Pamukkale Univ Muh Bilim Derg, 31(6), 1004-1018, 2025

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

DDoS_FL: Federated learning architecture approach against DDoS attack

DDoS_FL: DDoS saldırısına karşı Federe öğrenme mimarisi yaklaşımı

Büşra Büyüktanır[1], Zeki Çıplak[2], Abdullah Emir Çil[3], Özlem Yakar[3], Mahamoud Brahim Adoum[3], Kazım Yıldız[1]*

[1]Department of Computer Engineering, Faculty of Technology, Marmara University, İstanbul, Türkiye.
busra.buyuktanir@marmara.edu.tr, kazim.yildiz@marmara.edu.tr

[2] Computer Programming, Computer Technologies, İstanbul Gedik University, Istanbul, Türkiye.
zeki.ciplak@gedik.edu.tr

[3] Department of Computer Engineering, Institute of Pure and Applied Sciences, Marmara University, İstanbul, Türkiye.
acil@marun.edu.tr, ozlemyakar@marun.edu.tr, b.mahamoud@marun.edu.tr

Received/Geliş Tarihi: 01.05.2024 Revision/Düzeltme Tarihi: 15.02.2025 doi: 10.5505/pajes.2025.40456
Accepted/Kabul Tarihi: 13.03.2025 Research Article/Araştırma Makalesi

Abstract

The frequency and complexity of DDoS attacks have significantly increased with the growth of the internet, posing severe threats to network security. Traditional machine learning and deep learning-based detection systems often face limitations due to their reliance on centralized data collection, leading to privacy concerns, high computational costs, and challenges in adapting to heterogeneous data distributions. This study proposes DDoS_FL, a federated learning-based model designed to detect DDoS attacks without requiring data sharing between devices. The model has demonstrated effectiveness under both Independent and Identically Distributed (IDD) and Non-Independent and Identically Distributed (Non-IDD) data distributions while preserving data privacy and maintaining high detection accuracy. The proposed model is trained and evaluated using the CIC-DDoS2019 dataset, which includes various types of DDoS attacks. Experimental results show that federated learning significantly reduces training time compared to traditional centralized approaches while achieving detection accuracy ranging from 82% to 97%. Furthermore, the scalability of the model is analyzed based on the number of participating clients, highlighting the advantages of its distributed nature. Comparative analyses confirm that the proposed approach is competitive in both privacy preservation and detection performance. This study demonstrates that federated learning provides an effective solution for detecting DDoS attacks and has significant potential in enhancing network security.

Keywords: DDoS attack, Federated learning, Data privacy, Deep neural network, Deep learning, Information security.

Öz

DDoS saldırılarının sıklığı ve karmaşıklığı, internetin büyümesiyle birlikte önemli ölçüde artmış ve ağ güvenliği için ciddi tehditler oluşturmuştur. Geleneksel makine öğrenimi ve derin öğrenme tabanlı tespit sistemleri, genellikle merkezi veri toplama gereksinimi nedeniyle gizlilik ihlalleri, hesaplama maliyetleri ve heterojen veri dağılımına uyum sağlama konularında sınırlamalarla karşılaşmaktadır. Bu çalışma, cihazlar arasında veri paylaşımı gerektirmeden DDoS saldırılarını tespit etmek için federe öğrenme tabanlı bir model olan DDoS_FL'yi önermektedir. Model, hem Independent and Identically Distributed (IDD) hem de Non-Independent and Identically Distributed (Non-IDD) veri dağılımlarında etkinliğini kanıtlamış olup, istemciler arasında veri gizliliğini korurken yüksek tespit doğruluğu sağlamaktadır. Önerilen model, CIC-DDoS2019 veri kümesi kullanılarak eğitilmiş ve farklı DDoS saldırı türlerine karşı test edilmiştir. Deneysel sonuçlar, geleneksel merkezi yaklaşımlara kıyasla federe öğrenmenin eğitim süresini önemli ölçüde azalttığını ve %82 ila %97 arasında değişen tespit doğruluğu elde ettiğini göstermektedir. Ayrıca, istemci sayısına bağlı olarak modelin ölçeklenebilirliği analiz edilmiş ve dağıtık yapısının avantajları ortaya konmuştur. Karşılaştırmalı analizler, önerilen yöntemin hem gizlilik koruması hem de tespit başarımı açısından rekabetçi olduğunu göstermektedir. Bu çalışma, federe öğrenmenin DDoS saldırılarının tespiti için etkili bir yaklaşım sunduğunu ve ağ güvenliğinde önemli bir çözüm olabileceğini ortaya koymaktadır.

Anahtar kelimeler: DDoS saldırısı, Federe öğrenme, Veri gizliliği, Derin sinir ağı, Derin öğrenme, Bilgi güvenliği.

1 Introduction

The use of the internet has significantly increased in recent years due to the rapid advancements in information technologies, including a growing number of users, higher bandwidth capacities, and improvements in networking technologies. As a result, there is a possibility that cyberattacks may target the internet [1]. One of the greatest threats to internet services is Distributed Denial of Service (DDoS). An attack type which avoids users from accessing target machine is called a denial-of-service attack, such as a server, by overwhelming it with requests. A DDoS attack occurs when a DoS operation is carried out on multiple machines [2]. The first DDoS attack was executed in 1999. It is one of the most prevalent and sophisticated online threats. DDoS attacks can be carried out through multiple protocols and at various stages, making them difficult to detect [3].

Automatic attack detection is achieved through the use of machine learning (ML) and deep learning (DL) [4]. Detecting and blocking traditional DDoS attacks typically involve monitoring network traffic and identifying abnormal traffic patterns. However, these methods have some limitations. Traditional intrusion detection systems running on a central server raise concerns about data privacy. It can also lead to scalability issues, as large amounts of traffic must be processed at a central point. In addition, the model developed for attack detection to produce accurate results depends on the adequacy

*Corresponding author/Yazışılan Yazar

of the data. Therefore, a lack of data leads to poor model performance in DDoS detection methods.

The handling of big data make it necessary to ensure data security. To protect data privacy, the General Data Protection Regulation (GDPR), served by the European Union and effective as of May 23, 2018, and the Personal Data Protection Law (KVKK), implemented in Turkey, are legislative measures aimed at addressing people's privacy concerns. These measures taken to ensure data privacy are insufficient in practical applications, as they only address the legal aspects of the issue. Therefore, new technological solutions must be developed.

McMahan and friends suggested a federated learning (FL) architecture to protect data privacy from a technological perspective [5]. The design specifies that each client transmits model parameters to the central server after training the model on locally generated data. Each client receives an updated model after the server aggregates the models obtained from all clients. Since model training occurs locally and only the model is transmitted to the server instead of the data, data privacy is ensured. Furthermore, as more data becomes available, the performance of the trained model improves proportionally. Network anomaly detection is another application of FL technology, which is increasingly being adopted each day [6].

The increasing frequency and complexity of DDoS attacks have posed serious security threats to data confidentiality and effective detection methods. Traditional centralized systems introduce privacy risks during data collection and processing, while also facing limitations such as scalability. This study proposes a DDoS attack detection model, called DDoS_FL, developed on the FL architecture, as a solution to these challenges. The model offers an up-to-date and effective solution by combining locally trained models from each client on the server, while preserving data confidentiality. The aim of our work is to present a scalable method for detecting DDoS attacks and to develop a more secure solution by addressing critical issues such as data confidentiality and model performance. The performance of this model is assesed using the CIC-DDoS2019 dataset [3] and experimental results are presented.

1.1 Background

DDoS attacks are among the cyber threats that have existed since the early days of the internet, but have become more complex and damaging with recent technological advancements. These attacks aim to disable targeted systems by overwhelming their resources. DDoS attacks pose a significant risk to online services, leading to financial losses, disruption of business continuity, and decreased user confidence [7, 8].

While traditional DDoS detection systems rely on a centralized data analysis structure, this approach presents several challenges, particularly for large and distributed datasets. Legal restrictions on data privacy and access to data limit the effectiveness of centralized systems. In this context, FL offers a privacy-preserving method of model training by keeping data local and only sharing model updates. Federated learning stands out as an innovative approach in DDoS attack detection, better accuracy and scalability of detection models while ensuring data privacy.

Through an assessment of the viability and efficacy of FL-based DDoS intrusion detection systems, this work aims to address the existing gaps in this field. By evaluating robustness against various DDoS attack scenarios, including both **Independent and Identically Distributed (IID) [10, 11]** and **Non-Independent and Identically Distributed (Non-IID) [10, 11]** data distributions, the proposed methodology seeks to enhance attack detection rates while reducing false alarm rates. The model's ability to perform effectively in both data distribution scenarios is crucial for ensuring scalability and robustness in real-world, heterogeneous environments.

1.2 Contributions

This work introduces an innovative approach and improvements by using an FL-based DL architecture to detect DDoS attacks. Our main contributions are as follows:

- In this work, we present an FL architecture that enables model training without requiring data aggregation at a central location, while ensuring data privacy. With this method, multiple clients can individually train models on local data before combining the model parameters on a central server.
- The FL method significantly reduces model training time by 82% to 97% compared to traditional methods. This improvement results in substantial time and resource savings, especially in applications involving large and distributed datasets. Additionally, the method proves its effectiveness in both Non-IDD data distributions, providing scalability and adaptability in real-world scenarios.
- The developed model is suitable for real-world applications in areas such as network security and IoT. Furthermore, its ability to work effectively under Non-IDD data distributions makes it highly applicable to scenarios where data is not evenly distributed among clients.
- A multi-class model is proposed for classifying different types of DDoS attacks.
- Our model performs well on large and updated dataset such as CIC-DDoS2019, demonstrating its ability to adapt to various data structures, attack types, and real-time data distributions, IDD or Non-IDD. This highlights the robustness and generalisability of the model in different environments.

The structure of the article is organized as follows: In the second section, studies on DDoS attack detection and FL are discussed comprehensively. The third section explains the FL-based DDoS attack detection model and the methods used in this context in detail. The fourth section presents the results, where the performance of the developed application is evaluated. In the fifth section, the analysis and discussion of the findings are provided under the title "Results and Discussion." Finally, in the sixth section, the findings are evaluated, and suggestions for future studies are offered. This section aims to contribute to the direction of current research and inspire readers with new research areas.

