



Federated Learning Ensemble Voting-Based Methods to Deal with Non-IID and Imbalanced Data in Cybersecurity

Bruno H. Meyer^a, Michele Nogueira^b, Wagner M. Nunan Zola^a, and Aurora Pozo^a

^aDepartment of Informatics (UFPR), Curitiba, Brazil

^bDepartment of Computer Science (UFMG), Belo Horizonte, Brazil

✉ bruno.meyer@ufpr.br, michele@dcc.ufmg.br, wagner@inf.ufpr.br, aurora@inf.ufpr.br

KEYWORDS ABSTRACT

Federated Learning; ensemble models; cybersecurity

Cybersecurity is an essential topic in society due to the increasing application of technology across different application industries that are connected to the Internet. There are several methods to address cybersecurity challenges, such as Intrusion Detection Systems (IDS). This article presents new methods based on Federated Learning that can be applied to IDS to improve cybersecurity attack detection. Researchers use different approaches to create IDS solutions, including machine learning algorithms and techniques such as Federated Learning (FL) to classify network traffic as normal or malicious. FL is an emerging technology and is expected to benefit cybersecurity by improving attack detection and threat identification. It consists of a set of clients, each one with local data, whose goal is to collaboratively train one or more global models without centralizing the data, through several iterations known as rounds. An important characteristic to consider in cybersecurity data for machine learning algorithms is the presence of non-Independent and Identically Distributed (non-IID) and imbalanced data. Non-IID data describes FL datasets in which data is not evenly distributed between clients. Unlike previously proposed methods in other works, this article proposes approaches that focus on non-IID data while using FL to create models without requiring multiple training rounds. The FLENV and FLEWNV methods are proposed, implementing a federated learning framework that uses a single training round and aggregates clients through ensemble learning with normalized and weighted voting, focusing on non-IID and imbalanced data. The well-known Ton-IoT and Bot-IoT datasets were analyzed to evaluate the proposed methods and other possibilities commonly used in the state of the art. The approaches outperformed other methods from the literature in the experiments, indicating that leveraging knowledge of non-IID and imbalanced data in cybersecurity can lead to improved attack classification frameworks.

Bruno H. Meyer, Michele Nogueira, Wagner M. Nunan Zola and Aurora Pozo

Federated Learning Ensemble Voting-Based Methods to Deal with Non-IID and Imbalanced Data in Cybersecurity



1. Introduction

Communication technologies have significantly evolved in the past few decades, leading billions of people to perform daily activities using the Internet. The complexity of these activities, when analyzed through Internet traffic data, presents several challenges, particularly in cybersecurity. Cybersecurity issues emerge from various vulnerabilities and have attracted considerable attention due to potential financial losses and reputational damage for corporations.

The increasing number of devices connected to the Internet, especially through the Internet of Things (IoT), expands the attack surface, enabling attacks that generate high volumes of network traffic. In Distributed Denial of Service (DDoS) attacks, numerous devices are controlled to generate traffic against a target service, potentially denying service to legitimate users. Due to the sophisticated techniques used in such attacks, distinguishing malicious from legitimate traffic data is challenging.

Artificial intelligence (AI) assists in solving complex cybersecurity problems. Traditionally, defense techniques involve a centralized server processing substantial amounts of data collected from edge devices deployed on local networks. However, concerns over data privacy restrict the exchange of sensitive data between edge devices and servers, preventing data centralization due to legal or security reasons. Federated Learning (FL) is an AI technique that allows for the construction of models without centralizing data, enabling data privacy, and distributed processing among different devices. In FL, raw data never leaves edge devices; instead, edge devices train machine learning models locally, and only model parameters are exchanged between nodes. The goal is to create a global model by aggregating local models without transferring training data.

Federated Learning (FL) formalizes this scenario as a supervised learning problem where training data is distributed across multiple clients, and a single global model must be created without transferring data outside each client. Each client trains a local model using its data, and the global model is obtained by aggregating these local models.

Several methods have been proposed to detect cybersecurity attacks. Among these, Federated Learning has gained attention due to its ability to meet data privacy requirements while maintaining high detection performance (Alazab et al., 2021). AI algorithms improve methods that consider network features, as AI techniques can solve complex problems (Abdullahi et al., 2022). Attacks generate information in network traffic that can be used to identify abnormal activities. Therefore, FL methods for dealing with these attacks involve computational problems such as anomaly detection or classification, using data from different layers of the network protocol stack.

Federated Averaging (FedAvg) was the first model introduced in FL scenarios (McMahan et al., 2017), averaging parameters of local models (usually neural networks) to compose the global model. While FedAvg has been used to create cybersecurity solutions, it presents limitations, such as the computational resources required to train models and reduced effectiveness in predicting new instances (Ghimire and Rawat, 2022). Moreover, most studies in FL for intrusion detection systems (IDS) focus on proposing novel neural network architectures, which may not sufficiently address the characteristics of cybersecurity data. Classical machine learning algorithms, such as Random Forest, often outperform neural networks in supervised learning problems (Sewak, et al., 2018). Therefore, there is a need for studies that investigate FL solutions not based on neural networks for cybersecurity problems.

To address these limitations, we propose a new federated learning method that uses only one round of communication, employing ensemble methods with novel voting strategies to aggregate classifiers. Specifically, we introduce two methods: Federated Learning Ensembled Normalized Voting (FLENV) and Federated Learning Ensembled Weighted Normalized Voting (FLEWNV). These methods aim to

mitigate biases introduced by clients' contributions in federated scenarios with non-IID (non-independent and identically distributed) and imbalanced data, achieving higher classification performance in detecting cybersecurity attacks.

While ensemble voting strategies exist, none of them normalize or weight voter contributions according to client-level class coverage and class imbalance, nor are they tailored for non-IID cybersecurity FL scenarios. To our knowledge, this is the first work introducing normalized and weighted ensemble voting for federated intrusion detection.

In our research, we also address the lack of appropriate datasets for evaluating FL methods in cybersecurity. We provide methods for deriving cybersecurity datasets for FL from publicly available data, ensuring they reflect realistic scenarios with properties such as data heterogeneity and imbalanced distributions. We processed the Ton-IoT and Bot-IoT datasets using these methods to simulate federated clients and compare different FL techniques for detecting cybersecurity attacks.

We compared FLENV and FLEWNV with the standard FL algorithm using FedAvg and a one-round FL using a majority voting strategy. The comparison considers an IDS scenario where network traffic is classified using data from the processed datasets. Various numbers of clients were used to assess scalability. The Random Forest classifier was employed in the one-round FL implementations. The F1-score was used to measure the efficiency of each method in identifying cybersecurity threats, and additional analyses explored the correlation between F1-score and the variation of available data in each client.

Our experimental results demonstrate that FLENV and FLEWNV outperform the classical FedAvg algorithm and the original one-shot federated learning approach. They were able to identify at least one cyberattack connection for each type of attack in scenarios generated by the IP partitioning method. In total, 482 federated learning problems were created using techniques to process the well-known Bot-IoT and Ton-IoT datasets. As an additional contribution, the code developed to generate these derived datasets will be made publicly available.

This article proceeds as follows. Section 2 reviews state-of-the-art research in cybersecurity federated learning. Section 3 introduces federated learning. The proposed methods of FLENV and FLEWNV are detailed in Section 4. Section 5 describes the experimental setup and datasets used. The results are discussed in Section 6. Finally, Section 7 presents the concluding remarks.

2. Related Works

This section presents and discusses works focusing on Federated Learning methods applied to cybersecurity scenarios. Cybersecurity is one of the application domains for the emerging Federated Learning technology, in which several problems and solutions have been explored in previous works (Alazab et al., 2021; Ferrag et al., 2021; Ghimire and Rawat, 2022; Campos et al., 2022). It is possible to apply FL techniques to solve cybersecurity problems or to leverage cybersecurity solutions to address security limitations in FL. Ghimire and Rawat (2022) focused on the use of FL techniques to solve four main types of cybersecurity detection problems: spoofing, intrusion, anomaly, DoS/DDoS. Some solutions create subcategories from anomalous behavior to represent other types of cybersecurity attacks. Different types of data can be used to detect cybersecurity attacks. Commonly, network traffic is used to create detection models (Ghimire and Rawat, 2022). Moreover, most previous works that apply Federated Learning for cybersecurity rely on the FedAvg algorithm, which trains several client models simultaneously over several rounds. This application of FedAvg can follow different

strategies, such as using time-series data (Zhao et al., 2020), detection at the network node level, and detection at network flow level (Alazab et al., 2021).

The FedAvg algorithm implements frameworks that analyze captured network flow data where cybersecurity attacks should be detected. Deep learning algorithms commonly employed in these works are autoencoders (Schneible and Lu, 2017), Long Short-Term Memory (LSTM) (Zhao et al., 2020), Gated Recurrent Units (GRU) (Li et al., 2021a). However, some federated learning strategies do not rely on deep learning models. For example, Federated Forest (Liu et al., 2020; Aliyu et al., 2021) is a novel algorithm that implements the well-known Random Forest algorithm using distributed clients in the federated scenario. Ensemble learning is also an algorithm employed in several cybersecurity models and was already used for federated learning (Attota et al., 2021). Methods based on ensemble learning take advantage of aggregating several «weak» models to create a «stronger» model. In other words, weak and stronger models indicate the capacity of the final model to perform accurate predictions and avoid overfitting, also performing accurate predictions for data not used to train the models. When used in FL settings, ensemble learning can also reduce training time using only one training round among FL clients without losing its prediction capacity (Guha et al., 2019; Zhou et al., 2020). Currently, there are multiple approaches to combining different models using ensembles, including different types of voting strategies. However, a series of challenges can arise when applying ensemble approaches to Federated Learning (Dai et al., 2024). In Dai et al. (2024), an approach based on the Co-Boosting ensemble and the synthesizing of data was presented in an adversarial manner. Although the authors obtained a significant improvement compared to the baseline, their research did not explore or mention any voting strategy to ensemble the prediction of multiple classification models.

