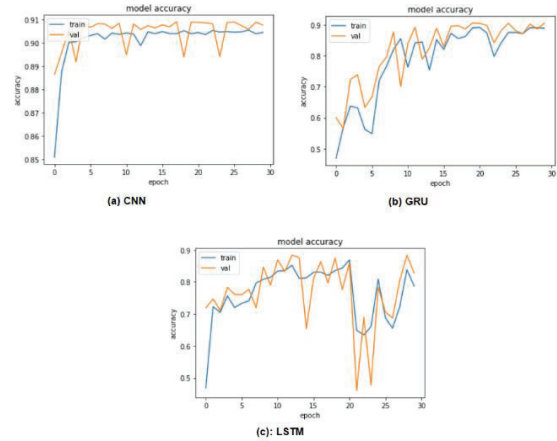


Figure 7. The performance of the centralized learning model

In Fig. 9a, we see that the accuracy when using CNN as a training model is relatively high, with accuracy = 90.41%. This

result converges very early from around round = 5. However, if deployed with GRU and LSTM models, until round = 25, these models are converged, and the accuracy result of LSTM is 88.71%, and for the GRU model is 85.67%.

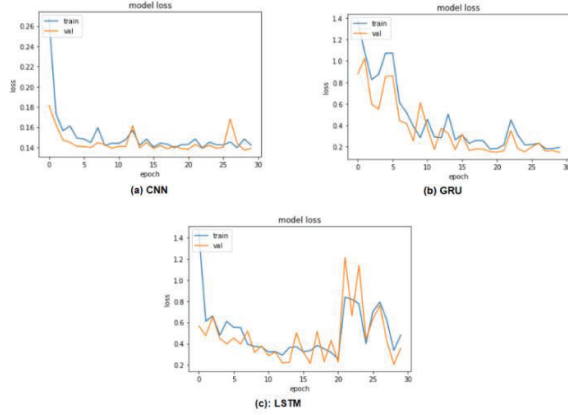


Figure 8. Loss value of centralized learning model

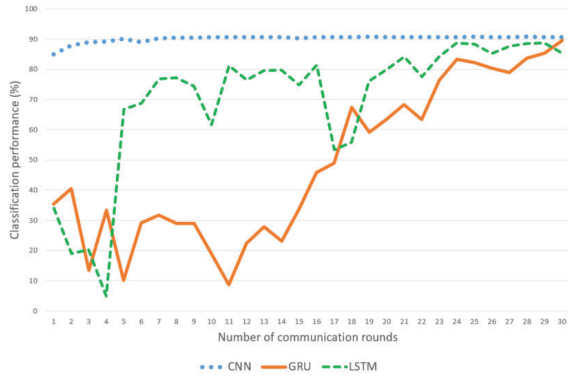


Figure 9. Classification performance of the FL model

Figure 10 shows the loss value when training distributed data. The CNN model gives the smallest loss value, and the performance of this model is better than the rest. For using the CNN model as a training model, the loss_val converges very quickly and reaches a value of loss_val = 1.63. In contrast, for GRU and LSTM algorithms, until round = 25, the result of the new model converged, and the loss_val value of the GRU is 1.72, and the loss_val value of the LSTM is 1.68.

Observing the accuracy of the models when deployed for centralized and distributed data, we see that the accuracy does not differ too much. The value fluctuates around 1% for accuracy. However, with centralized data, data privacy cannot be guaranteed. So in the case of ensuring data privacy, especially in IoT networks, the Federated learning model will respond better.

Figure 11 compares the training times of the centralized and FL-based approaches. The results show that the training time of the model when the data is concentrated is much better.

Obviously, the distributed data training adds transmission, computation, and configuration update time for all clients so that the execution time will be longer.

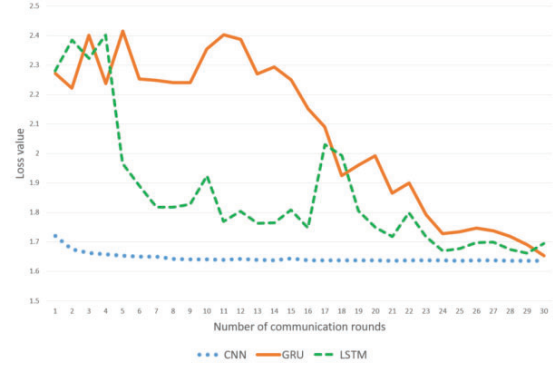


Figure 10. Loss value of federated learning model over the round

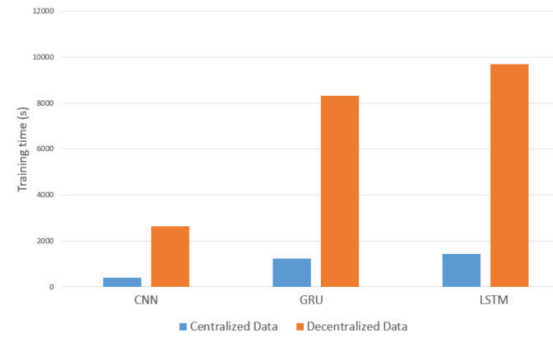


Figure 11. Time training between centralized data and decentralized data.