2 Related work

FL is a ML technique that provides major benefits in terms of data access and privacy since it trains locally on devices rather than requiring data collection in a central location. It lowers the possibility of privacy violations while sensitive data stays on devices and enables the creation of general models using data gathered from various devices. Education, wearable

technology, finance, healthcare, blockchain and the internet of things are just a few of the industries that use FL technology [12], which offers solutions to issues like data access and privacy [13].

This article discusses the use of FL for DDoS attack detection. In this section, past studies on federated learning, federated learning with attacks and DDoS attacks in the context of federated learning between 2019-2024 are summarised. In Figure 1, the distribution of the studies in the related databases in the fields of FL, attacks on FL and DDoS attacks in the context of federated learning is shown graphically.

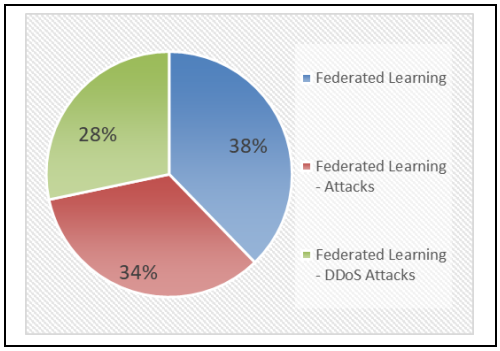


Figure 1. Graphical representation of the number of studies reviewed in the literature.

2.1 Past studies about Federated learning

Zue et al. developed a model that recognizes Chinese text in 2019. The aim of the study is to demonstrate the application potential of the FL approach through data consisting of textual images that need to be preserved. To achieve this, comparative experiments are conducted on the PySyft [14] and TensorFlow Federated (TFF) [15] frameworks. While previous similar studies used alphanumeric libraries or single-character images, the proposed model performed text recognition on a larger collection of images. The experimental results show an accuracy rate of 49.20% with TFF. Additionally, in the experiments carried out on the non-distributed dataset, an accuracy rate of 54.33% was achieved with TFF [16].

Jabłecki et al. are performed cloud-based medical image analysis using FL techniques. Two Deep Neural Network (DNN) models, ResNet50 and EfficientNetB0, are used with TensorFlow Federated, PySyft and Flower frameworks to facilitate the analysis. Comparative analyses are performed between two DNN models and three FL frameworks. Experimental results show that EfficientNetB0 outperforms ResNet50 in terms of accuracy, regardless of the parameter settings. Moreover, the accuracy of EfficientNetB0 are improved with an increasing number of local epochs, while ResNet50 are reached its highest accuracy with four local epochs. In addition, on a single client proposed models in the test set showed superior accuracy. All findings emphasize that FL is a valuable approach that not only provides a reasonable level of computational security, but also allows efficient models [17].

Yazdinejad et al. are proposed a model for the authentication of drones using Radio Frequency (RF) features and the FL method. The model, designed with the DNN method, is used together with the Stochastic Gradient Descent (SGD) optimization to enable the authentication process in drones. In the study, study was developed in the PySyft environment by using a dataset containing RF data of 1500 Phantom and 1500 Mavric type drones. The outcomes demonstrated that the suggested approach has higher performance compared to classical ML methods, with an accuracy rate of 90.7% [18].

Dasari et al. are developed a method using FL architecture to prevent unnecessary energy consumption in smart buildings. In the study, a DNN model is applied to the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) [19] dataset. The proposed method requires less data compared to similar studies conducted with ML algorithms. The ASHRAE dataset created for energy estimation consists of three-year data from more than 1000 buildings, including building (1449, 6), meter (20216100,4) and weather (139773, 9) data. The DNN model consists of a fully connected Feed Forward Neural Network (FFNN) with the Rectified Linear Unit (ReLU) activation function. The applications are implemented using the PySyft and PyTorch libraries in the Python programming language. In the experimental phase, 10, 25, 50, and 100 building data are used to make a comparison between the Centralized Learning (CL) and FL methods. Accuracy is selected as the evaluation metric. According to the experimental results, it can be said that the results obtained with FL show better performance [20].

Borger et al. are developed an application within the scope of FL architecture for the prediction of violent events in patients in a psychiatric ward in a simulated environment. The dataset [21] is used within the scope of the application is the violent event dataset created to assess the risk of violence among patients in the psychiatric ward of UMC Utrecht. Since the dataset consists of hospital data, FL architecture is adopted using Natural Language Processing (NLP) (Doc2Vec) methods to train the model with more data without compromising patient privacy. Four models are trained and compared for the application: two of them local, one of FL and one data-centralized model. The applications are implemented using the PySyft library in Python. F1-score is selected as the performance metric. When the models are tested on the combined test data the FL model obtained an F1-score value of 0.388 and the data-centric model obtained an F1-score value of 0.397. The results show that models trained with FL outperform data-centralized learning and have similar performance to data-centric models [22].

2.2 Attacks and federated learning

It is challenging to build a robust DL model and guarantee data security in intelligent intrusion detection techniques that are trained on a single client or central server. An independent and identically distributed (IID) approach based on the FL-assisted Long Short-Term Memory (FL-LSTM) architecture was developed by Zhao et al. as a solution to this issue. Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) algorithms are first compared, followed by local LSTM approaches, centralized learning-LSTM (CL-LSTM), and centralized learning-CNN (CL-CNN). The study measures the success rates of the suggested approach. The dataset utilized in the experiments consists of 10,000 command blocks (each block containing 50 commands) that involve specific attacks such as massive reads, file deletions, directory traversal attacks and batch uninstalls. To estimate the model's performance, comparisons are made based on F1-score values, accuracy, recall, and precision. The evaluation's findings indicate that the suggested FL-LSTM approach outperforms the others (acc (99.21%), pre (99.19%), rec (99.23%), and F1-score (99.21%)) [23].

Singh et al. are aimed to use Artificial Neural Network (ANN) based Autoencoder model and FL approach to find anomaly events in the data flow. These anomalies can be false or malicious entries in the transaction pool. PyTorch models and the PySyft library were used in the experimental phase. Two unsupervised anomaly detection datasets were obtained from the Harvard Dataverse. The experimental results highlighted that the federated model outperformed 97% of the full model with the F1-score and could classify all anomalies as positive. Therefore, it has been argued that this method is a better option for in-network anomaly detection [24].

Tang et al. suggested a network intrusion detection approach based on FL. Experiments were conducted out on the CICIDS2017 dataset. The dataset was trained using the Gated Recurrent Unit (GRU) DL algorithm. The experiment was implemented in Python and the PyTorch 1.3.0 DL framework. Experimental findings were evaluated using performance measures The performances of the CL and FL methods were compared in the simulation environment. According to the experimental studies, it was shown that the FL method achieved nearly the same accuracy as the centralized learning method, with an accuracy value of 97.2%. The proposed method demonstrates the applicability of the model in network intrusion detection while also ensuring data privacy in network traffic [25].

A privacy-preserving FL (PPFL) solution for Android malware detection was developed by Hsu et al. The suggested PPFL approach allows mobile devices to work together to train a classifier without disclosing private data, including authorization settings, application programming interface (API) calls, and the local model that every mobile client has learned. Secure multi-party computation techniques and Support Vector Machine (SVM) are used in the study to build the PPFL system. The performance of CL and FL architectures in various applications is compared. The experiments examined how specific attributes were used across different datasets and mobile devices. It can be concluded that the PPFL system created using FL achieved a 93% success rate and demonstrated strong malware data privacy. This study is the first privacy-conscious Android malware detection system built on the FL framework [26].

FELIDS, a FL-based intrusion detection system, is proposed by Friha et al. to ensure the security of agricultural IoT (Agri-IoT) networks. The proposed system utilizes three DL techniques: CNN, RNN, and DNN-based neural networks. Furthermore, models based on CL and FL are compared. Three different current traffic datasets are used for the FELIDS system: CSE-CIC-IDS2018 [27], MQTTset [28], and InSDN. Upon examining the experimental results, the FELIDS system applied to all three datasets performed better than the results from DL methods, but in most cases, it performed similarly to the CL model. For the FELIDS system, the corresponding accuracy rates are 98.63%, 99.71%, and 99.05%. Consequently, it can be concluded that, compared to alternative techniques, the proposed FELIDS system has the best accuracy in identifying attacks [29].

2.3 DoS/DDoS attacks and federated learning

The research indicates that DDoS attack detection is another application of FL methods. These methods are employed to trace the attack's origin and identify irregularities in network data.

In the study prepared by Siracusa and Doriguzzi-Corin in 2022, the FLAD (Federated Learning Adaptive to DDoS Attack Detection) system is presented for the detecting of DDoS attacks using distributed ML techniques. This new method detects DDoS attacks in the network by creating a training model without collecting data from multiple devices in a central location using the FL technique. FLAD is applied to the CICDDoS2019 [30] DDoS attack dataset created by the Canadian Cyber Security Institute. It consists of several days of network activity, benign network traffic, and 13 different DDoS attack types. In the FLAD system developed within the scope of the study, an F1-Score value between 0.90 and 0.97 (average 0.9667) is achieved. This work aims to advance the field of network security, particularly by using distributed ML techniques [31].

Zhang et al. conducted the FL-based FLDDoS model for DDoS attack detection. This model is based on the combination of various features (e.g., number of TCP SYN packets, UDP traffic sent to the target, etc.) for DDoS attack detection, and the learning algorithm detects attacks using these features. Autoencoder-based RNN, MLP, and CNN models are used to automatically extract features in the proposed model and improve its performance. The datasets used in the experiments are CICDDoS [32], NLSKDD [33], and CICIDS [34]. Since the data distribution in the datasets used for attack detection is unbalanced, a K-Means-based hierarchical aggregation algorithm and the SMOTEENN [35] data resampling algorithm were used as a solution to this problem. In addition, the proposed FLDDoS model is compared with the Federated Averaging (FedAvg) [36] algorithm. The experimental findings show that the suggested model gets accuracy rates for the CICIDS and NLSKDD datasets were 93.26% and 99.13%, respectively., and demonstrates good detection performance in attack detection. The results show that the proposed FLDDoS model increases the accuracy by 4% compared to traditional methods [37].