Intrusion Detection Systems (IDS) are commonly analyzed in research focusing on cybersecurity and Federated Learning (Zhao et al., 2020; Khoa et al., 2020; Farid et al., 2021; Ghimire and Rawat, 2022; Li et al., 2021a; Ferrag et al., 2022). Datasets constructed for supervised learning technique evaluation in IDS were commonly used in past literature. These datasets were proposed for a centralized learning task, and the most common procedure adopted for federated learning consists of randomly partitioning these datasets in artificial client subsets. However, realistic federated learning in IoT scenarios usually produces different patterns and amounts of data in each client (Ghimire and Rawat, 2022), also referred to as non-independent and identically distributed (non-IID).

Some works used the Dirichlet distribution (Lin, 2016) to effectively produce non-IID data partitioning for federated learning using centralized supervised learning data (Lin et al., 2020). Novel federated learning frameworks were proposed focusing on problems that use non-IID and imbalanced data (Zhu et al., 2021). These frameworks mitigate common issues of classic FL methods such as FedAvg for these data types. For instance, local models trained with little data can introduce bias to the global model. However, the frameworks proposed to surpass these issues do not concentrate on cybersecurity scenarios. There is already evidence of the negative effect that non-IID data can have on FedAvg-based algorithms (Li et al., 2019; Hsu, Qi, and Brown, 2019; Wang et al., 2024). In Nguyen et al. (2022), an LSTM-based method is proposed that incorporates over-fitting prevention techniques based on impact factor definitions for each federated client, thereby mitigating problems caused by non-IID. Recent works have defined the One-Shot Federated Learning framework, where only one round of communication is required to create a global model. In Guha et al. (2019), an ensemble distillation strategy is applied. In a single round, the clients generate a local model that is sent to the server, where a global model tries to copy the output of these models. The idea to create distilled data using the client's private data was employed by Zhou et al. (2020), which can also be performed in one round. A disadvantage in using distillation strategies to implement One-Shot Federated Learning is the requirement to solve a

new optimization problem. In this setting, a new training step is required to create a model that mimics another. Also, strategies in these frameworks will be restricted to using prediction models that can be trained to simulate the prediction of other models. This is not the general case of Ensemble Learning based techniques that can view the prediction model as black boxes.

Although the models mentioned before could successfully solve the problems in previous research experiments, there need to be studies that focus on effective techniques for mitigating non-IID problems in cybersecurity federated learning. No research using One-Shot Federated Learning for cybersecurity issues could be found in the literature review done in this article. Also, using the distillation method to aggregate client models in federated learning is one of the many options. Some works have already used plurality voting to solve this task (Yue et al., 2022). In Zhao et al. (2024), the authors proposed an approach that focuses on using data heterogeneity (non-IID) to apply Knowledge Distillation in FL scenarios. Although the authors demonstrated that their proposal was able to improve model accuracy and overcome different challenges, the limitations observed in their experiments indicate opportunities to explore alternative approaches for the transfer of knowledge between client and server models in FL scenarios. It is possible to use different voting strategies (Leon et al., 2017), and no study has proposed strategies for ensemble voting specifically for cybersecurity scenarios with non-IID and imbalanced data.

Prior to one-round FL methods rely mainly on knowledge distillation, which introduces an additional optimization step. Our approach differs by aggregating black-box models directly through normalized and weighted voting, removing the need for extra training.

3. Federated Learning

Federated Learning was initially proposed to overcome the limitations of machine learning implemented with distributed processing. These limitations come with the requirement to centralize the data used to train models. When using centralized processing to create supervised learning models with distributed processing, the transfer of training data can require significant execution time. Also, these data can be sensitive, requiring privacy and making it impracticable to apply distributed processing by centralizing the training data.

Instead of sharing data among nodes, it is possible to share parameters of trained models. This characteristic is the main idea of federated learning. Each node that contains training data, also referred to as clients, can train a local model. This model must be parametric so that communication can be established in which clients send only the model parameters to a server. Then, this server can aggregate all parameters into a single model that will take advantage of the training given by all clients. This process is represented in Figure 1, which shows several devices training local models and sending them to a server that creates a single model. In the federated learning context, the device types can be heterogenous, as can the data used to train the local models. Heterogenous data means the scarcity or absence of one or more labels considering a classification problem. Other strategies go beyond centralized models, such as those that use hierarchical and decentralized topologies for data sharing, which reduces latency and communication time. However, the most common models are those based on centralized strategies, which are appropriate for several realistic scenarios.

Federated Learning training consists of a process where several rounds are executed. In each round, a subset of clients is selected, and these clients train their learning models locally using their own data. The clients then send the updated parameters to the server, without transmitting the training data.

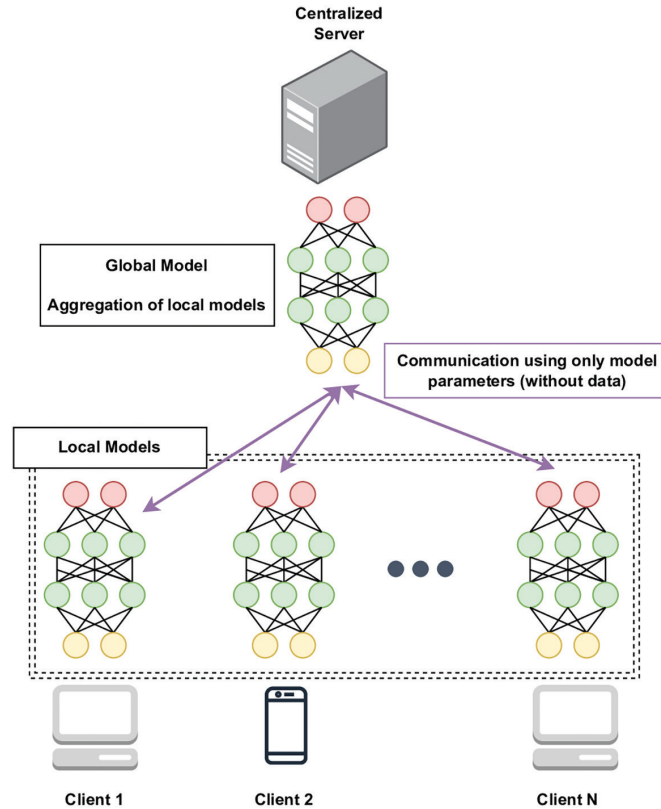


Figure 1. Classical Federated Learning framework

At the end of the round, the server aggregates the received parameters to create a single model, which is sent back to all clients and used as the local model in subsequent rounds. The global model can be created in several ways, with Federated Averaging (FedAVG) being the main algorithm, which consists of averaging the parameters. This type of strategy only works for some types of machine learning techniques, with neural networks being the most used in this context where the weights learned in the networks are the parameters sent to the server. Several rounds are executed until the global model reaches convergence, or some user-defined criteria of the federated model are reached.

3.1. One-Round Federated Learning

One-Shot Federated Learning (Vinyals et al., 2016) was proposed as a strategy for training Federated Learning models using only one round of communication and training. It is important to note that this strategy is different from classical One-Shot Learning (Vinyals et al., 2016), where metric learning is applied to create models capable of comparing the similarity of input instances using only a few examples. The «One-Shot» of One-Shot Federated Learning refers to the simplification of the federated training process of models to only a single communication round. For simplicity, in this paper we refer to One-Shot Federated Learning as One-Round Federated Learning (ORFL).

In its original formulation, ORFL relied on an ensemble learning approach to aggregate the models of each client using the semi-supervised distillation setting to provide privacy guarantees (Papernot et al., 2016). Ensemble learning methods in machine learning consist of merging several models into one so that its ensemble can give better results when compared to each model (Jurek et al., 2014). The distillation strategy in ORFL involves training a teacher model at each client to predict probabilities of inputs, also known as «soft» labels. The centralized server then learns with these soft predictions to create the same input as all teacher models. In this paper, the distillation strategy is not addressed, although our proposal could be extended with this idea.

3.2. Data Distribution in Federated Learning and Cybersecurity

An important detail to consider in FL is the diversity of data and hardware resources in each client. Several factors can affect diversity, such as the nature of the data or the number of customers. In supervised learning databases, each instance is labeled with one of two or more classes. Imbalance is a concept addressed in centralized learning techniques that prevent problems such as biased models generated by supervised learning algorithms. An imbalance occurs when a supervised learning database has a considerable proportion of classes in its data. For instance, this can happen in the cybersecurity context when data contains normal traffic that is not related to attacks, which is the most common type of traffic.

In a federated learning scenario, if the data pattern is the same among clients, it is considered independent and identically distributed (IID). Conversely, when there are differences among the clients' data distributions, the data is referred to as non-IID (Zhu et al., 2021). One type of non-IID consists of the label distribution skew, which happens when each client contains a different proportion of classes following the Dirichlet distribution (Yurochkin et al., 2019). In the Dirichlet distribution, a parameter $\beta \geq 0$ is used to control the concentration of classes in each client. As beta is closer to 0, the more concentrated the classes are in one or few clients.