IV. CONCLUSIONS

The study suggests using the Federated-Learning approach for IoT malware abnormal traffic detection and protecting user privacy. We looked into the effectiveness of two different approaches using the N-BaIoT dataset: (i) federated learning, where each device owner trains a separate model that is regularly aggregated on a server, and (ii) a centralized approach, where each device owner trains a single, isolated model with the centralized dataset. The results show that although the centralized approach still gives higher accuracy than Federated Learning in detecting abnormal IoT traffic, this difference is not much. However, the centralized approach model will not guarantee the security and integrity of private information as in the FL model. In addition, when comparing three models CNN, LSTM, and GRU, applied in the FL approach, we found that CNN gives the best results and is suitable for simple FL models.

In future works, we plan to optimize the FL-model, especially the global parameter and improve the processing time to get better results.

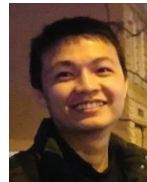
ACKNOWLEDGMENT

The study was financially supported by the Russian Science Foundation within of scientific project No. 22-49-

02023 "Development and study of methods for obtaining the reliability of tethered high-altitude unmanned telecommunication platforms of a new generation.

REFERENCES

- [1] Fan, Xiaochen, et al. "BuildSenSys: Reusing building sensing data for traffic prediction with cross-domain learning." *IEEE Transactions on Mobile Computing* 20.6 (2020): 2154-2171.
- [2] Alrawi, Omar, et al. "Sok: Security evaluation of home-based iot deployments." 2019 IEEE symposium on security and privacy (sp). IEEE, 2019.
- [3] Kulik, V., & Kirichek, R. (2018, November). The heterogeneous gateways in the industrial internet of things. In 2018 10th International congress on ultra modern telecommunications and control systems and workshops (ICUMT) (pp. 1-5). IEEE.
- [4] Kumar, J. Sathish, and Dhiren R. Patel. "A survey on internet of things: Security and privacy issues." *International Journal of Computer Applications* 90.11 (2014).
- [5] Butun, Ismail, Patrik Österberg, and Houbing Song. "Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures." *IEEE Communications Surveys & Tutorials* 22.1 (2019): 616-644.
- [6] Radhakrishnan, Divya. "Internet of Things: Privacy and Security Issues: A qualitative study about the Internet of Things and
- [7] Kairouz, Peter, et al. "Advances and open problems in federated learning." *Foundations and Trends® in Machine Learning* 14.1–2 (2021): 1-210.
- [8] Otoum, Safa, Ismaeel Al Ridhawi, and Hussein T. Mouftah. "Blockchain-supported federated learning for trustworthy vehicular networks." *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020.
- [9] Meidan, Yair, et al. "N-baiot—network-based detection of iot botnet attacks using deep autoencoders." *IEEE Pervasive Computing* 17.3 (2018): 12-22.
- [10] Arbex, Gustavo Vitral, et al. "IoT DDoS Detection Based on Stream Learning." 2021 12th International Conference on Network of the Future (NoF). IEEE, 2021.
- [11] Ferdowsi, Aidin, and Walid Saad. "Generative adversarial networks for distributed intrusion detection in the internet of things." 2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019.
- [12] Almiani, Muder, et al. "Deep recurrent neural network for IoT intrusion detection system." *Simulation Modelling Practice and Theory* 101 (2020): 102031.
- [13] Xia, Qinyu, Shi Dong, and Tao Peng. "An Abnormal Traffic Detection Method for IoT Devices Based on Federated Learning and Depthwise Separable Convolutional Neural Networks." 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC). IEEE, 2022.
- [14] Sim, Khe Chai, et al. "Domain Adaptation Using Factorized Hidden Layer for Robust Automatic Speech Recognition." *Interspeech*. 2018.
- [15] Almiani, Muder, et al. "Deep recurrent neural network for IoT intrusion detection system." *Simulation Modelling Practice and Theory* 101 (2020): 102031.
- [16] Vladimirov, Sergey, and Ruslan Kirichek. "The IoT identification procedure based on the degraded flash memory sector." *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Springer, Cham, 2017. 66-74.
- [17] Lou, Yang, et al. "Predicting network controllability robustness: A convolutional neural network approach." *IEEE Transactions on Cybernetics* (2020). Hussain, Fatima, et al. "Machine learning in IoT security: Current solutions and future challenges." *IEEE Communications Surveys & Tutorials* 22.3 (2020): 1686-1721.
- [18] Amanullah, Mohamed Ahzam, et al. "Deep learning and big data technologies for IoT security." *Computer Communications* 151 (2020): 495-517.
- [19] Sim, Khe Chai, et al. "Domain Adaptation Using Factorized Hidden Layer for Robust Automatic Speech Recognition." *Interspeech*. 2018.
- [20] Siniosoglou, Ilias, et al. "A unified deep learning anomaly detection and classification approach for smart grid environments." *IEEE Transactions on Network and Service Management* 18.2 (2021): 1137-1151.
- [21] Larriva-Novo, Xavier, et al. "An IoT-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets." *Sensors* 21.2 (2021): 656.
- [22] Cohen, Patricia, Stephen G. West, and Leona S. Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Psychology press, 2014.
- [23] Conrod, Patricia J., et al. "Effectiveness of a selective, personality-targeted prevention program for adolescent alcohol use and misuse: a cluster randomized controlled trial." *JAMA psychiatry* 70.3 (2013): 334-342.
- [24] Cerda, Patricio, Gaël Varoquaux, and Balázs Kégl. "Similarity encoding for learning with dirty categorical variables." *Machine Learning* 107.8 (2018): 1477-1494.
- [25] Guo, Cheng, and Felix Berkhahn. "Entity embeddings of categorical variables." *arXiv preprint arXiv:1604.06737* (2016).
- [26] Zhou, Yuyang, et al. "Building an efficient intrusion detection system based on feature selection and ensemble classifier." *Computer networks* 174 (2020): 107247.
- [27] Mahfouz, Ahmed, et al. "Ensemble classifiers for network intrusion detection using a novel network attack dataset." *Future Internet* 12.11 (2020): 180.
- [28] Ma, Xiaolei, et al. "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data." *Transportation Research Part C: Emerging Technologies* 54 (2015): 187-197.
- [29] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014).
- [30] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014).
- [31] Do, Phuc Hao, et al. "An Efficient Feature Extraction Method for Attack Classification in IoT Networks." 2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). IEEE, 2021.
- [32] Volkov, A., Khakimov, A., Muthanna, A., Kirichek, R., Vladkyo, A., & Koucheryavy, A. (2017, June). Interaction of the IoT traffic generated by a smart city segment with SDN core network. In *International Conference on Wired/Wireless Internet Communication* (pp. 115-126). Springer, Cham.