Li et al. Proposed a new architecture called FLEAM against DDoS attacks by combining FL architecture and fog/edge computing in IIoT devices. In this architecture, Iterative Model Averaging (IMA)-based GRU models are developed to overcome various attacks emerging in IIoT. The IMA-GRU model performs accurate detections on distributed data. The UNSW NB15 dataset [38] is used for the applications. Compared to classical solutions, the proposed FLEAM architecture has about 72% lower DDoS attack mitigation time, while achieving about 47% higher DDoS attack mitigation accuracy. Finally, the evaluation of the IMA-GRU model shows that the accuracy in DDoS attack detection based on the UNSW NB15 dataset is about 98% on FL, which is almost the same as the accuracy achieved with centralized training [39].

The Weighted Federation Learning (WFL) model is suggested by Ali et al. for the identification of Low Rate-DDoS (LR-DDoS) attacks. To ensure the success of the proposed model, three distinct ANN training procedures were employed which Bayesian Regularization (BR), Scaled Conjugate Gradient (SCG) and Levenberg-Marquardt (LM) algorithms. The tests were conducted using the CAIDA dataset [40]. The applications used MATLAB 22 software. With a classification accuracy of 98.85%, the WFL model is the most used model approach for intrusion detection systems, according to the test results [41].

Fotse et al. Proposed a FL scheme named FedLAD for the detection of DDoS attacks in large-scale Software-Defined

Networks (SDN). The accuracy assessment for DDoS attack detection in the FedLAD approach is performed with three different aggregation techniques: FedAVG, Astraes, and Ranking Client techniques. These techniques are used to combine the new models participating in the FL process. The FedLAD scheme is evaluated on the CICDoS2017 [42], CICDDoS2019, and InSDN [43] datasets. Oracle VirtualBox Manager 6.0 is used to simulate the experiment. The experimental results show that the proposed FedLAD method has an accuracy of approximately 98% (FedAVG: 92.62%, Astraes: 97.54%, Ranking Client: 97.64%) across all three techniques compared to related studies. This work presents a new technique for DDoS attack detection in SDN using the FL approach [44].

Zainudin et al. proposed a unified model addressing FL-based DDoS classification to ensure the security of SDN-based IIoT networks. In the study, comparisons are made using FL-based Chi-square FS and FL-based feature selection (FS) techniques, along with the Pearson correlation coefficient (PCC). In addition, the FedAvg algorithm is used to calculate the training parameters collected from SDN. An FL-based CNN-MLP model is also proposed for DDoS classification. Applications for DDoS attack detection are carried out on the CICDDoS2019 dataset. Comparisons are made with CNN, Multi-MLP, Existing CNN-MLP, and the proposed FedDDoS models for FL-based DDoS classification. Analyzing the results reveals that a calculation time of 3.917 ms yields an accuracy of 98.37% [45].

Lee et al. proposed a personalized FL-based DDoS attack detection model using the DBSCAN clustering (FedDB) method. LSTM models are combined with the FL framework FedMe framework [46] for model training; DBSCAN clustering is used for model clustering and modification. In this model, DBSCAN clustering improves the overall detection accuracy by addressing data distribution imbalances and also enhances the proposed method. The CICDDOS2019 dataset is used for the evaluation of the proposed approach. When alpha = 0.2 in the dataset, certain DDoS attack types are not included in the data of each client. On the other hand, when alpha = 0.9, they are included as DDoS attack types in the data of each client. According to the experimental results, when the performance evaluation of the proposed FedDB model using FedAvg, FedMe, CL, and FL methods is examined, when alpha = 0.9, the FedDB method showed accuracy values of 0.95, and when alpha = 0.2, the FedDB method showed accuracy values of 0.97. As a result, the proposed FedDB approach protects data privacy by increasing model accuracy as a solution to data distribution imbalances in DDoS attack detection [47].

Table 1 presents a comparison of the functionality provided by literature solutions and current studies. This study aims to overcome the limitations of traditional ML and DL techniques, such as privacy risks and insufficient training performance due to the necessity of centralized data collection. The DDoS_FL model was developed as a FL-based architecture, enabling the detection of DDoS attacks without requiring data sharing between devices.

Table 1. State of the art works.

Year [Ref.]	Application	FL Architecture	Method	Dataset(s)	Key_findings
2025 [44] Fotse et al.	DDoS attack detection in SDN	FedLAD	FedAVG, Astraes, Ranking Client	CICDoS2017, CICDDoS2019, InSDN	Deep learning models achieve 98% accuracy, but they require longer training times and higher resource allocation as network scales expand.
2024 [47] Lee et al.	DBSCAN clustering based DDoS attack detection	FedDB	FedAvg, Fedme, CL, FL	CICDDOS2019	The K-means method, with an accuracy of 95% for α=0.9 and 97% for α=0.2, tends to include outliers in clusters when dealing with unevenly distributed data.
2023 [40] Ali et al.	LR-DDoS attack detection	WFL	LM, BR, SCG	CAIDA	The WFL model, utilizing Bayesian Regularization, Scaled Conjugate Gradient, and Levenberg-Marquardt algorithms, achieved 98.85% accuracy in detecting Low-Rate DDoS attacks on the CAIDA dataset. It stands out as a widely adopted approach for intrusion detection systems.
2022 [45] Zainudin et al.	FL-based DDoS classification for security of SDN-based IIoT networks	FedDDoS	FedDDoS, CNN, Multi-MLP, Existing CNN-MLP, FedAvg	CICDDoS2019	The FL-based CNN-MLP model proposed by Zainudin et al. Achieved 98.37% accuracy in DDoS classification with a computation time of 3.917 ms on the CIC-DDoS2019 dataset. Comparisons with various models highlight its effectiveness in securing SDN-based IIoT networks.
2022 [31] Siracusa & D.Corin	DDoS attack detection	FLAD	FedAvg, FLAD, FLDDoS	CIC-DDoS2019	The proposed model attains an F1-score between 0.90 and 0.97, yet it lacks a test set to assess its performance against various attack types. Additionally, the FEDAVG method struggles with imbalanced and Non-IID data.
2021 [37] Zhang et al.	FL-based DDoS attack detection	FLDDoS	RNN, MLP, CNN, K-Means-based hierarchical aggregation algorithm, data resampling algorithm, FedAvg	CICDDoS, NLSKDD, CICIDS	The FLDDoS model by Zhang et al. Achieved 93.26% accuracy on the CICIDS dataset and 99.13% accuracy on the NLSKDD dataset, improving detection performance by 4% compared to traditional methods. Using autoencoder-based RNN, MLP, and CNN models, along with K-Means-based hierarchical aggregation and SMOTEENN resampling, the model effectively handles imbalanced data in DDoS attack detection.
2021 [39] Li et al.	DDoS in industrial IoT	FLEAM	IMA, GRU, FL	UNSW NB15	With 98% accuracy, the model faces challenges due to the diversity of DDoS attacks, the complexity of Industrial IoT environments, and the inherent limitations of federated learning.
Our Proposed Model Buyuktanir et al.	FL based DDoS attack approach	DDoS_FL	FL, DL,	CIC-DDoS2019	This paper addresses the limitations of traditional ML and DL methods by introducing the DDoS_FL model, which enables DDoS detection without data sharing by providing both high accuracy (82-97%) and data privacy. The model achieves an F1-score between 0.89 and 0.99, significantly reduces the training time and effectively detects various types of DDoS attacks as verified on the CIC-DDoS2019 dataset.

Its most important difference compared to the existing literature is that it provides high accuracy (82 to 97 percent) along with the data privacy preservation feature and significantly reduces training times. Experiments conducted with the CIC-DDoS2019 dataset show that the model can successfully detect various types of DDoS attacks. This study aims to introduce the FL approach to the literature as a promising solution for effective DDoS attack detection while preserving data privacy.

The use of FL techniques against DDoS attacks helps make the learning models used to detect the attack more up-to-date and accurate. In addition, FL also helps protect distributed data and provides better scalability, as it does not require a central server. According to literature research, FL techniques are seen as a promising method for developing defense mechanisms against DDoS attacks. However, more research needs to be conducted, and further testing of how the techniques work in real-world scenarios is required.

3 Proposed approach

This section contains the detailed methods of the model developed for detecting DDoS attacks using the FL method. FL is a distributed learning approach that provides training on local devices instead of a central server to protect data privacy. The goal of this work is to create a more safe and scalable approach for identifying DDoS attacks using FL.

3.1 Dataset and data preprocessing

Before the model training, preprocessing steps were performed on the raw dataset. In order to implement the DL model, the ready dataset is first loaded into the system.

Within the scope of the study, the DDoS Evaluation Dataset (CIC-DDoS2019), an up-to-date and well-designed dataset shared by the Canadian Institute for Cybersecurity, was used to detect DDoS attacks and classify attack types [3].

The CIC-DDoS2019 dataset consists of a total of 79 attributes (columns) and 431,371 (rows) observations. Two datasets were derived from the raw dataset to be used in Binary Classification (BC) and Multiclass Classification (MC). The BC dataset was used for DDoS attack detection, and the MC dataset was used for DDoS type classification.

First, 12 features that were deemed unnecessary due to their lack of contribution to model training were removed from both datasets. The deleted attribute names are: "FIN Flag Count","Bwd PSH Flags", "ECE Flag Count", "Fwd URG Flags", "Bwd URG Flags", "PSH Flag Count", "Bwd Avg Packets/Bulk", "Fwd Avg Bytes/Bulk", "Fwd Avg Packets/Bulk", "Fwd Avg Bulk Rate", "Bwd Avg Bytes/Bulk" and "Bwd Avg Bulk Rate". As a result, a total of 67 features remained in the raw dataset.

The "Label" and "Class" attributes in the raw dataset serve as identifiers for the BC and MC datasets, respectively. The "Class" attribute contains values of 0 and 1, while the "Label" attribute contains 17 different DDoS types and the "Benign" value. Subsequently, the BC dataset was created first, followed by the MC dataset.