When Federated Learning is applied to cybersecurity scenarios, it is expected that only a part of the federation observes and captures network traffic related to attacks. Therefore, when flows (connections) are labeled as normal traffic or an attack, the majority of clients will contain normal flows, and the attack instances will be concentrated on some clients' data. This is, as mentioned before, the definition of non-IID data, which is a problem that will be addressed in this paper.

3.3. Dirichlet Distribution

Most popular datasets used as benchmarks in cybersecurity research were not proposed focusing on the Federated Learning scenarios. However, it is possible to artificially generate a benchmark using these popular datasets for evaluating Federated Learning techniques. This can be done by partitioning the data into subsets, where each data subset will represent the data produced by one device in a federated scenario.

However, artificially generating data through processing real datasets can introduce some challenges and limitations. For instance, the generated data could not faithfully represent a real-world scenario, mainly due to the limited number of devices used in the original dataset. Unfortunately, this issue is already a problem for non-federated cybersecurity researchers. It is a limitation that can be surpassed by works focusing on producing datasets with several devices to simulate or emulate a real cybersecurity challenge. Another challenge when using standard datasets to evaluate Federated Learning techniques is the problem generated by randomly partitioning the data. Assuming no criteria for the

partitioning procedure, the clients will tend to contain similar data. However, this characteristic is not observed in cybersecurity data, where datasets are composed of several imbalanced and non-IID data.

The Dirichlet distribution is a statistical tool that can be used to create partitions of supervised-learning datasets assuming non-IID and imbalanced data. This strategy was adopted by previous research that evaluated Federated Learning models (Hsu et al., 2019) (Yurochkin et al., 2019) (Lin et al., 2020). The Dirichlet distribution enables the generation of a random variable that will follow a non-homogeneous pattern. In other words, the same pattern of unbalanced and non-IID federated learning data can be represented in this distribution, where only a few clients contain the majority of data or most data of some classes.

4. Ensemble Voting for Imbalanced Data in Federated Learning

An ensemble is a machine learning method that combines supervised learning algorithms to create a unique model. This combination can be made in several ways. One possibility to create an ensemble of classification algorithms consists of counting each classifier's output class and selecting the class with the majority of votes. This strategy is known as the hard voting method. Other possibilities create constraints for the selected classifiers in the ensemble, such as the soft voting method, where the classifier output consists of predicted probabilities for each class. It has been shown in several works that the combination of weak classifiers through ensembles can create a better model compared to each weak classifier individually (Leon et al., 2017).

The proposed framework of One Round Federated Learning described by (Vinyals et al., 2016) can be implemented using Ensemble methods instead of the semi-supervised distillation method (Lin et al., 2020). The implementation considers each federated client model as a voter in an Ensemble of aggregation. This ensemble strategy allows two ORFL characteristics: The training step can be performed only once on clients, and the clients can generate new models when new data are available. Since the machine learning model parameters generated by the clients do not reveal information about the raw training data, Federated Learning data privacy is still guaranteed. This section will describe two novel voting strategies for implementing an ensemble in ORFL that can overcome challenges observed in cybersecurity datasets.

4.1. Ensemble Voting for Cybersecurity Federated Learning

Figure 2 illustrates the majority (or hard) voting strategy in ensemble algorithms. Hard voting assumes an equal contribution of votes in the combined ensemble classifiers. Although it is a simple policy for aggregating models, it is a popular and efficient method in several machine-learning applications. However, cybersecurity federated learning scenarios have peculiar characteristics such as imbalanced and non-IID data.

When federated learning is applied to detect and classify network connections as normal or attack traffic, the proportion of classes observed in each client should be considered. This article introduces Federated Learning Ensembled Normalized Voting (FLENV). FLENV is detailed in Figure 3, where the method is mainly differentiated from hard voting. The focus is on considering the possibility that some clients may not have data related to some types of attacks. If the previous characteristic happens with too much frequency, the models composing the Ensemble can introduce a bias to ignore classes that appear only in a few clients. In Equation 1 TO_c represents the total of clients that contains observations for any class (or label) c , assuming C_i as the set of classes observed in the i -th client.

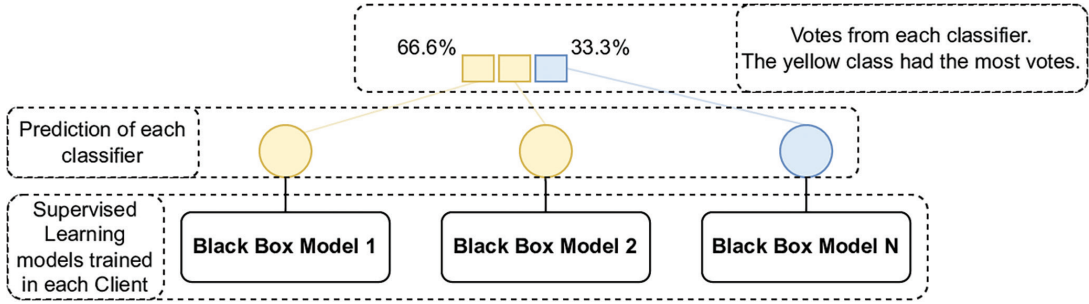


Figure 2. Majority (hard) voting strategy for Federated Learning. The figure shows an example where three supervised learning models generated predictions for an instance. In the example, two predictions are related to one class (yellow) and one prediction to another (blue). In this scenario, the class with most votes (66.6 %) is considered as the final prediction

Intuitively, TO_c counts how many clients contain at least one example of class c . This prevents models from ignoring classes that are absent in most clients. Similarly, it is possible to define TV_c as the total of clients that vote for a class C as shown in Equation 2, where V_i represents the label prediction of the i -th client. Finally, a normalized probability can be calculated for each class as defined in Equation 3 and using the total of observations of the class in the model's predictions and the total of clients able to predict this class. Figure 3 provides an example scenario where non-IID is considered, and FLENV could present better classification results compared to the majority voting strategy, so that over-observed classes do not create a bias in the ensemble.

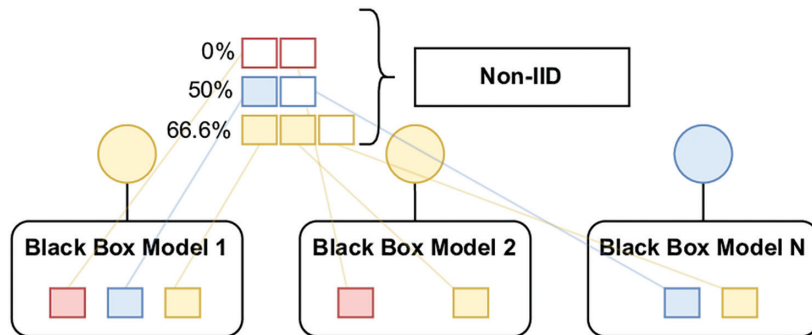


Figure 3. FLENV strategy for Federated Learning

Although the FLENV voting policy compensates for non-IID data problems in ORFL methods that use ensembles, it does not consider the influence of imbalanced data. Figure 4 presents a scenario where data is non-IID and imbalanced among clients of a Federated Learning scenario. When class observations are poorly distributed among clients, some clients' models are expected to become specialized in some classes. Then, if these specialist models predict a label in which they are specialized, an Ensemble model could take advantage of the property mentioned before and give a higher weight to this vote.

Federated Learning Ensembled Weighted Normalized Voting (FLEWNV) is proposed to take advantage of this weight characteristic. The idea is also supported by observing weighted contribution methods in other federated learning strategies for model aggregation, such as in the FedAvg algorithm. Equation 4 defines $TCW_{c,i}$ to represent the total of training instances of a specific class c present in the i -th client, where X_i is the set of instances of the training set and Y_i the corresponding set of labels for these instances. Equation 5 defines $TCWN_c$, the total count of observations of each class c among clients.

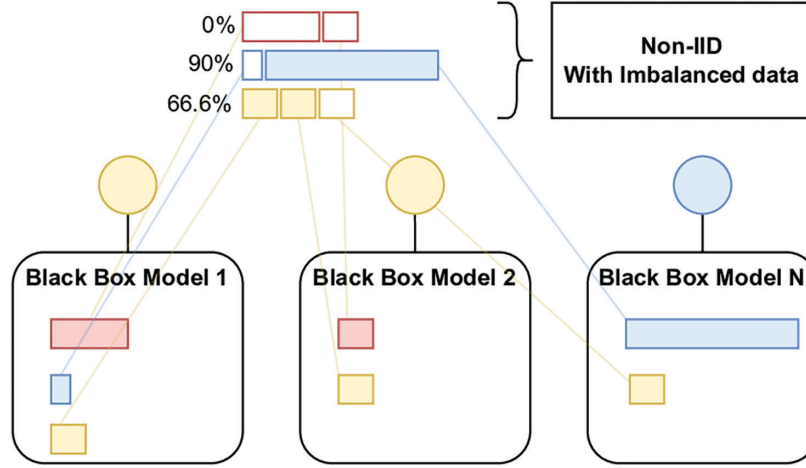


Figure 4. FLEWNV strategy for Federated Learning

Similarly to the FLENV vote method, FLEWNV counts each client model's votes as described in Equation 6. However, the number of training observations of the predicted label is used to weigh the vote. Finally, using these weighted votes, the probability of each class can be calculated using Equation 7. This weighting amplifies the influence of client models that have observed many samples of a given class, which is crucial when attack classes are rare. FLEWNV policy in Ensemble classification can significantly boost the possibility of predicting unseen and infrequent classes among several clients, as shown in Figure 4. However, extremely imbalanced and non-IID data can lead to the FLEWNV strategy to create a bias for classes when classification algorithms used in clients fail to create generalized models that avoid overfitting problems. Therefore, in some scenarios, FLENV does not consider weights and can possibly surpass the classification performance of FLEWNV. In cybersecurity federated learning, these aspects should be considered together with the data used to train the supervised learning models, such as the total number of attack classes, the usage of pre-processing models to filter unnecessary data, and the number of clients. Compared to classic majority voting, normalization reduces bias introduced by clients lacking certain classes. Weighted normalization further accounts for class imbalance, aligning the voting strength with the statistical evidence held by each client.