Phuc Hao Do received his MS degree in Computer science from the University of Danang - University of Science and Technology in 2017. He is currently a Ph.D. student in the Department of Communication Networks and Data Transmission at the Bonch-Bruевич Saint-Petersburg State University of Telecommunications, Russia. His research interests include AI, ML, D and its application in different fields like network, blockchain.

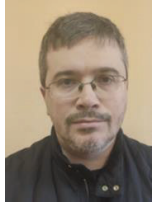


Dr. Duc Tran Le acquired his degree of Ph.D. at Admiral Makarov State University of Maritime and Inland Shipping, Russia in 2018. He works in Information Technology Faculty, The University of Danang - University of Science and Technology, Danang, Vietnam from 2019. His research areas include Internet of Things, wireless network, network security, QoS, WLAN, Software-defined networking.



Dr. Sc. Vladimir M. Vishnevsky received the Engineering degree in applied mathematics from the Moscow Institute of Electronics and Mathematics, Russia, in 1971, the Ph.D. degree in queueing theory and telecommunication networks and the D.Sc. degree in telecommunication networks from the V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences (ICS RAS), in 1974 and 1988, respectively. He became a Full Professor with ICS RAS in 1989 and the Moscow Institute of Physics and Technology in 1990. He was an Assistant Head of the Institute of Information Transmission Problems of RAS from 1990 to 2010 and an Assistant Head of laboratory with ICS RAS from 1971 to 1990. He is currently the Head of Telecommunication Networks Laboratory, ICS RAS. He is a member of

Expert Councils of Russian High Certifying Commission and Russian Foundation for Basic Research, member of IEEE Communication Society, International Telecommunications Academy and New York Academy of Science. He has authored over 300 papers in queuing theory and telecommunications. He is a Co-Chair of IEEE conferences - ICUMT, RTUWO, and the General Chair of DCCN conference. His research interests lie in the areas of computer systems and networks, queuing systems, telecommunications, discrete mathematics (extremal graph theory, mathematical programming) and wireless information transmission networks.



Dr Aleksandr Berezkin, is working at the Bonch Bruevich Saint Petersburg State University of Telecommunications as the Associate Professor of Department of Programming Engineering and Computer Science. Science interest are Computer Vision and Machine Learning. In 2009, he defended his thesis with the topic "Model and method of decoding error correction based on neural network". Now he is doctoral student at the Department of Programming Engineering and Computer Science.



Dr. Sc. Ruslan Kirichek is working at the Bonch Bruevich Saint Petersburg State University of Telecommunications as the head of Department of Programming Engineering and Computer Science. He was born in 1982 in Tartu (Estonia). He graduated Military-Space Academy A.F. Mozhaiskogo and the Bonch-Bruevich St. Petersburg State University of Telecommunications in 2004 and 2007, respectively. He received Ph.D. at the Bonch-Bruevich St. Petersburg

State University of Telecommunications in 2012 and Dr.Sc. at the Povolzhskiy State University of Telecommunications and Informatics in 2018. From 2008 to 2013 he worked as a senior researcher at the Federal State Unitary Enterprise "Center-Inform". Since 2012 he has been working as the Head of the Internet of Things Laboratory at the Bonch-Bruevich Saint Petersburg State University of Telecommunications. Since 2017 he has been working as ITU-T Q12/11 Rapporteur in "Testing of Internet of things, its applications and identification systems". He is a General Chair of the International Conference "Internet of Things and Its Enablers" (inthiten.org).