Enhancing Network Intrusion Detection using Federated Learning

MSc Research Project
Cybersecurity

Jawad Altaf
Student ID: 23203803

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Jawad Altaf		
Student ID:	X23203803		
Programme:	Masters in Cyber Security	**Year:**	2023-2024
Module:	Research in Computing		
Supervisor:	Prof.Vikas Sahni		
Submission Due Date:	12th August, 2024		
Project Title:	Enhancing Network Intrusion Detection using Federated Learning.		
Word Count:	**7035**	**Page Count: 22**	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Jawad Altaf

Date: 11th August, 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Network Intrusion Detection using Federated Learning

Jawad Altaf
23203803

ABSTRACT

Cyber-attacks are increasing at an alarming rate as IOT, industrial control systems and other devices connected to the internet are exposed to malware, DDOS, DOS and malicious activities. Past research work is on centralized intrusion detection, which introduces issues like single point of failure, data privacy and scalability. Federated learning (FL) provides solutions to the issues concerning privacy and scalability by working and learning locally on distributed devices.

This research introduces a novel approach for enhanced intrusion detection using federated learning with Gated recurrent neural network integrated into flower federated framework, while comparing it with centralized machine learning technique using GRU (Gated Recurrent Unit). This research demonstrated centralized learning showed high accuracy of 97%, however, federated learning models preserved the privacy of data with moderated performance measures in terms of multiple clients. The research indicates that FL could be a useful approach for creating efficient and private NIDS solutions.

Keywords: Federated Learning, NIDS (Network intrusion Detection System), GRU, Flower

1. Introduction

Machine learning algorithms aggregate all training data centrally to train the model (Shastri, 2024). Centralizing client data for training has the potential to provide significant protection, but it has negative impact on privacy. As threats increase in the cyber world more attention needs to be paid to improve network intrusion detection systems (NIDS). Therefore, this research aims at advancing the detection of NIDS through federated learning. Using federated learning approach as an alternative to centralized learning, where the data is distributed between devices while maintaining the privacy of the information, provides an opportunity for real-time intrusion detection.

1.1 Background

Today, new types of cyber threats continue evolving and can pose a threat to the network infrastructures of the entire world. In response to such threats, it is imperative to create high-performance, accurate, and privacy-preserving intrusion detection systems (IDS). NIDS connected in centralized data centers are typical for traditional approaches, which is why latency, privacy, and scalability challenges emerge. As the basis of this research, the TII-SSRC-23 [1]dataset is employed to examine how FL helps in intrusion detection, thereby solving the issues of data security and scalability. The model keeps data localized and processes it in a distributed manner; therefore, FL provides a viable solution to improve NIDS. This work aims to show how FL can be integrated and trained in enhancing the NIDS accuracy and efficiency, thus assisting in the development of privacy-preserving and scalable solutions for CTDS.

1.2 Motivation

The significance of this study resides in making new breakthroughs in the NIDS field by utilizing federated learning (FL). With more and more threats being developed and executed in the cyber world, there is a need for NIDS that are effective, fast, and that do not breach user privacy or suffer from the scalabilities. As will be described later, conventional centralized methods do not adequately meet these requirements. To improve the real-time detection of over constrained scenarios and at the same time address the issues of privacy caused by data centralization, this study seeks to use the TII-SSRC-23 dataset and FL. They provide insights to design and build stronger and non-proportional NIDS for enhancing the global network for cyber security.

1.3 Federated Learning and Its Advantages

FL is a networked version of machine learning where a model is trained by several devices/servers but the data remains decentralized (Shastri, 2023). This method improves the level of privacy as well as security by detail because raw data is never moved out of the device or server where it is created. Instead of distributing all the data to a central server, FL sends the model updates (gradients) from each client to the server and the server updates the global model. This then is brought back to the original participants where the global model is 'fitted' using local data to obtain the final version of the model. This process of applying the model carries on several folds until the model's performance meets the desired norm (Bag, 2024). In terms of the benefits of federated learning, one can state that it allows preventing the disclosure of data and keeping them safe. Due to the federated learning system's focus on data storage at a local level and the sharing of model updates only, the risk of data leaks, or violation of data protection legislation such as GDPR, is minimized. This is even more pertinent in areas of operation like healthcare, especially in patient records or in financial institutions where the data of transactions must be secured (Li *et al*., 2020).

Federated learning also tackles some of the problems to do with diversity of the data and the representation of that data. The traditional centralized machine learning models are flawed in terms

[1] Dania Herzalla, Willian T. Lunardi, and Martin Andreoni. (2023). TII-SSRC-23 Dataset [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DS/3631110

of bias because the data fed into the model biases are often homogeneous (Shastri, 2023). However, FL is less affected by the data collection infrastructure because it gathers data from multiple devices and contexts, thus creating more accurate and transferable models. For example, in a medical research context, FL can integrate information regarding different groups of patients across different locations while maintaining the patients' privacy, thereby improving the models used in medicine that is generalizable (Bag, 2024). FL is more adaptable to the conditions of low connectivity to other devices and low processing power. Since the data is stored on the communication devices, constant and high-volume data transmission to a central server is not required. Due to this, FL is a perfect solution for edge computing situations, where smartphones, IoT objects, and sensors can together train models while working in isolation and not necessarily being connected to the internet all the time. It also entails less computation on central servers to increase scalability since the computation is distributed across many devices (Jin *et al*., 2023).