$$TO_c = | \{ i: c \in C_i \} | \quad \text{Equation (1)}$$

$$TV_c = | \{ i: V_i = c \} | \quad \text{Equation (2)}$$

$$\text{FLEV } P_c = \frac{TV_c}{TO_c} \quad \text{Equation (3)}$$

$$\text{TCW}_{c,i} = \sum | \{ X_{i,j} : X_{i,j} \in X_i, Y_{i,j} = c \} | \quad \text{Equation (4)}$$

$$\text{TCWN}_c = \sum \text{TCW}_{c,i} \quad \text{Equation (5)}$$

$$\text{TWV}_c = \sum | \{ X_{i,j} : X_{i,j} \in X_i, Y_{i,j} = c \} | \quad \text{Equation (6)}$$

$$\text{FLEWV } P_c = \frac{\text{TWV}_c}{\text{TCWN}_c} \quad \text{Equation (7)}$$

5. Methods

This section describes the experiments performed to validate the concepts and methods presented in this article. The experiments were performed using a computer with an i7-5930K 3.50GHz processor (12 threads) and 40GB of RAM. All code related to the implementation and evaluation of methods was developed using the Python language with TensorFlow, Scikit-learn, and SciPy libraries.

5.1. Datasets and data processing

All experiments have used the Ton-IoT (Moustafa, 2021), and Bot-IoT (Koroniotis et al., 2019) datasets. Ton-IoT data consists of 6 types of labels, with instances representing flow connections with 40 feature values. Bot-IoT data is similar to Ton-IoT but considers only 14 features and five types of labels. Features related to timestamps, IP addresses, and source and destination ports were not used in the experiments of this article, which is a recommended and common practice in others research. The number of training and test instances for each class is detailed in Tables 1 and 2. Both datasets are publicly available, anonymized, and widely used for FL/IDS research, ensuring ethical and reproducible experimentation.

Table 1. Ton-IoT dataset classes distribution

	Normal	Backdoor	Injection	Password	Scanning	XSS
Training set	166799	11120	11120	11120	11120	11120
Test set	133201	8880	8880	8880	8880	8880

Table 2. Bot-IoT dataset classes distribution

	Normal	DDoS	DoS	Reconnaissance	Theft
Training set	370	1541315	1320148	72919	65
Test set	107	385309	330112	18163	14

The Bot-IoT and Ton-IoT datasets were used to build several Federated Learning scenarios, using partitioning methods to simulate several client sets using the Dirichlet distribution. Four different scenarios were defined for 2, 4, 8, 16, 32, and 64 clients when the partitions were generated with the Dirichlet distribution. Each of these four sub-scenarios assumes different values of β in the Dirichlet distribution, which controls the non-IID characteristic. Also, due to the randomness related to the partitions generated with the chosen distribution, ten random seeds were used to increase the statistical validation of the experimental results. The same methods used by Q. Li et al. (2021b) were implemented to create the partitioning following the Dirichlet distribution.

Another scenario was defined using a deterministic partitioning method for each original dataset, which considers the selection of network connections that share a local address (IPV4) and assumes public address as one unique set of data representing the Internet, generating 16 clients for Bot-IoT and 30 clients for Ton-IoT. This method is closest to a realistic federated learning scenario because monitoring tools used to capture network traffic will be deployed in specific locations of networks or the Internet.

Therefore, 482 federated learning problems were used to execute experiments that help answer the research questions defined later in this Section. For each problem, a test set was used to evaluate the classification results generated by each classification method. The code developed to process data and generate problems will be publicly available in the future. Due to the computational cost of generating 482 federated partitions, a third dataset could not be incorporated in this research. However, Bot-IoT and Ton-IoT already represent two extremes of non-IID severity and class imbalance, providing sufficient experimental coverage.

5.2. Federated Learning Methods

This article focuses on the efficiency of standard and straightforward methods used in cybersecurity federated learning and compares these methods with our contributions (FLENV and FLEWNV). The classical FedAvg algorithm was used as a baseline to compare directly with ORFL strategies. The Deep Learning architecture and configuration follow the ANN description used in (Ferrag et al., 2021), where the model is trained using 50 rounds. We have implemented the framework described in (Zhou et al., 2020) using an Ensemble of Random Forest classifiers, where a different federated client trained each Random Forest. The trees in the Random Forest models were limited to a maximum depth of 5, which prevented model over-fitting. Also, this depth limitation is essential to preserve data privacy because the raw training data cannot be retrieved from these trees. For each random seed used to perform dataset processing to create partitions for each client, the same seed was used to initialize the classification algorithms. The Ensemble was used considering three types of voting strategies: majority voting (a straightforward solution), FLENV, and FLEWNV. Therefore, four different solutions were compared and applied to solve each cybersecurity federated learning scenario.

5.3. Experiments and Evaluation

The experiments described in this article were elaborated to answer the following research questions (RQ):

- RQ1: How does classic Federated Learning classification efficiency using FedAvg differ from One Round Federated Learning when using cybersecurity data?
- RQ2: What properties should be considered in Ton-IoT and Bot-IoT datasets regarding the characteristics of imbalanced and non-IID data?

- RQ3: Can the FLENV and FLEWNV methods surpass the classification result when compared with traditional Deep Learning or majority voting in Federated Learning scenarios for cybersecurity?

5.3.1. Evaluation Metrics

- F1-score: The classification efficiency of the final model generated by the federated learning methods in the test set. The F1-score computation is described in Equation 8, which is measured to mitigate imbalanced class distribution problems observed in other metrics such as accuracy.
- Standard Deviation (STD) of clients training set sizes: The standard deviation of lists that count the whole network flows in each client used to train the local models. The more significant the STD, the bigger the non-IID degree of the scenario.
- Spearman Correlation: A measurement that describes the correlation strength (ρ) between two variables and the probability (p-value) that the correlation does not exist. In the experiments of this article, the correlation is measured to identify the relationship between the F1-score and the STD of the client's training set sizes.
- Entropy: Another measurement used to identify the non-IID of datasets and the imbalance characteristics. The Shannon entropy (Lin, 2002) value is computed for each label array in the client training set, and then these values are averaged.
- Dirichlet distribution β : Parameter used to generate dataset partitioning. The smaller β , the more non-IID will be the dataset used in the federated learning scenario.

$$F_{1\text{-score}} = \frac{2}{recall^{-1} + precision^{-1}} \quad \text{Equation (8)}$$

5.3.2. Classic Federated Learning and ORFL

Experiments were conducted to determine the evolution of the F1-score in each training round of the FedAvg algorithm and identify the difference between classic federated learning with multiple rounds and ORFL. Also, the F1-score was highlighted in our results to identify in which cases ORFL can surpass federated learning using deep learning and multiple training rounds.

5.3.3. Dataset Analysis

The multiple partitioning of Ton-IoT and Bot-IoT datasets enabled the analysis of the correlation between the F1-score and the STD of the clients' training set sizes. This correlation can be measured for each classification strategy and indicates if non-IID and imbalanced data strongly affect the classification result. The more negative the correlation is (assuming that there is a correlation), the more the strategy efficiency struggles with non-IID and imbalanced data.

5.3.4. Non-IID and Imbalanced Data Mitigation

FLENV and FLEWNV were proposed to improve ORFL ensemble-based algorithms to mitigate the challenges of non-IID and imbalanced data, a central cybersecurity problem. However, several

difficulties can harm the proposed methods' good function and create undesirable or unexpected results. This article will highlight the experiments' findings that compare the analyzed classification methods to identify FLENV and FLEWNV viability. The average F1-score of 10 executions was computed for each combination of total clients and Dirichlet β . Also, for each combination, the scenario-related dataset entropy was measured to identify correlations with the F1-score.

6. Discussion

This section discusses the results of the experiments. First, the application of the classical FedAvg algorithm will be analyzed on the datasets used in this article, and its classification efficiency will be compared with ORFL strategies. Then, an overall analysis of the datasets used in the experiments will be introduced, focusing on the characteristics that may influence the classification performance in relation to non-IID and imbalanced data. Finally, all analyzed methods will be compared across scenarios with different numbers of clients, non-IID degrees, and levels of imbalanced data.

Figure 5 shows the training evolution of the deep learning method using FedAvg with 50 rounds using the IP address partitioning scenario of Ton-IoT and Bot-IoT datasets. The F1-score of the generated method has achieved convergence before round 50 and is stochastic once the F1-score is significantly different for consecutive rounds. Also, it is possible to observe that in the Bot-IoT dataset, the convergence is more stochastic, which can be explained by the fact that the data labels are more imbalanced among clients and can harm the F1-score metric when the final model is not able to detect one or more classes.