Since the BC dataset was created to detect attacks on network traffic, the "Class" attribute was used as the target variable in the models trained with this dataset. In the dataset, "Benign" is labeled as "0", and other attacks are labeled as "1". Table 2 shows the distribution of the "Class" attribute in the BC dataset.

Table 2. Class names and numbers in the BC dataset.

Class	Count
Bening / "0"	97831
Attacks / "1"	333540

Table 2 shows that the different "Class" values are unbalanced in the dataset. The "Label" attribute in the raw dataset contains information about "Benign" and "DDoS types". If the "Label" value is 0, the "Class" value is definitely 0. For other values that the "Label" takes, the "Class" value is definitely 1. In this case, the other attributes in the dataset have no significance or effect. A prediction model built on this data could make a decision about the "Class" simply by looking at the "Label" attribute. To avoid this issue, the "Label" attribute was also removed from the BC dataset.

Thus, the models created with the BC dataset (Models 1, 2, 3, 4, 5) were trained using the remaining 66 features, with 65 features as independent variables and 1 feature as the target variable.

Since the MC dataset was created to classify DDoS attack types, the "Label" attribute was used as the target variable in the models produced with this dataset. The "Label" attribute was converted into a column containing numeric values from 0 to 17 using Label Encoding. "Benign", which represents normal traffic in the dataset, is labeled with a value of "0". The dataset contains 17 different attack types, which are labeled from "1" to "17". The "Class" attribute was not removed from the MC dataset. The value of the "Class" attribute is 0 when the "Label" value is also 0. In this context, the "Class" attribute provides important information for predicting one of the multiple classes.

In other words, when the "Class" attribute takes the value 0, a corresponding class of the "Label" is determined. However, when the "Class" attribute takes the value 1, the value of the "Label" cannot be determined by looking at the "Class" attribute alone. Therefore, the "Class" attribute is not removed from the MC dataset, as it carries partial information.

Thus, the models created with the MC dataset (Models 6, 7, 8, 9, 10) were trained using the remaining 67 features, with 66 features as independent variables and 1 feature as the target variable.

Figure 2 shows the names of the attack types in the MC dataset and the number of occurrences of each in the dataset. As seen in Figure 2, the attack types are unbalanced in the dataset.

In both datasets, Z-Score Normalization (Standardization) [48] process was applied, which transforms the values in the columns of the attributes other than the target variables into a standard distribution using the mean and standard deviation. Thus, different scales and distributions in the dataset were eliminated and it was aimed to improve the model performance. After all these preprocessing steps were completed, the datasets were made ready for the training of the model.

3.2 Federated learning

FL, which has recently gained popularity in the field of ML, is used in scenarios where data does not need to be collected in a central location due to privacy and security concerns [49, 50, 51]. After training models on their local datasets, data owners (clients) distribute the modified parameters of the learned models with a central server.

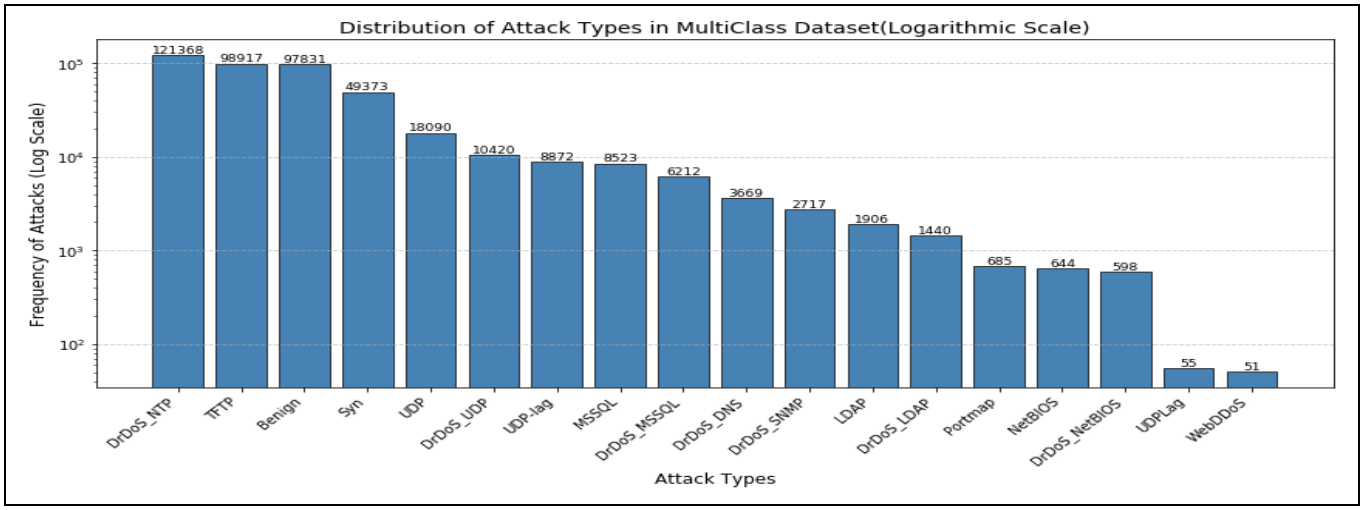


Figure 2. Distribution of attack types in the Label attribute of the MC dataset.

In this method, data owners can contribute to the final model generated on the server without sharing their data. FL is suitable for and used in many application areas that require security and confidentiality, such as financial data, customer data, and patient data. Figure 3 illustrates the working architecture of both the classical learning method and the FL method for systems with clients and servers. In Figure 3, the model training steps for both methods are numbered.

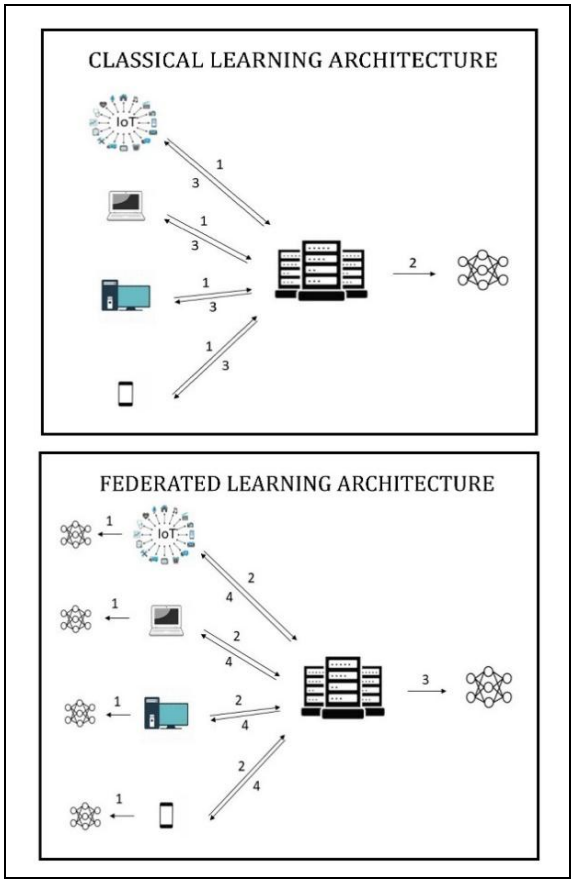


Figure 3. The working architecture of classical learning method and FL method.

In the conventional technique, clients send their data to the server in the first phase, the server uses the incoming data to train the model in the second phase, and the trained model is then distributed to each client in the third phase. For the FL method; The first step is model training with the data on the clients, the second step sending the models to the server, the third step is the merging of the models on the server, and the fourth step is the distribution of the current model to the clients. Proposed models were trained and their performances were compared by using the DDoS dataset, classical method, and FL method.

3.3 DDoS_FL

Within the scope of the study, a DDoS attack detection model named DDoS_FL based on FL architecture was designed. Since the FL architecture is implemented in systems with clients and server, the success of DDoS_FL models according to different client numbers is measured by comparing them with the model developed using the classical learning method. All models are developed with DNN models. Virtual clients/devices were created in the simulation used to set up the DDoS_FL architecture. To observe the operation with different client numbers, the operations were repeated for client numbers assumed to be 5, 10, 50, and 100. In model training, the BC and MC datasets, obtained by editing the CIC-DDoS2019 dataset, were used. Firstly, the aim was to distribute the data evenly across the clients, each of which was evaluated individually, assuming the numbers to be 5, 10, 50, and 100, respectively. To achieve this, 371 samples were removed from the entire dataset, reducing the total number of data points from 431,371 to 431,000. The 371 extracted samples were set aside as test data. Then, 10% of the 431,000 data points were reserved as test data. Thus, a total of 10% + 371 data points were used for testing. The data allocated for training were equally distributed among 5, 10, 50, and 100 users. In the simulation environment, firstly, since observations will be made using the IID data distribution, the data are equally distributed. For the Non-IID data distribution, a total of 10% plus 371 data points were used for testing purposes. The training data was divided into three repeated distributions with different proportions between 5, 10, 50 and 100 clients and the results obtained under these different distributions were analysed.

Data distribution to virtual clients is done using TensorFlow [52]. The records in the training data are assumed to be generated on different clients to ensure an IID data distribution and are randomly partitioned based on the number of clients. For instance, if the number of clients is five, the training data is randomly divided into five equal parts. To achieve a Non-IID data distribution, 80% of the classes in the data are randomly distributed within the same class, while 20% are assigned to different classes. In other scenarios, 65% of the classes are kept within the same class, and 35% are distributed across different classes, in another case, 50% are the same while the other 50% are assigned to different classes. Additionally, the data owned by users in each round is mixed. In each round, nearly different data is assigned to each user. Thus, a simulation environment is created in which devices regenerate data and reproduce models with that data, aiming to closely resemble real-world conditions.

On the other hand, time measurements were taken in each round. After the rounds were completed, the total time spent by the model was calculated, allowing for comparison with other models and non-federated models.

3.4 Model architectures, systems used and libraries

Experiments are processed on Windows 10 OS, an Intel Core i7-12650H CPU 2.30 GHz processor, 16GB RAM, 512GB SSD, and an NVIDIA GeForce RTX 3060 Laptop GPU. Jupyter Notebook was chosen as the development environment, and Python 3.10 was used as the programming language. For the DL model, an experimental environment was set up using the libraries TensorFlow [52], Keras [53], Pandas [54], and Scikit-learn [55].