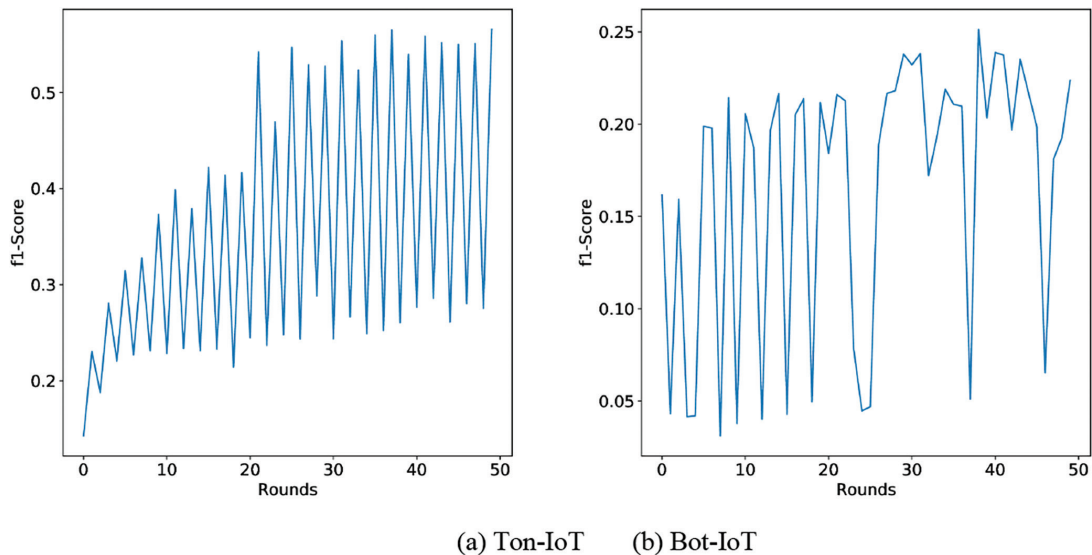


Figure 5. Evolution of global F1-score over training rounds using FedAvg, illustrating stochastic convergence due to non-IID distributions

Figure 6 presents a direct comparison of FedAvg using deep learning and the three other ORFL-based methods using the IP address partitioning scenario. In the Bot-IoT dataset, the other methods outperformed the deep learning algorithm. This characteristic is expected since no fine-tuning was performed for any method, and noncomplex deep learning architectures can struggle with large, non-IID, and imbalanced data. However, FedAvg outperformed the majority voting strategy and FLENV in the Ton-IoT dataset. This difference can be explained by the fact that the architecture design was inspired by (Ferrag et al., 2021), which used the Ton-IoT dataset in their experiments. Also, Ton-IoT contains 30 clients in this scenario, while Bot-IoT contains only 16. This difference in total clients can increase the challenges generated by the non-IID and imbalanced data. Nevertheless, FLEWNV could surpass all other techniques when considering the F1-score. This evidence corroborates the idea behind the proposal of the FLEWNV technique, that focuses on characteristics of cybersecurity federated learning data.

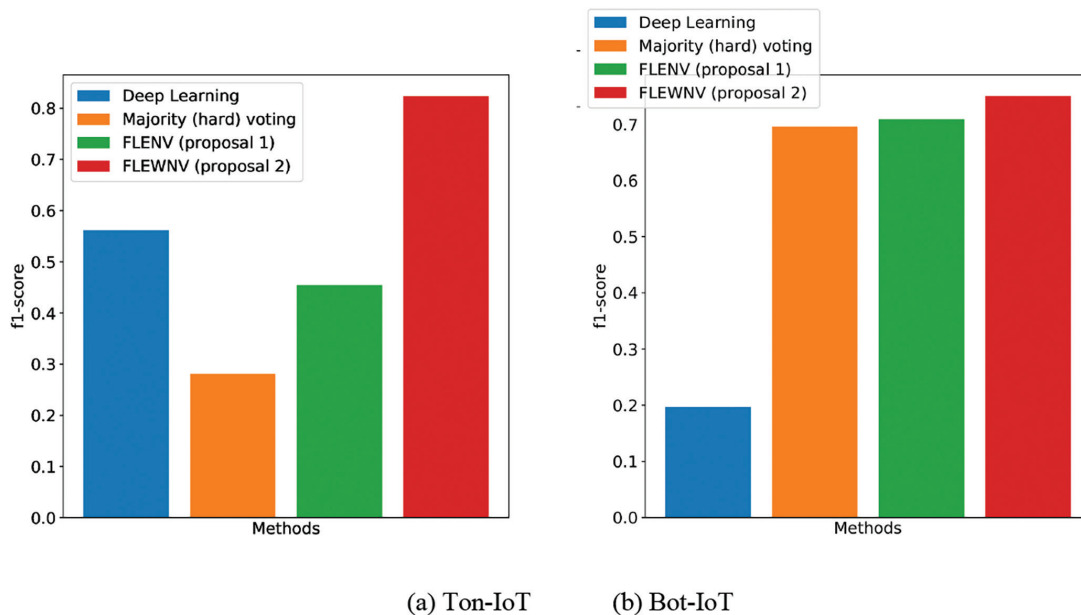


Figure 6. F1-score for all techniques using an IP address partitioning strategy to generate a federated learning problem. The Deep Learning strategy was trained using 50 rounds

Tables 3 and 4 show individual classification metrics for different classes and methods on the Ton-IoT and Bot-IoT datasets. As seen in the tables, the lowest recall achieved by FLEWNV was 0.664 in the «xss» class for the Ton-IoT dataset and 0.197 for the «Theft» class in the Bot-IoT dataset. These classes are among those with fewer training instances. In contrast, the majority voting method presented recall values equal to 0 for these classes and others. The majority voting method achieved recall equal to 0 for four different classes in the Ton-IoT dataset. When the recall is equal to 0, no true positive classification happens, meaning attacks would not be detected in a cybersecurity scenario. These results are strong evidence of the capacity of FLEWNV (and FLENV) to deal with imbalanced data in a way that the final model has a fair tendency to predict all labels.

Table 3. Individual classification metrics of ToN-IoT dataset for each class of the test set. The results are related to the IP address partitioning strategy

Class	Support	Method Name	Precision	Recall	F1-score
backdoor	8880	Majority (hard) voting	0.000	0.000	0.000
		FLENV (Proposal 1)	0.805	0.096	0.111
		FLEWNV (Proposal 2)	0.865	1.000	0.928
		Deep Learning	0.000	0.000	0.000
injection	8880	Majority (hard) voting	0.000	0.000	0.000
		FLENV (Proposal 1)	0.964	0.039	0.074
		FLEWNV (Proposal 2)	0.599	0.878	0.709
		Deep Learning	0.619	0.529	0.537
normal	133201	Majority (hard) voting	0.822	1.000	0.904
		FLENV (Proposal 1)	0.842	0.987	0.911
		FLEWNV (Proposal 2)	0.949	0.934	0.943
		Deep Learning	0.944	0.919	0.932
password	8880	Majority (hard) voting	0.000	0.000	0.000
		FLENV (Proposal 1)	0.971	0.248	0.390
		FLEWNV (Proposal 2)	0.648	0.720	0.684
		Deep Learning	0.462	0.533	0.487
scanning	8880	Majority (hard) voting	0.577	0.939	0.701
		FLENV (Proposal 1)	0.485	0.957	0.637
		FLEWNV (Proposal 2)	1.000	0.810	0.900
		Deep Learning	0.392	0.955	0.555
xss	8880	Majority (hard) voting	0.000	0.000	0.000
		FLENV (Proposal 1)	0.919	0.007	0.017
		FLEWNV (Proposal 2)	0.977	0.664	0.789
		Deep Learning	0.954	0.744	0.837

Table 4. Individual classification metrics of BoT-IoT dataset for each class of the test set. The results are related to the IP address partitioning strategy

Class	Support	Method Name	Precision	Recall	F1-score
DDoS	385309	Majority (hard) voting	0.911	0.990	0.949
		FLENV (Proposal 1)	0.894	0.991	0.942
		FLEWNV (Proposal 2)	0.834	1.000	0.909
		Deep Learning	0.455	0.000	0.000
DoS	330112	Majority (hard) voting	0.988	0.887	0.936
		FLENV (Proposal 1)	0.988	0.864	0.922
		FLEWNV (Proposal 2)	1.000	0.766	0.866
		Deep Learning	0.425	0.623	0.414

Class	Support	Method Name	Precision	Recall	F1-score
Normal	107	Majority (hard) voting	1.000	0.479	0.645
		FLENV (Proposal 1)	1.000	0.485	0.651
		FLEWNV (Proposal 2)	1.000	0.438	0.604
		Deep Learning	0.886	0.098	0.178
Reconnaissance	18163	Majority (hard) voting	0.996	0.990	0.990
		FLENV (Proposal 1)	0.991	0.988	0.988
		FLEWNV (Proposal 2)	0.992	0.975	0.983
		Deep Learning	0.274	0.633	0.211
Theft	14	Majority (hard) voting	0.000	0.000	0.000
		FLENV (Proposal 1)	0.182	0.059	0.088
		FLEWNV (Proposal 2)	0.636	0.197	0.296
		Deep Learning	0.073	0.026	0.038

The classification results of several scenarios of Ton-IoT and Bot-IoT are described in figures 7 and 8. The figures present a correlation analysis between the F1-score and a metric that measures how unfair the total data distributed among clients is. Methods that were proposed to overcome the non-IID and imbalanced data challenges (FLENV and FLEWNV) presented a more positive correlation. This correlation means that other methods that presented a correlation with high statistical probability (small p-values) and high negative correlation indices are the same methods with smaller F1-score. This observation serves as evidence that FLENV and FLEWNV could preserve classification efficiency even with unfair data distributed among clients. Specifically, Bot-IoT is a dataset with more complex problems than Ton-IoT. The correlation shown in Figure 8 highlights this difference, where the correlation is not significant for the three methods. The only method with a correlation detected was the FedAvg (deep learning) which presented the worst average F1-score. This result can be explained by the fact that FedAvg will behave similarly to centralized learning when using extremely imbalanced data, leading to a model equivalent to conventional deep learning methods in cybersecurity. However, it is essential to highlight that deep learning was the worst method when considering the required training rounds and the F1-score metric.

To isolate the effects of normalization and weighting, we examined three variants: majority voting, FLENV, and FLEWNV. Normalization alone consistently improved minority-class recall, while weighting further improved detection of rare classes by amplifying specialist clients' predictions.

As mentioned in the previous analysis, FLENV and FLEWNV were able to outperform the other methods in the experiments proposed in this article. Figures 9 and 10 present an additional analysis focused on the number of clients and the non-IID degree of data for several scenarios. It is also evident in these comparisons that FLENV and FLEWNV could easily surpass the F1-score of other methods. Specifically, deep learning presented the worst F1-score in any comparison. For higher values of β , the achieved F1-score is high, which can be explained by the fact that, in these cases, the non-IID degree is small. In contrast, when β is small, the difference between the majority and the proposed method in this article is clear. Although FLEWNV presented the best overall classification results, there is an interesting result in Figure 10d, where FLENV could surpass FLEWNV when the number of clients is higher than 32 (i.e. 32 in the experiment or 20 as can be interpolated in the figure) and $\beta = 1$. These scenarios have a higher non-IID potential, with highly imbalanced data. In these cases, some clients are expected to receive no instances of some

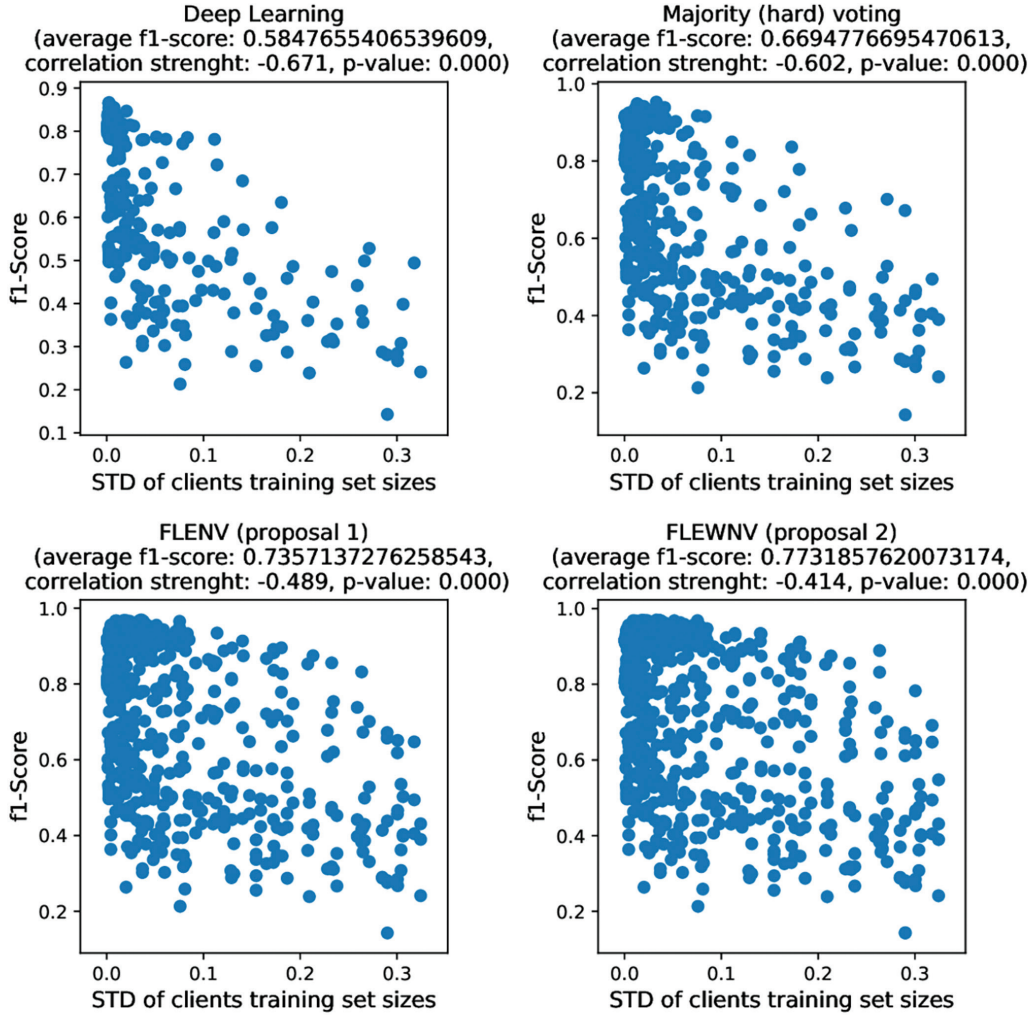


Figure 7. Correlation between client training size variation and F1-score in the ToN-IoT dataset

attack types. Also, a few clients can receive most instances of one specific class and generate a supervised model that cannot be generalized and is biased to select only one class. If this characteristic happens, it is better not to consider weights in the ensemble voting, which explains why FLENV could present a better F1-score than FLEWNV.

6.1. Research Questions Discussion

The experiments executed in this article allowed for a direct comparison between classical federated learning based on multiple rounds of training and ORFL. Using FedAvg and ensemble methods to solve cybersecurity federated learning problems, it was possible to observe that ORFL-based methods

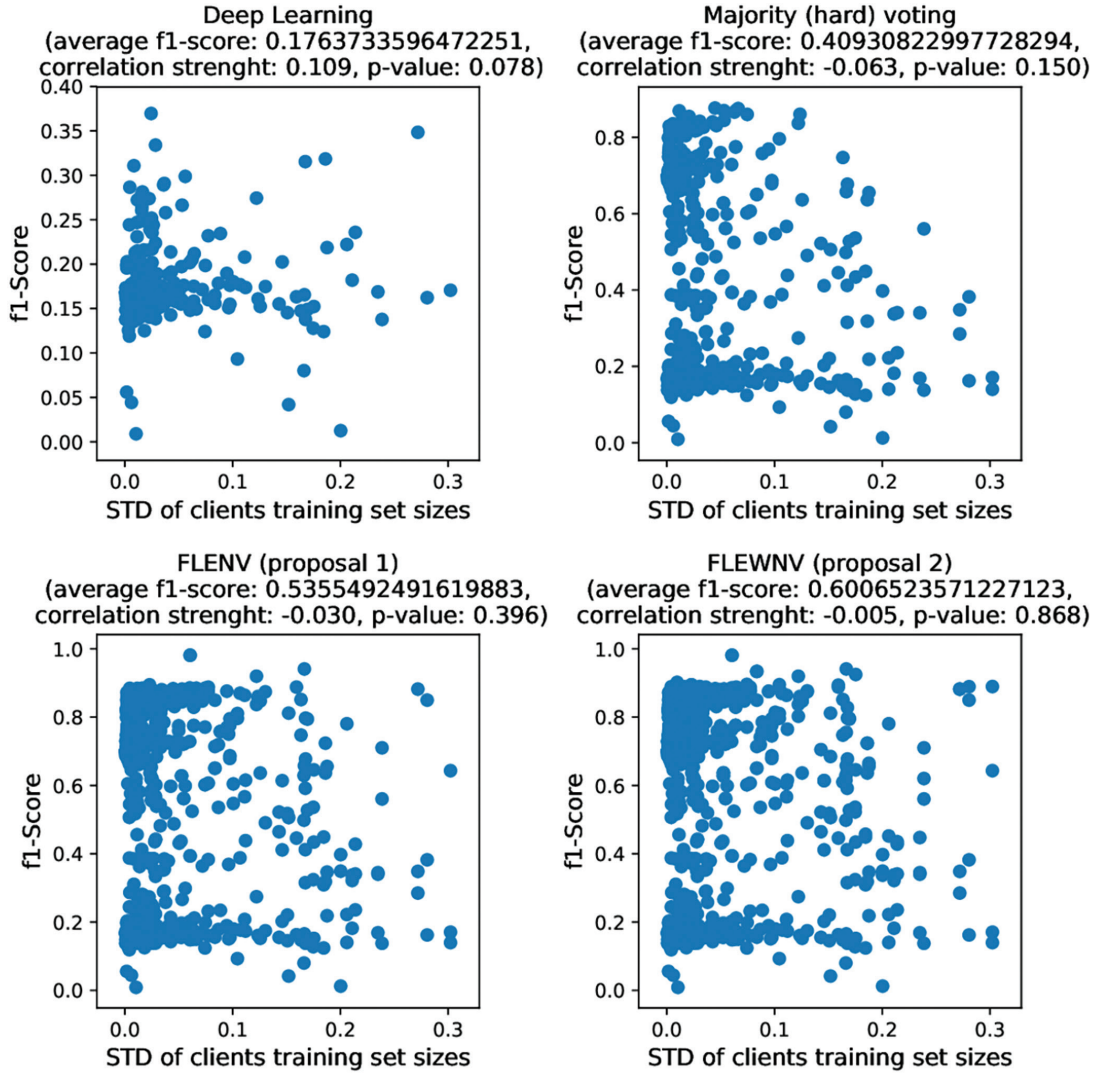


Figure 8. Correlation between client training size variation and F1-score in the BoT-IoT dataset

could easily surpass deep learning classification algorithms. In a specific scenario where an IP partitioning strategy was used in the Ton-IoT dataset, deep learning outperformed the majority voting strategy and FLENV but was outperformed by FLEWNV.