For BC and MC, two models were developed primarily using the classical approach, without the FL approach. These are Model 1 and Model 6. Then, the versions of these models created with the FL approach (Models 2, 3, 4, 5, and Models 7, 8, 9, 10) were compared with Model 1 and Model 6. Two separate DL architectures were designed for binary and multiple classifications. The DL architecture used for BC is called Binary Architecture Deep Neural Network (BA-DNN), and the DL architecture used for MC is called Multi Architecture Deep Neural Network (MA-DNN).

There are 65 neurons in the input layer of the BA-DNN architecture. Each of these neurons corresponds to an attribute in the dataset that the model will take as input. The ReLU activation function is used in the input layer. It converts values below zero to zero, while leaving values above zero unchanged [56]. This function is generally preferred in DL models because it speeds up the training process and requires less computation. BA-DNN has two hidden layers, each containing 35 neurons and the ReLU is used. Additionally, the dropout technique was applied to each hidden layer to prevent overfitting [57]. Dropout sets the activations of randomly selected neurons in the specified percentage (0.3 in this example) to zero. Finally, there is a single neuron forming the output of the model. Here, the sigmoid activation function is used [58]. The sigmoid function converts the model's output to a value between 0 and 1. This function is generally used in binary classification tasks because it guarantees that the model's output can be understood as a probability value. Figure 4 shows the BA-DNN architecture.

MA-DNN shows similarities to BA-DNN architecture in many ways. There are 66 neurons in the input layer. The neurons in the input layer correspond to each seperate variable in the dataset to be used for MC. The ReLU is used in the input layer as an activation function.

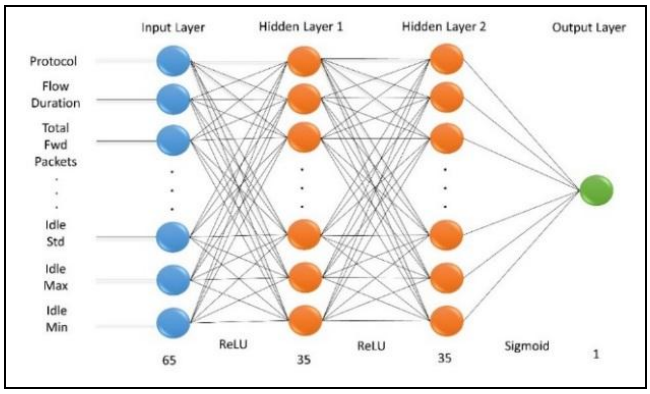


Figure 4. BA-DNN architecture.

As in BA-DNN, there are two hidden layers, each containing 30 neurons. Again, after each hidden layer, the dropout technique was applied to prevent overfitting. The dropout rate is set to 0.1 in this architecture. Finally, there are 18 neurons that form the output of the model. The softmax activation function is used here [58]. It converts each input item into a range of 0 to 1. It is generally used in multiclass classification problems to convert the outputs in the last layer into class probabilities. was utilized as an optimizer for both datasets to minimize the model's loss function and speed up the training process by updating the parameters [59]. Figure 5 shows the MA-DNN architecture.

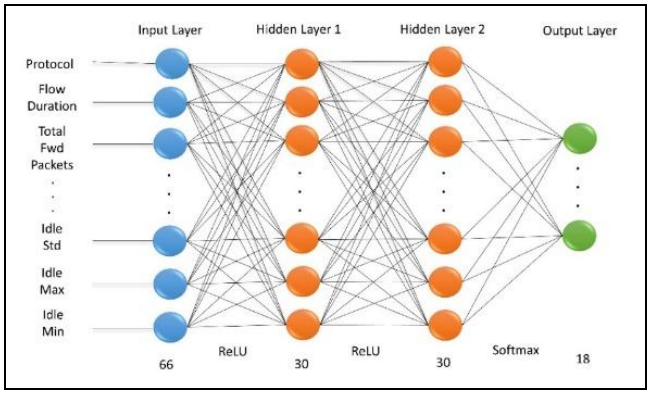


Figure 5. MA-DNN architecture.

4 Experimental results

This section discusses the experimental setup for the model created using the FL method for DDoS attack detection and classification, the evaluation criteria for the outcomes, and the experimental results of testing the developed DNN model on the dataset in question according to IID and Non-IID data distribution. Accuracy, precision, recall and F1-score measures are carried out in this study to get the performance of model. These metrics, commonly used in the literature, give a complete evaluation of the model's classification performance [60, 61]. Particularly in an FL design where the data distribution may be uneven, each of these measures emphasizes distinct aspects of the model and is crucial for ensuring its reliability. Sensitivity shows how well the model detects all positive cases, accuracy indicates overall success, precision measures how well the model controls false positives, and F1-Score shows how well precision and sensitivity are balanced in positive predictions.

The model developed with classical learning and the models developed with FL were compared. Table 3 shows the designed model information and comparison results. When the results in Table 3 are examined, in identifying the existence of DDoS in the network, for BC, it was observed that the performance of the model trained with the traditional method (Model 1) and the models trained with the FL method (Models 2 and 3) had the same accuracy. When the metrics of Models 4 and 5, where the number of users is higher, are examined, it can be concluded that the FL method slightly reduces the model performance compared to the classical method. However, since this decrease is 1/1000, it is a reduction that can be ignored, considering the advantages of FL. For MC, the performance of the classical model (Model 6) and the other FL-based models (Models 7, 8, 9, and 10) were very close to each other, although there was a slight decrease. It is not unusual for federated models to be somewhat less efficient than classical learning models in general.

When the DDoS types in the dataset used for MC are examined, it is observed that there are 17 different types, which are unevenly distributed in the dataset. This imbalance led to a decrease in the performance of the MC models. Among of this some DDoS types have very few data points in the dataset. For example, while there are 121,368 data points belonging to the DDoS type named DrDoS_NTP, there are only 51 data points for the DDoS type named WebDDoS. It is likely that the trained DL model cannot effectively learn the characteristics of DDoS types with a small number of data points, leading to incorrect predictions.

According to Table 3, the confusion matrix for the optimal FL models Model 2 for BC and Model 7 for MC are presented. The confusion matrix for Model 2 and Model 7 are seen in Figure 6 and Figure 7 seperately. These visualizations offer a detailed overview of the classification performance for each respective model.

When the model training times in both datasets are evaluated, it is seen that all FL-based models are shorter than the training times in the models developed with the classical method Figure 8. Shows the variation of the training times according to the number of users of the FL-based models for the BC and MC datasets.

In this study, three different comparable scenarios are constructed in order to systematically analyse the impact of Non-IID data distribution on federated learning. Theoretically, many different Non-IID scenarios can be designed. However, testing all possibilities would not only exceed the study time, but also make it difficult to compare the results. Therefore, for the purpose of interpreting the Non-IID effect, three scenarios with a gradual change in the class distribution were determined for a balanced and comparable structure.

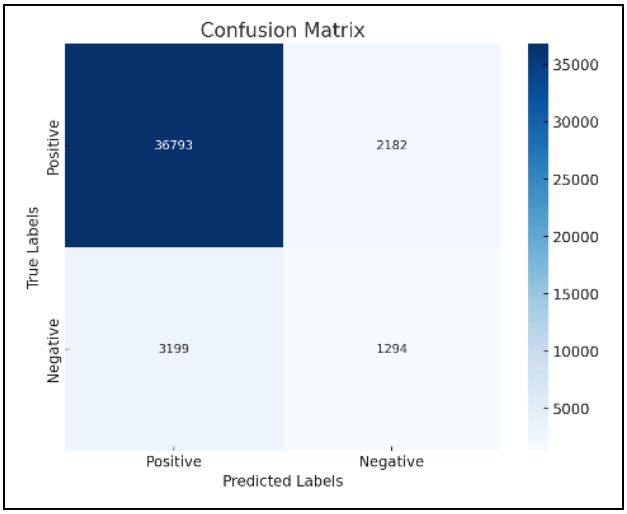


Figure 6. The Model 2 confusion matrix.

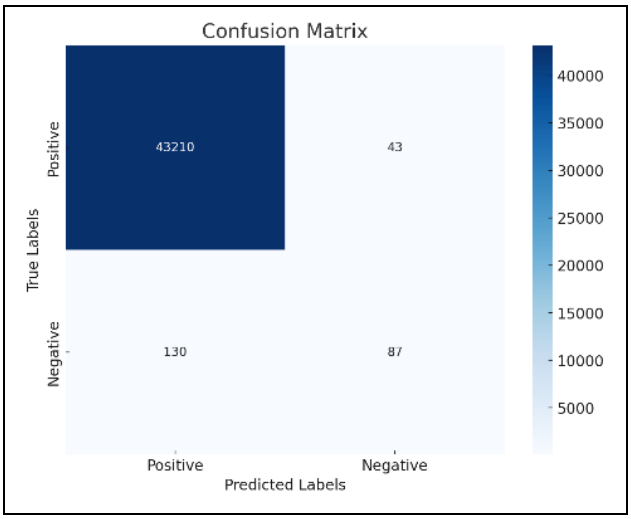


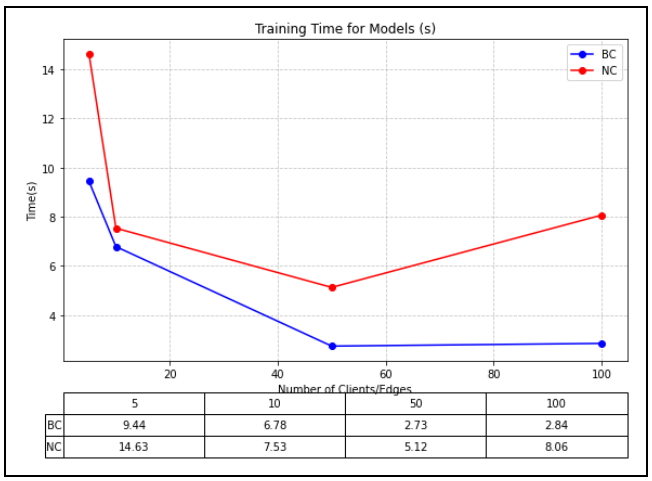
Figure 7. The Model 7 confusion matrix.