The analysis of the experiments described in this section allows us to answer RQ2. When the degree of non-IID and imbalanced data is high in a federated learning problem, the models tend to have a lower classification performance. Several metrics can be used to identify these properties, such as the achieved F1-score, the STD of the client's training set sizes, and the entropy related to the label arrays of each vector. Bot-IoT presents a higher degree of non-IID and is imbalanced compared to Ton-IoT.

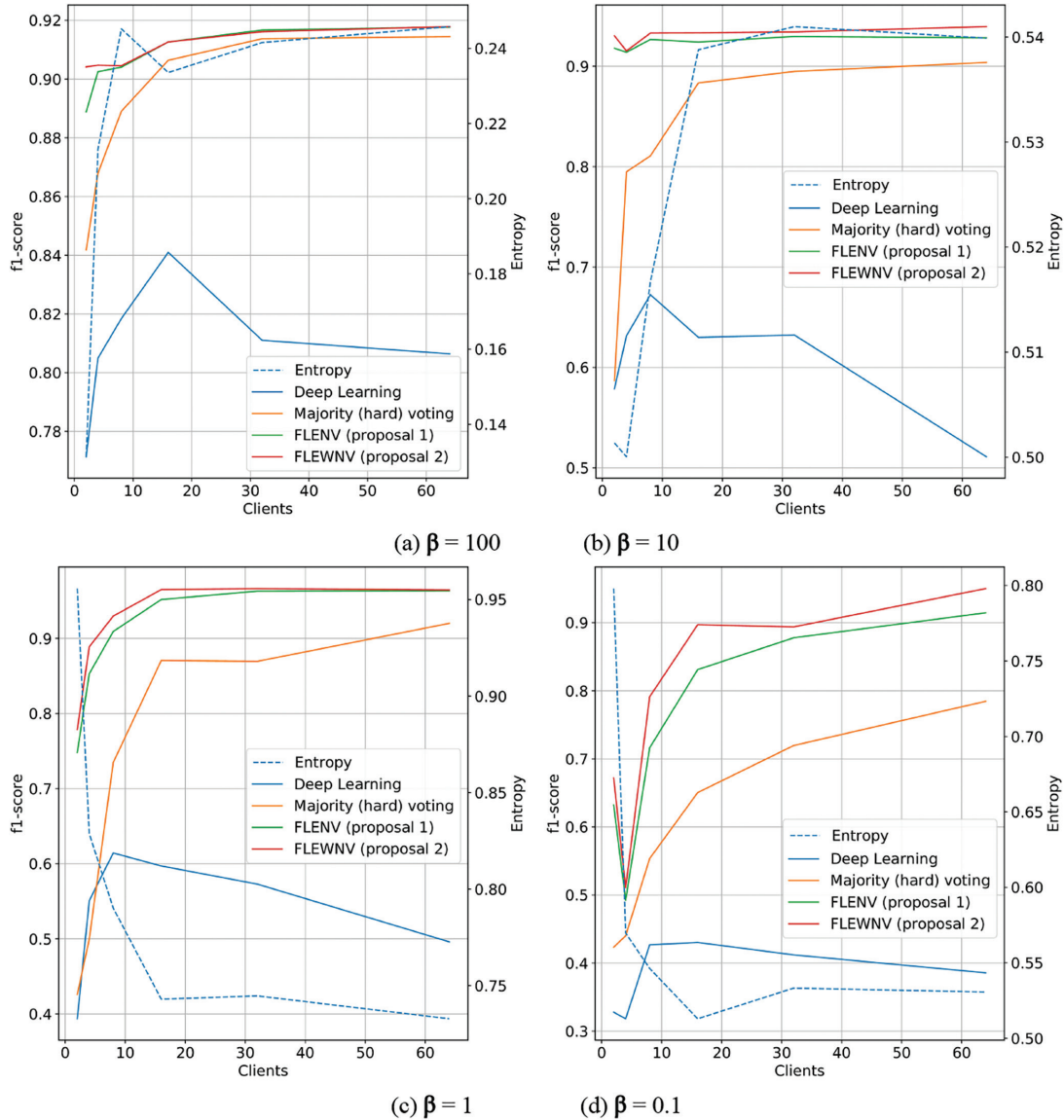


Figure 9. ToN-IoT scalability analysis assumes an increase in the total number of clients to generate several problem scenarios, where four non-IID degrees are considered, controlled by the β parameter. The F1-score is used to measure the effectiveness of each federated learning model analyzed in this article

When this degree is sufficiently high, techniques can take advantage of the knowledge related to the information of the total data used to train each federated client model.

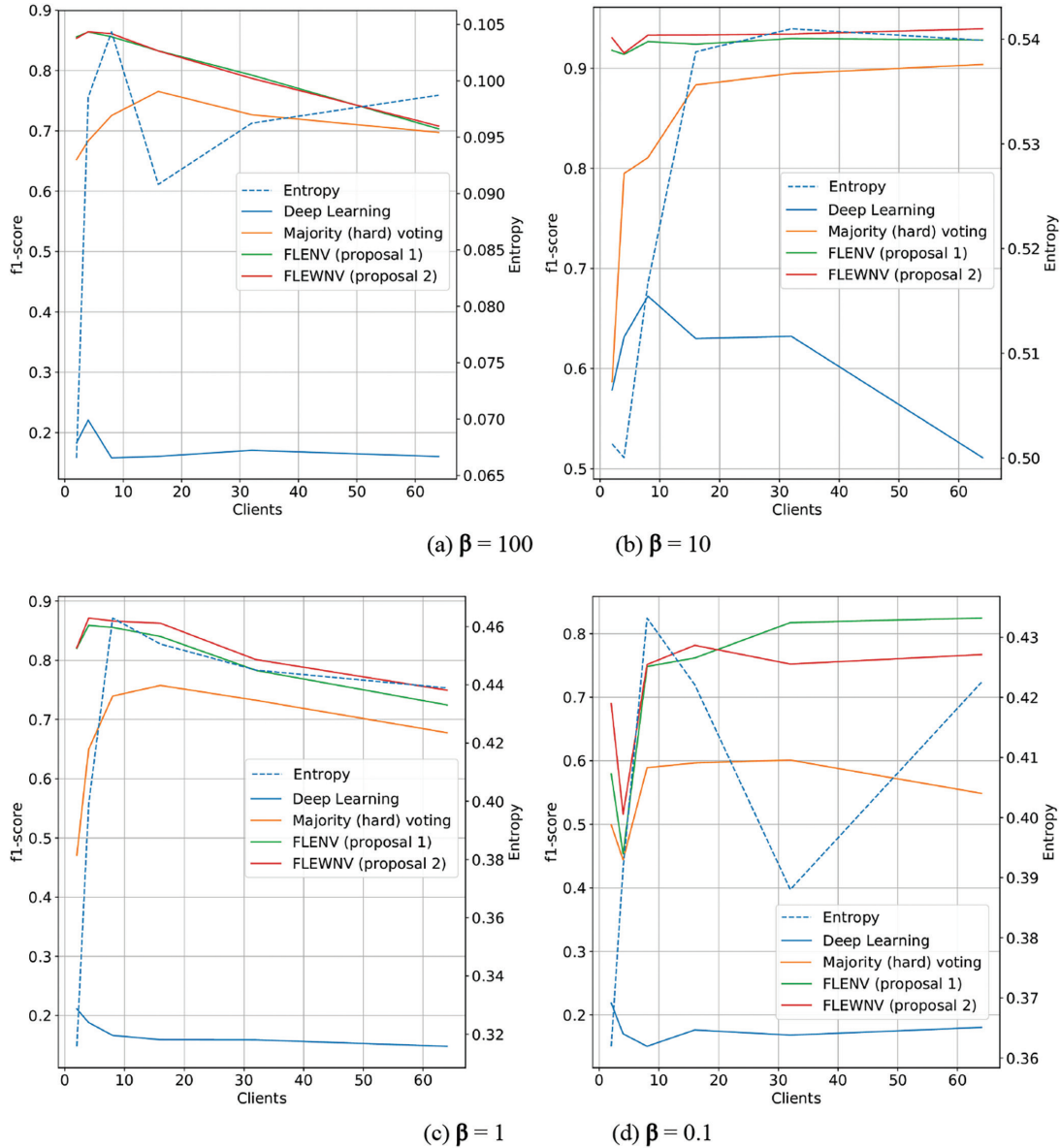


Figure 10. BoT-IoT scalability analysis assumes an increase in the total number of clients to generate several problem scenarios, where four non-IID degrees are considered, controlled by the β parameter. The F1-score is used to measure the effectiveness of each federated learning model analyzed in this article

Considering the overall average F1-scores of FLENV and FLEWNV highlighted in our experiments, it is possible that these methods achieved better classification performance than the other methods evaluated in our experiments. As can be seen in Figures 7 and 8, the FLEWNV presents a higher

F1-score average. However, in some cases, as seen in Figure 10d, FLENV obtained better results than FLEWNV.