Table 3. Evaluation of the models trained on the CIC-DDoS2019 dataset. (IID).

No	Model	Architecture	Users	Epochs/Rounds	Elapsed Time(s)	Accuracy	Precision	Recall	F1-score
1	Non-Fed + BC		1	10	84.35	0.997	0.999	0.997	0.998
2	Fed + BC		5	5	9.44	0.997	0.999	0.997	0.998
3	Fed + BC	BA-DNN	10	6	6.78	0.997	0.999	0.996	0.998
4	Fed + BC		50	5	2.73	0.996	0.998	0.996	0.997
5	Fed + BC		100	6	2.84	0.996	0.998	0.996	0.997
6	Non-Fed + MC		1	10	82.03	0.932	0.954	0.932	0.916
7	Fed + MC		5	7	14.63	0.920	0.944	0.920	0.901
8	Fed + MC	MA-DNN	10	7	7.53	0.919	0.942	0.919	0.902
9	Fed + MC		50	7	5.12	0.912	0.921	0.912	0.891
10	Fed + MC		100	12	8.06	0.912	0.909	0.912	0.891

BC: Binary Classification, **MC:** MultiClass Classification, **Fed:** Federated Approach,

BA-DNN: Binary Architecture Deep Neural Network**, MA-DNN:** Multi Architecture Deep Neural Network.



	5	10	50	100
BC	9.44	6.78	2.73	2.84
NC	14.63	7.53	5.12	8.06

Figure 8. The variation of the training times according to the number of users of the FL-based models for the BC and MC datasets. (IID).

In the first scenario, 80% of each client's data was randomly distributed from a single class and the remaining 20% from other classes. In the second scenario, this was reduced to 65%, so that clients were exposed to more class diversity. In the last scenario, the classes were distributed completely evenly (50% from a dominant class and 50% mixed from other classes), creating a more balanced structure in the training process of the clients. This configuration allows a direct comparison of the impact of different data distributions on federated learning to understand the progressive impact of the Non-IID level. Using the same model architectures for each scenario, we retrained with clients with different data distributions and analysed the results in detail. This approach contributes to the step-by-step evaluation of the impact of different Non-IID structures on model performance and to understand how different situations affect the federated learning process.

The experimental results in Table 4 clearly demonstrate the effects of the Non-IID data distribution on federated learning. In the binary classification (BC) task, the effect of Non-IID is quite low. One of the most important reasons for this is that the BC problem is a somewhat simpler task. Since the model only needs to discriminate 'is there an attack or not?', even users with a predominance of a single class do not affect the overall model much. Therefore, even in the Non-IID system for BC, the accuracy and other metrics are quite close to the IID system.

On the other hand, in the multiple classification (MC) task, the effect of Non-IID was much more pronounced. While the accuracy was 0.920 for 5 users in the IID system, it decreased to 0.802 in the Non-IID system. This is mainly due to the fact that some of the clients focus heavily on a single class. When the centralised model combines these imbalanced learnings during the training of federated learning, some classes are learned extremely well, while others are hardly learned at all. As a result, for small numbers of users, the Non-IID effect further destabilised the overall balance of the model. The decrease in F1-score supports this. The decrease in F1-score indicates that the false positive rate and false negative rate of the model increased in some classes.

However, as the number of users increased, the generalisation ability of the central model increased and the accuracy increased. For example, the Non-IID model with 100 users experienced an accuracy decrease of only 0.02 compared to the model with 100 users in the IID system (IID: 0.912, Non-IID: 0.892). This is due to the fact that with 100 users, the centralised model is able to get information from a wider perspective of different clients and thus generalise. The effect of clients focusing too much on a single class during the federation process is balanced with the data from more clients, allowing the model to learn more comprehensively [62].

It is clearly seen that Non-IID increases the training time for small numbers of users. Especially for 5 and 10 users, while the training time was 14.63 and 7.53 seconds in the IID system, it increased to 24.13 and 10.98 seconds in Non-IID. The reason for this is that the model makes more errors in weight updates during the federation process and requires more rounds due to the high differences between local models. However, the training times for 50 and 100 users were almost the same as IID. This is due to the fact that updating the centralised model becomes more stable as the number of clients increases. When there are many clients, the impact of the data of different users becomes more balanced and the negative impact of Non-IID is reduced [63].

When the accuracy values of Models 7, 8, 9 and 10 are analysed, it is observed that the accuracy increases as the number of users increases. In Model 7 (5 users) the accuracy was 0.802, in Model 8 (10 users) 0.849, in Model 9 (50 users) 0.885 and in Model 10 (100 users) 0.892. In other words, the accuracy of the model gradually increased as the number of users increased.

Table 4. Evaluation of Non-IID models trained on the CIC-DDoS2019 dataset with 80% of the classes in the data being the same and 20% randomly distributed as other classes.

No	Model	Architecture	Users	Epochs/Rounds	Elapsed Time(s)	Accuracy	Precision	Recall	F1-score
1	Non-Fed + BC		1	10	84.35	0.997	0.999	0.997	0.998
2	Fed + BC		5	3	5.8	0.996	0.999	0.996	0.998
3	Fed + BC	BA-DNN	10	11	12.31	0.997	0.999	0.996	0.998
4	Fed + BC		50	6	3.55	0.996	0.999	0.996	0.997
5	Fed + BC		100	4	2.12	0.995	0.998	0.996	0.997
6	Non-Fed + MC		1	10	82.03	0.932	0.954	0.932	0.916
7	Fed + MC		5	28	24.13	0.802	0.935	0.802	0.773
8	Fed + MC	MA-DNN	10	16	10.98	0.849	0.898	0.849	0.844
9	Fed + MC		50	8	3.72	0.885	0.865	0.885	0.865
10	Fed + MC		100	14	5.9	0.892	0.880	0.892	0.870

The main reason for this increase is that as the number of users increases, the negative effects of the Non-IID distribution are eliminated. In the first scenario, 80% of each user's data is randomly selected from a single class and 20% is randomly selected from other classes. With a small number of users, the centralised model performed unbalanced learning due to some clients over-focusing on certain classes. Especially in Model 7 (5 users), since there were very few clients, the class imbalance of each client severely affected the overall performance of the model. This resulted in the model generally learning some classes very well but ignoring others, and the accuracy dropped significantly. As the number of users increased, the centralised model combined data from more clients and was better able to balance between different classes.

Another reason is that the FedAvg algorithm can perform a more balanced update as the number of users increases [64]. When the number of users is low, the weights of some clients may influence the central model more and cause certain classes to become dominant. However, when the number of users increases, each client's model has a smaller impact and the central model is updated with a broader data perspective. This allows the model to minimize the Non-IID effect and develop a more balanced decision mechanisim.

When the results in Table 5 are compared with the results of the IID system in Table 3 and Table 4 (the first Non-IID scenario), the effects of the Non-IID distribution on the accuracy, training time and generalisation capacity of the model become clearer. Unlike the first Non-IID scenario, the data distribution used in Table 5 is organised in such a way that 65% is a single class and 35% is mixed from other classes. This change slightly reduces the Non-IID effect and improves the accuracy performance of the model.

In binary classification models, similar to the IID system, it is seen that Non-IID does not have a great effect. Accuracy values remained at the level of 0.996 in all models and did not change significantly. Since the Non-IID effect creates more problems in multi-class learning, the accuracy in the BC system remained almost the same as the IID system. However, the point to be considered here is the change in Elapsed Time. Especially in models with 5 and 10 users, the training time decreased significantly. For example, while the training time was 12.31 seconds for 10 users in Table 4, it decreased to 3.43 seconds in

In the multi-classification task, it is seen that the Non-IID effect decreases compared to Table 4. Especially the accuracy values have increased compared to Table 4. While the accuracy value of Model 7 (5 users) was 0.802 in Table 4, it increased to 0.836 in Table 5. While the accuracy value of Model 8 (10 users) was 0.849, it was 0.857 in Table 5. The reason for this is that the class imbalance is less than the first Non-IID scenario. In the first scenario, since 80% of the users' data came from a single class, the model had difficulty in learning some classes. However, in the second scenario, since the dominant class ratio was reduced to 65%, the model was more exposed to other classes and its generalisation capacity increased. It enabled the model to achieve high accuracy in multi-class.

In Table 6, the third Non-IID scenario gave the best results compared to the other two scenarios. The users' data was mixed with 50% dominant class and 50% other classes. This distribution provides a more balanced structure than the first two Non-IID scenarios and is almost identical to the IID system, especially in terms of the binary classification task. . This is because binary classification involves only two classes and the data is split 50%-50% on each client, so that the federated learning works exactly as in the IID system.

This experiment proves that the impact of Non-IID is highly dependent on the structure of the data distribution. If the data is distributed completely unbalanced, as in the first scenario, the model performance is severely degraded. However, when a more balanced distribution is provided, it becomes much easier for the central model to compensate for the Non-IID effect. As a result, this third Non-IID scenario yielded exactly the same results as the IID system, especially in the BC task, and very close accuracy values to the IID system in the MC task. This shows that Non-IID does not always have a negative effect and that the model can continue to learn stably when the data distribution is well adjusted. Especially as the number of users increased, the generalisation capacity of the central model increased and the negative effects of Non-IID were further reduced. This study shows that the creation of a balanced data structure across classes is critical to reduce the impact of Non-IID data distribution and that client diversity in the federated learning process can improve model accuracy.

Table 5. Evaluation of Non-IID models trained on the CIC-DDoS2019 dataset with 65% of the classes in the data being the same and 35% randomly distributed as other classes.