7. Conclusions

This article presented a study concerning non-IID and imbalanced data in cybersecurity federated learning problems. Two methods, named FLENV and FLEWNV, were proposed to surpass the challenges identified in these problems. The methods were based on a federated learning framework that uses only one round of training (ORFL), where clients create a model and send it to a server that aggregates these models with ensemble learning. FLENV and FLEWNV are voting methods for an ensemble that takes advantage of the knowledge of the total data present in each client to create normalized or weighted voting.

Several experiments were conducted to identify the performance of FLENV and FLEWNV compared with two other methods: FedAvg with deep learning and the ORFL ensemble-based method using the classical majority (hard) voting strategy. The experiments focus on the capacity of these methods to achieve high classification performance. Although other characteristics may be investigated in future research, such as privacy issues, model parameter tuning, and computational speed, the proposals presented in this research enabled the advances of federated learning methods in the cybersecurity field. FLENV and FLEWNV were able to surpass other methods in the experiments, evidencing the advantage that normalized and weighted voting can improve ORFL ensemble methods. A novel procedure based on IP address was introduced for partitioning cybersecurity datasets and generating Federated Learning problems. The FLEWNV method achieved recall values higher than 0.7 for all classes in the Ton-IoT dataset and 0.197 for the Bot-IoT dataset when evaluated on the problems generated by the IP address partitioning procedure. In contrast, the majority voting strategy produced several recall values equal to 0 for different classes in both datasets, meaning that some cybersecurity attacks could not be detected. Also, the classical Deep Learning method using the FedAvg algorithm achieved an overall lower accuracy, precision, recall, and F1-score when compared with ORFL methods.

References

- Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L. F., & Abdulkadir, S. J. (2022). Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review. *Electronics*, 11(2), 198. <https://doi.org/10.3390/electronics11020198>
- Alazab, M., Rm, S. P., Maddikunta, P. K. R., Gadekallu, T. R., & Pham, Q. V. (2021). Federated learning for cybersecurity: Concepts, challenges, and future directions. *IEEE Transactions on Industrial Informatics*, 18(5), 3501-3509. <https://doi.org/10.1109/TII.2021.3119038>
- Aliyu, I., Feliciano, M. C., Van Engelenburg, S., Kim, D. O., & Lim, C. G. (2021). A blockchain-based federated forest for SDN-enabled in-vehicle network intrusion detection system. *IEEE Access*, 9, 102593-102608. <https://doi.org/10.1109/ACCESS.2021.3094365>
- Attota, D. C., Mothukuri, V., Parizi, R. M., & Pouriyeh, S. (2021). An ensemble multi-view federated learning intrusion detection for IoT. *IEEE Access*, 9, 117734-117745. <https://doi.org/10.1109/ACCESS.2021.3107337>

- Campos, E. M., Saura, P. F., González-Vidal, A., Hernández-Ramos, J. L., Bernabe, J. B., Baldini, G., & Skarmeta, A. (2022). Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges. *Computer Networks*, 203, 108661. <https://doi.org/10.1016/j.comnet.2021.108661>
- Dai, R., Zhang, Y., Li, A., Liu, T., Yang, X., & Han, B. (2024). Enhancing one-shot federated learning through data and ensemble co-boosting. *arXiv*. <https://doi.org/10.48550/arXiv.2402.15070>
- Farid, F., Elkhodr, M., Sabrina, F., Ahamed, F., & Gide, E. (2021). A smart biometric identity management framework for personalised IoT and cloud computing-based healthcare services. *Sensors*, 21(2), 552. <https://doi.org/10.3390/s21020552>
- Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access*, 10, 40281-40306. <https://doi.org/10.1109/ACCESS.2022.3165809>
- Ferrag, M. A., Friha, O., Maglaras, L., Janicke, H., & Shu, L. (2021). Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis. *IEEE Access*, 9, 138509-138542. <https://doi.org/10.1109/ACCESS.2021.3118642>
- Ghimire, B., & Rawat, D. B. (2022). Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things. *IEEE Internet of Things Journal*, 9(11), 8229-8249. <https://doi.org/10.1109/JIOT.2022.3150363>
- Guha, N., Talwalkar, A., & Smith, V. (2019). One-shot federated learning. *arXiv*. <https://doi.org/10.48550/arXiv.1902.11175>
- Hsu, T. M. H., Qi, H., & Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv*. <https://doi.org/10.48550/arXiv.1909.06335>
- Jurek, A., Bi, Y., Wu, S., & Nugent, C. (2014). A survey of commonly used ensemble-based classification techniques. *The Knowledge Engineering Review*, 29(5), 551-581. <https://doi.org/10.1017/S0269888913000155>
- Khoa, T. V., Saputra, Y. M., Hoang, D. T., Trung, N. L., Nguyen, D., Ha, N. V., & Dutkiewicz, E. (2020). Collaborative learning model for cyberattack detection systems in iot industry 4.0. *In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference* (pp. 1-6). IEEE. <https://doi.org/10.1109/WCNC45663.2020.9120761>
- Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems*, 100, 779-796. <https://doi.org/10.1016/j.future.2019.05.041>
- Leon, F., Floria, S. A., & Bădică, C. (2017). Evaluating the effect of voting methods on ensemble-based classification. *In Proceedings of the 2017 IEEE International Conference on Innovations in Intelligent Systems and Applications* (pp. 1-6). IEEE. <https://doi.org/10.1109/INISTA.2017.8001122>
- Li, J., Lyu, L., Liu, X., Zhang, X., & Lyu, X. (2021a). FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT. *IEEE Transactions on Industrial Informatics*, 18(6), 4059-4068. <https://doi.org/10.1109/TII.2021.3088938>
- Li, Q., He, B., & Song, D. (2021b). Model-contrastive federated learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10713-10722). <https://doi.org/10.1109/CVPR46437.2021.01057>

- Li, X., Huang, K., Yang, W., Wang, S., & Zhang, Z. (2019). On the convergence of FedAvg on non-IID data. *arXiv*. <https://doi.org/10.48550/arXiv.1907.02189>
- Lin, J. (2002). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1), 145-151. <https://doi.org/10.1109/18.61115>
- Lin, J. (2016). On the dirichlet distribution. *Department of Mathematics and Statistics, Queens University*, 40.
- Lin, T., Kong, L., Stich, S. U., & Jaggi, M. (2020). Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33, 2351-2363.
- Liu, Y., Liu, Y., Liu, Z., Liang, Y., Meng, C., Zhang, J., & Zheng, Y. (2020). Federated forest. *IEEE Transactions on Big Data*, 8(3), 843-854. <https://doi.org/10.1109/TBDATA.2020.2992755>
- McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *In Artificial Intelligence and Statistics* (pp. 1273-1282). PMLR.
- Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. *Sustainable Cities and Society*, 72, 102994. <https://doi.org/10.1016/j.scs.2021.102994>
- Nguyen, N. H., Nguyen, P. L., Nguyen, T. D., Nguyen, T. T., Nguyen, D. L., Nguyen, T. H., ... & Truong, T. N. (2022). FedDRL: Deep reinforcement learning-based adaptive aggregation for non-IID data in federated learning. *In Proceedings of the 51st International Conference on Parallel Processing* (pp. 1-11). <https://doi.org/10.1145/3545008.3545085>
- Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., & Talwar, K. (2016). Semi-supervised knowledge transfer for deep learning from private training data. *arXiv*. <https://doi.org/10.48550/arXiv.1610.05755>
- Schneible, J., & Lu, A. (2017). Anomaly detection on the edge. *In MILCOM 2017-2017 IEEE Military Communications Conference* (pp. 678-682). IEEE. <https://doi.org/10.1109/MILCOM.2017.8170817>
- Sewak, M., Sahay, S. K., & Rathore, H. (2018). Comparison of deep learning and the classical machine learning algorithm for the malware detection. *In Proceedings of the 2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing* (pp. 293-296). IEEE. <https://doi.org/10.1109/SNPD.2018.8441123>
- Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29.
- Wang, C., Xia, H., Xu, S., Chi, H., Zhang, R., & Hu, C. (2024). FedBnR: Mitigating federated learning non-IID problem by breaking the skewed task and reconstructing representation. *Future Generation Computer Systems*, 153, 1-11. <https://doi.org/10.1016/j.future.2023.11.020>
- Yue, K., Jin, R., Wong, C. W., & Dai, H. (2022). Federated learning via plurality vote. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6), 8215-8228. <https://doi.org/10.1109/TNNLS.2022.3225715>
- Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., & Khazaeni, Y. (2019). Bayesian nonparametric federated learning of neural networks. *In Proceedings of the International Conference on Machine Learning* (pp. 7252-7261). PMLR.
- Zhao, R., Yin, Y., Shi, Y., & Xue, Z. (2020). Intelligent intrusion detection based on federated learning aided long short-term memory. *Physical Communication*, 42, 101157. <https://doi.org/10.1016/j.phycom.2020.101157>

- Zhao, S., Liao, T., Fu, L., Chen, C., Bian, J., & Zheng, Z. (2024). Data-free knowledge distillation via generator-free data generation for non-IID federated learning. *Neural Networks*, 179, 106627. <https://doi.org/10.1016/j.neunet.2024.106627>
- Zhou, Y., Pu, G., Ma, X., Li, X., & Wu, D. (2020). Distilled one-shot federated learning. *arXiv*. <https://doi.org/10.48550/arXiv.2009.07999>
- Zhu, H., Xu, J., Liu, S., & Jin, Y. (2021). Federated learning on non-IID data: A survey. *Neurocomputing*, 465, 371-390. <https://doi.org/10.1016/j.neucom.2021.07.098>