No	Model	Architecture	Users	Epochs/Rounds	Elapsed Time(s)	Accuracy	Precision	Recall	F1-score
1	Non-Fed + BC		1	10	84.35	0.997	0.999	0.997	0.998
2	Fed + BC		5	3	4.73	0.996	0.999	0.996	0.998
3	Fed + BC	BA-DNN	10	3	3.43	0.996	0.999	0.996	0.998
4	Fed + BC		50	5	3.07	0.996	0.999	0.996	0.998
5	Fed + BC		100	7	3.69	0.996	0.999	0.996	0.997
6	Non-Fed + MC		1	10	82.03	0.932	0.954	0.932	0.916
7	Fed + MC		5	18	23.39	0.836	0.936	0.836	0.839
8	Fed + MC	MA-DNN	10	16	13.15	0.857	0.935	0.857	0.851
9	Fed + MC		50	14	6.61	0.888	0.888	0.888	0.875
10	Fed + MC		100	12	5.16	0.902	0.908	0.902	0.881

Table 6. Evaluation of Non-IID models trained on the CIC-DDoS2019 dataset with 50% of the classes in the data being the same and 50% randomly distributed as other classes.

No	Model	Architecture	Users	Epochs/Rounds	Elapsed Time(s)	Accuracy	Precision	Recall	F1-score
1	Non-Fed + BC		1	10	84.35	0.997	0.999	0.997	0.998
2	Fed + BC		5	5	11.39	0.997	0.999	0.996	0.998
3	Fed + BC	BA-DNN	10	4	5.78	0.996	0.999	0.996	0.998
4	Fed + BC		50	4	2.43	0.996	0.999	0.996	0.997
5	Fed + BC		100	5	2.09	0.995	0.998	0.996	0.997
6	Non-Fed + MC		1	10	82.03	0.932	0.954	0.932	0.916
7	Fed + MC		5	13	22.77	0.908	0.946	0.908	0.882
8	Fed + MC	MA-DNN	10	22	22.67	0.912	0.945	0.912	0.898
9	Fed + MC		50	13	6.78	0.910	0.927	0.910	0.896
10	Fed + MC		100	15	5.22	0.921	0.938	0.919	0.901

5 Results and discussion

In this study, we developed a DDoS attack detection based on FL architecture. Our findings indicate that the FL strategy maintains high accuracy levels comparable to traditional methods, while drastically reducing training time by 82% to 97%, depending on the number of clients involved. This reduction is crucial for real-time applications that demand high speed and efficiency.

The primary advantage of the DDoS_FL model is its ability to train directly on client devices, eliminating the need to centralize sensitive data. In addition to enhancing data privacy [65], this approach aligns with contemporary legal requirements focused on data protection, such as GDPR and KVKK. By utilizing local computations instead of centralized data aggregation, we minimize vulnerability to data breaches.
The model's resilience to various DDoS attack scenarios was thoroughly validated using the CIC-DDoS2019 dataset. Our FL model demonstrated a slight reduction in training times as the number of clients increased, highlighting the scalability of our method. Scalability is essential in environments with a large number of IoT devices or network edges, as seen in modern network topologies.
Compared to previous studies, DDoS_FL not only improves operational efficiency but also offers a practical foundation for deployment in a wide range of settings, from small-scale networks to large distributed systems. The reduction in training time without compromising accuracy illustrates the practical applicability of FL in real-world scenarios.
Additionally, our evaluation under both IID and Non-IID data distributions demonstrates the adaptability of the proposed model to different real-world data conditions. While IID data distribution provides more balanced learning across clients, Non-IID scenarios introduce variations in local datasets, reflecting realistic network environments. Our model maintains high detection accuracy under both conditions, proving its robustness against data heterogeneity. These results further emphasize the effectiveness of our FL-based approach in securing distributed systems against DDoS attacks, regardless of data distribution characteristics.
Our results emphasize FL's efficiency in increasing the security features of DDoS detection systems while addressing privacy concerns. The balance between performance and privacy presents a new paradigm in cybersecurity, particularly in DDoS protection. Further research and development in this field will help refine these models to better address the evolving cyber threat landscape.

6 Conclusion and future works

The aim of this study is to propose a FL-based DL architecture that detects anomalous traffic and classifies network traffic. DNN offers a significant advantage in the analysis of network traffic by combining extraction and classification capabilities thanks to its multi-layered structure. In order to train DNN models within the scope of the study, the current dataset named CIC-DDoS2019 was preferred. This dataset has been prepared in two different types, BC and MC, for DDoS attack detection and classification. The BC dataset was developed in order to identify network traffic attacks. "Bening" is labeled "0" and others are labeled "1". The MC dataset was proposed for the classification of attacks. Seventeen different attack types are labeled as 1-17, with normal traffic as "Bening" 0.

Models were trained using classical and FL methods and their performances were compared. FL-based models were created with different user numbers and different epoch numbers. Comparisons were made for both prepared datasets. Additionally, additional experiments were performed for IID and Non-IID data distributions.

This study compares traditional learning methods with federated learning (FL) techniques on various datasets and examines the impact of Non-IID data distribution on FL. Experiments compared the accuracy of traditional learning and FL methods on BC and MC datasets, revealing that FL achieved significantly shorter training times but similar accuracy to traditional learning. The effect of Non-IID data distribution was more pronounced on the MC dataset, while it was less impactful on the BC dataset.

The accuracy of FL models increased with the number of users, and the impact of Non-IID data was more significant with fewer users. Balanced data distributions improved FL model accuracy and reduced training times. Specifically, with 100 users, the FL model's accuracy stabilized, and the effect of Non-IID data decreased substantially. The results suggest that more balanced class distributions can improve model performance and mitigate the effects of Non-IID data.

By using the FL method, the need to transfer all the data from the clients to the server has been eliminated. Thus, the internet traffic between the edges and the server was significantly reduced and data privacy was ensured. The developed models are both based on FL, which is the latest technology and gives almost precise results in a shorter time compared to the classical method, IDS and SDN. It shows that it can be used as a reliable tool in cyber security areas such as IDS and SDN-based systems, and it is thought that using these models in systems where network traffic is managed will contribute to the early detection and prevention of DDoS attacks.

This study addresses the fundamental challenges of data privacy and centralized data collection. A federated learning (FL)-based approach is proposed to detect DDoS attacks.

However, there are several areas planned for future work to further enhance the applicability and robustness of the proposed model. In real-world scenarios, data distribution among clients is continuously changing. Therefore, integrating online learning techniques [66, 67] to dynamically update the model and adopting adaptive federated learning approaches (e.g., personalized FL or meta-learning-based FL) can enhance the model's adaptability to different network environments. Additionally, due to the evolving nature of DDoS attacks, new attack types may emerge over time. To ensure rapid adaptation to these novel threats, federated transfer learning techniques [68] can be leveraged, enabling previously trained models to adjust to new attack patterns efficiently.

7 Author contribution statements

Within the scope of this study, Büşra Büyüktanır, Zeki Çıplak, Abdullah Emir Çil, Özlem Yakar, Mahamoud Brahim Adoum, and Kazım Yıldız contributed equally to the study by participating in the development of the idea, literature review, evaluation of results, as well as the writing and proofreading processes.

8 Ethics committee approval and conflict of interest statement

"There is no need to obtain permission from the ethics committee for the article prepared". There is no conflict of interest with any person/institution in the article prepared."

9 References

[1] Ganal S, Küçüksille E, Yalçınkaya MA. "PhisherHunter: Module design for automatic detection of phishing websites and preventing user abuse". *Pamukkale University Journal of Engineering Sciences*, 29(5), 468-480, 2023.

[2] Peng T, Leckie C, Ramamohanarao K. "Survey of network-based defense mechanisms countering the DoS and DDoS problems". *ACM Computing Surveys*, 39(1), 3-es, 2007.

[3] Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA. "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy". *IEEE 2019 International Carnahan Conference on Security Technology,* Chennai, India, 1-3 October 2019.

[4] Altuncu MA. Developing an intrusion detection and prevention system using machine learning and deep learning methods. PhD Thesis, Kocaeli University, Kocaeli, Türkiye, 2021.

[5] McMahan B, Moore E, Ramage D, Hampson S, Arcas BAy. "Communication-efficient learning of deep networks from decentralized data". *Artificial Intelligence and Statistics*, Fort Lauderdale, USA, 20-22 April 2017.

[6] Sun Y, Esaki H, Ochiai H. "Adaptive intrusion detection in the networking of large-scale LANs with segmented federated learning". *IEEE Open Journal of the Communications Society*, 2, 102-112, 2020.

[7] Anli YA, Ciplak Z, Sakaliuzun M, Izgu SZ, Yildiz K. "DDoS detection in electric vehicle charging stations: A deep learning perspective via CICEV2023 dataset". *Internet of Things*, 28, 101343, 2024.

[8] Cil AE, Yildiz K, Buldu A. "Detection of DDoS attacks with feed forward based deep neural network model". *Expert Systems with Applications*, 169, 114520, 2021.

[9] McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. "Communication-efficient learning of deep networks from decentralized data". *20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, USA, 20-22 April 2017.

[10] Dolaat KMM, Erbad A, Ibrar M. "Enhancing global model accuracy: Federated learning for imbalanced medical image datasets". *International Symposium on Networks, Computers and Communications,* Paris, France, 23-26 October 2023.

[11] Lu Z, Pan H, Dai Y, Si X, Zhang Y. "Federated learning with non-IID data: A survey". *IEEE Internet of Things Journal*, 11(11), 19188-19209, 2024.

[12] Li T, Sahu AK, Talwalkar A, Smith V. "Federated learning: Challenges, methods, and future directions". *IEEE Signal Processing Magazine*, 37(3), 50-60, 2020.

[13] Yang Q, Liu Y, Cheng Y, Kang Y, Chen T, Yu H. "Federated learning: Synthesis lectures on artificial intelligence and machine learning". *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3), 1-207, 2019.

[14] OpenMined. "Pysyft". https://github.com/OpenMined (11.04.2023).

[15] Lim WYB, Luong NC, Hoang DT, Jiao Y, Liang YC, Yang Q, Niyato D, Kim DI, Miao C. "Federated learning in mobile edge networks: A comprehensive survey". *IEEE Communications Surveys and Tutorials*, 22(3), 2031-2063, 2020.

[16] Zhu X, Wang J, Hong Z, Xia T, Xiao J. "Federated learning of unsegmented Chinese text recognition model". *IEEE 31st International Conference on Tools with Artificial Intelligence,* Portland, OR, USA, 4-6 November 2019.

[17] Jabłecki P, Ślazyk F, Malawski M. "Federated learning in the cloud for analysis of medical images – experience with open source frameworks". *MICCAI Workshop on Distributed and Collaborative Learning*, Lima, Peru, 4-8 October 2021.

[18] Yazdinejad A, Parizi RM, Dehghantanha A, Karimipour H. "Federated learning for drone authentication". *Ad Hoc Networks*, 120, 102574, 2021.

[19] Clayton J, Kontokosta CE, Hong T, Corgnati SP, D'Oca S, et al. "The ASHRAE Great Energy Predictor III competition: Overview and results". *Science and Technology for the Built Environment*, 26(10), 1427-1447, 2020.

[20] Dasari SV, Mittal K, Bapat J, Das D. "Privacy enhanced energy prediction in smart building using federated learning". *IEEE International IoT, Electronics and Mechatronics Conference*, Toronto, Canada, 21-24 April 2021.

[21] Mosteiro P, Rijcken E, Zervanou K, Kaymak U, Scheepers F, Spruit M. "Making sense of violence risk predictions using clinical notes". *Health Information Science: 9th International Conference,* Amsterdam, USA, 20-23 October 2020.

[22] Borger T, Mosteiro P, Kaya H, Rijcken E, Salah AA, Scheepers F, Spruit M. "Federated learning for violence incident prediction in a simulated cross-institutional psychiatric setting". *Expert Systems with Applications*, 199, 116720, 2022.

[23] Zhao R, Yin Y, Shi Y, Xue Z. "Intelligent intrusion detection based on federated learning aided long short-term memory". *Physical Communication*, 42, 101157, 2020.

[24] Singh S, Bhardwaj S, Pandey H, Beniwal G. "Anomaly detection using federated learning". *Proceedings of International Conference on Artificial Intelligence and Applications*, Singapore, 9-10 July 2020.

[25] Tang Z, Hu H, Xu C. "A federated learning method for network intrusion detection". *Concurrency and Computation: Practice and Experience*, 34(10), e6812, 2022.

[26] Hsu RH, Wang YC, Fan CI, Sun B, Ban T, Takahashi T, Wu TW, Kao SW. "A privacy-preserving federated learning system for Android malware detection based on edge computing". *15th Asia Joint Conference on Information Security*, Taipei, Taiwan, 20-21 August 2020.

[27] Sharafaldin I, Lashkari AH, Ghorbani AA. "Toward generating a new intrusion detection dataset and intrusion traffic characterization". *4th International Conference on Information Systems Security and Privacy*, Funchal, Madeira, Portugal, 22-24 January 2018.

[28] Vaccari I, Chiola G, Aiello M, Mongelli M, Cambiaso E. "MQTTset, a new dataset for machine learning techniques on MQTT". *Sensors*, 20(22), 6578, 2020.

[29] Friha O, Ferrag MA, Shu L, Maglaras L, Choo KR, Nafaa M. "FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things". *Journal of Parallel and Distributed Computing*, 165, 17-31, 2022.

[30] Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA. "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy". *International Carnahan Conference on Security Technology*, Chennai, India, 1-3 October 2019.

[31] Doriguzzi-Corin R, Siracusa D. "FLAD: Adaptive federated learning for DDoS attack detection". *Computers and Security*, 137, 103597, 2024.

[32] Canadian Institute for Cybersecurity. "CICDDoS". https://www.unb.ca/cic/datasets/ddos-2019.html (21.12.2024).

[33] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. "A detailed analysis of the KDD CUP 99 data set". *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, Canada, 8-10 July 2009.

[34] Canadian Institute for Cybersecurity. "CICIDS". https://www.unb.ca/cic/datasets/ids-2017.html (21.12.2024).

[35] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. "SMOTE: Synthetic minority over-sampling technique". *Journal of Artificial Intelligence Research*, 16, 321-357, 2002.

[36] McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. "Communication-efficient learning of deep networks from decentralized data". *20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, USA, 20-22 April 2017.

[37] Zhang J, Yu P, Qi L, Liu S, Zhang H. "FLDDoS: DDoS attack detection model based on federated learning". *IEEE 20th International Conference on Trust, Security, and Privacy in Computing and Communications*, Shenyang, China, 20-22 October 2021.

[38] Moustafa N, Slay J. "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)". *Military Communications and Information Systems Conference*, Canberra, Australia, 10-12 November 2015.

[39] Li J, Lyu L, Liu X, Zhang X, Lyu X. "FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT". *IEEE Transactions on Industrial Informatics*, 18(6), 4059-4068, 2021.

[40] CAIDA. "Datasets–DDoS Attack (2007)". https://data.caida.org/datasets/security/ddos-20070804 (01.09.2022).

[41] Ali MN, Imran M, Uddin MS, Kim BS. "Low rate DDoS detection using weighted federated learning in SDN control plane in IoT network". *Applied Sciences*, 13(3), 1431, 2023.

[42] Jazi HH, Gonzalez H, Stakhanova N, Ghorbani AA. "Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling". *Computer Networks*, 121, 25-36, 2017.

[43] Elsayed MS, Le-Khac NA, Jurcut AD. "InSDN: A novel SDN intrusion dataset". *IEEE Access*, 8, 165263-165284, 2020.

[44] Fotse YSN, Tchendji VK, Velempini M. "Federated learning based DDoS attacks detection in large scale software-defined network". *IEEE Transactions on Computers*, 74(1), 101-115, 2025.

[45] Zainudin A, Akter R, Kim DS, Lee JM. "FedDDoS: An efficient federated learning-based DDoS attacks classification in SDN-enabled IIoT networks". *13th International Conference on Information and Communication Technology Convergence*, Jeju Island, Republic of Korea, 19-21 October 2022.

[46] Matsuda K, Sasaki Y, Xiao C, Onizuka M. "FedMe: Federated learning via model exchange". *SIAM International Conference on Data Mining*, Alexandria, USA, 28-30 April 2022.

[47] Lee YC, Chien WC, Chang YC. "FedDB: A federated learning approach using DBSCAN for DDoS attack detection". *Applied Sciences*, 14(22), 9995, 2024.

[48] Ali PJM, Faraj RH, Koya E. "Data normalization and standardization: A technical report". Machine Learning Technical Reports, Koya University, Koya, Erbil, Iraq, 1(1), 1-6, 2014.

[49] Büyüktanır B, Yıldız K, Ülkü EE, Büyüktanır T. "du-CBA: Veriden habersiz ve artırımlı sınıflandırmaya dayalı birliktelik kuralları çıkarma mimarisi". *Journal of the Faculty of Engineering and Architecture of Gazi University*, 38(3), 1919-1930, 2023.

[50] Demir A, Kulaksız A Y, Büyüktanır B, Büyüktanır B, Yıldız K. "Model Training and Real World Analysis Using Health Data with Federated Learning". *The International Open Source Conference (UAKK 2024)*, June 2025.
[51] Büyüktanır B. "YENİ NESİL MODEL EĞİTİM YAKLAŞIMI: FEDERE ÖĞRENME (NEW GENERATION MODEL EDUCATION APPROACH: FEDERATED LEARNING)". *UBAK DERLEME: International Scientific Compilation Research Congress*, No. 1, pp. 208–212, 29 February 2024.
[52] Google Brain Team. "TensorFlow". https://www.tensorflow.org/ (16.06.2023).
[53] Chollet F. "Keras". https://keras.io/ (16.06.2023).
[54] McKinney W. "Pandas". https://pandas.pydata.org/ (16.06.2023).
[55] INRIA. "Scikit-learn". https://scikit-learn.org/stable/ (16.06.2023).
[56] Schmidt-Hieber J. "Nonparametric regression using deep neural networks with ReLU activation function". *Annals of Statistics*, 48(4), 1875-1897, 2020.
[57] Gal Y, Ghahramani Z. "A theoretically grounded application of dropout in recurrent neural networks". *30th International Conference on Neural Information Processing Systems,* Barcelona, Spain, 5-10 December 2016.
[58] Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S. "Deep learning approach for intelligent intrusion detection system". *IEEE Access*, 7, 41525-41550, 2019.
[59] Bottou L. "Large-scale machine learning with stochastic gradient descent". *19th International Conference on Computational Statistics,* Paris, France, 22-27 August 2010.
[60] Campos EM, Saura PF, González-Vidal A, Hernández-Ramos JL, Bernabe JB, Baldini G, Skarmeta A. "Evaluating federated learning for intrusion detection in Internet of Things: Review and challenges". *Computer Networks*, 203, 108661, 2022.
[61] Büyüktanır B, Altınkaya Ş, Karataş Baydoğmuş G, Yıldız K. "Federated learning in intrusion detection: Advancements, applications, and future directions". *Cluster Computing*, 28(7), 1-25, 2025.
[62] Yang C, Wang Q, Xu M, Chen Z, Bian K, Liu Y, Liu X. "Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data". *Proceedings of the Web Conference*, Ljubljana, Slovenia, 19-23 April 2021.
[63] Abay A, Zhou Y, Baracaldo N, Rajamoni S, Chuba E, Ludwig H. "Mitigating bias in federated learning". *arXiv preprint*, arXiv:2012.02447, 2020.
[64] Huang T, Lin W, Shen L, Li K, Zomaya AY. "Stochastic client selection for federated learning with volatile clients". *IEEE Internet of Things Journal*, 9(20), 20055-20070, 2022.
[65] Ozturk O, Buyuktanir B, Baydogmus G K, Yildiz K. "Differential Privacy in Federated Learning: Mitigating Inference Attacks with Randomized Response". *arXiv preprint arXiv:2509.13987*, 2025.
[66] Hoi SC, Sahoo D, Lu J, Zhao P. "Online learning: A comprehensive survey". *Neurocomputing*, 459, 249-289, 2021.
[67] Mitra A, Hassani H, Pappas GJ. "Online federated learning". *60th IEEE Conference on Decision and Control*, Austin, USA, 13-15 December 2021.
[68] Liu Y, Kang Y, Xing C, Chen T, Yang Q. "A secure federated transfer learning framework". *IEEE Intelligent Systems*, 35(4), 70-82, 2020.