It is a problem that data and computational resources are distributed non-uniformly among the devices and this causes imbalances in the training phase. It might be so for a simple reason that some devices contain more data or are equipped with more powerful processors, thereby making a non-proportionate contribution to the model. Also, protecting the model updates during transmission and defending it against adversarial attacks are some of the issues of concern. Some of these difficulties can be solved by ideas like differential privacy and secure multiparty computing to make Fl systems more secure. Federated learning is an innovative approach to machine learning, which emphasizes privacy, security, and decentralization aspects. These advantages lie in the key contributions that FL brings to building models with improved generalizability and robustness in use cases that involve sensitive and distributed data, as well as edge computing architectures. Nonetheless, despite the nature of these difficulties, constant advancements in research and development initiatives make FL a progressively realistic and desirable solution for different usages (Li *et al*., 2020).

1.4 Research Question and Objectives

Research Question: How can federated learning improve the accuracy and efficiency of network intrusion detection systems while addressing data privacy.

1.4.1 Objectives

- Test the efficiency of federated learning to enhance the reliability and authenticity of NIDS.
- Evaluate the effectiveness of the federated learning approach and make comparison with centralized learning approach.

1.4.2 Limitations

There are several limitations in this study as follows:

- First, FL has inherently higher computational cost, when practiced in a distributed network. Unbalanced and more heterogeneous data collected from multiple devices can be noisy and pose a problem to the stability and reliability of the FL model.
- There are also some practical issues related to the coordination and administration of the update process across multiple independent systems, which lead to a certain delay and, therefore, to the reduction of the effectiveness of the NIDS in real time. This is important when trying to understand the outcomes and evaluate the feasibility of the solutions presented in the study.

1.5 Structure of The Report

The structure of the rest of this report is as follows: The related work on the network intrusion detection systems and federated learning are described in Section 2. Section 3 provides an account of the research methodology applied in this study, such as data gathering, preparation, and utilizing federated learning models. Sections 4 and 5 outline the design specification and experimental results and their assessment and summarizes the results in relation to the current research and points out the desirable implications. Phyton coding and associated libraries and federated learning framework were used in this section. Section 6 explains the evaluations of centralized and federated learning models. Lastly, in Section 7, the author gives an overall summary and makes suggestions for future investigations and uses of the study.

2. Related Work

In this section, this research delivers a review of the current literature on the use of machine learning and federated learning for intrusion detection. It also looks at the past developments, current issues, and the future developments of these technologies in intrusion detection. This review includes the type of learning that might be used in IDS, namely supervised learning, unsupervised learning, reinforcement learning, and even federated learning.

2.1 Scope

Selection Criteria	Journal articles, conference papers. Research published during the period between **2020** and **2024**. Literature targeted federated learning and machine learning, deep learning for NIDS.
Exclusion Criteria	Source: IEEE, Wiley, Science Direct, Mdpi. Search equations: Machine Learning, Federated Learning.
Targeted Area	Federated Learning and Machine Learning approaches for NIDS.

2.2 Literature Review

Wang, Li and Wu (2022) presented a federated transfer learning approach for intrusion detection termed, with an algorithm named FETLSVMP. This method leverages federated learning to ensure privacy while aggregating data from different organizations and utilizes transfer learning to address distribution disparities. The paper focusses on techniques such as homomorphic encryption to maintain data privacy and personalized model adaptation for each organization. However, it might come with negative transfer issues, where there is a risk of negative transfer and application of knowledge from one domain adversely impacts the performance of another domain.

Li *et al.* (2021) proposed a federated deep learning framework that incorporates three modules: a convolution neural network (CNN), a gated recurrent unit (GRU), and a multi perceptron (MLP). The output from the multi classification CCN-GRU Model is combined and input to MLP module. However, the framework convergence is not addressed in their discussion.

Alazab *et al.*'s (2023) research paper examines the use of federated learning to enhance privacy preservation for IDS. The purposed methodology includes federated averaging, differential privacy, and secure aggregation techniques to maintain data privacy and security while training a shared model across multiple clients. The experiment was conducted using the NSL-Dataset, demonstrating that federated learning achieved higher accuracy and lower loss as compared to traditional deep learning model. However, the study lacks implementation on new datasets which contain diverse attack scenarios.

In Huang *et al*. (2024) the aim of the paper is to use federated learning-based intrusion detection system for industrial internet of things. They proposed the DVACNN fed model that combines convolutional neural networks (CNNs) with attention mechanisms and variational autoencoders to enhance data privacy and detection accuracy. This methodology involves differential privacy, federated averaging, and secure aggregation techniques, used on TON_IOT and BOT_IOT datasets. While the model showed improved performance in terms of accuracy, precision and false positive rate, the paper lacks implementing federated learning setup using the Flower framework.

Rahman *et al*. (2020) discussed a privacy preserving federated learning (FL) approach for IOT intrusion detection, addressing limitations of centralized and on device learning methods. The proposed scheme involves local training and inference on devices, sharing only updated models with a central server for aggregation. This methodology utilizes the NSL-KDD dataset for evaluation, highlights its efficiency in maintaining data privacy and reducing communication overhead while achieving accuracy comparable to centralized approaches. Some issues related to

FL include device dropout, slower response time and clients with low frequent data were not discussed.

Chen *et al.* (2020) examined FEDAGRU, a federated learning approach combining gated recurrent unit and support vector machine (SVM) models for intrusion detection in wireless edge networks (WENs). FedAGRU improves detection accuracy by 8% compared to centralized algorithms and reduces communication costs by 70%. It is also highly robust to poising attacks. However, the model uses the datasets of KDD CUP 99, CICIDS 2017 which are old and do not contain DDOS attack scenarios as compared to the TII-SSRC23 dataset.

Attota *et al*. (2021) presented MV-FLD, a federated learning-based intrusion detection model for IOT network that enhances attack identification accuracy by using multi-view learning with BI-flow, UNI-flow and packet data partitions. This approach trains models locally without transferring raw data to a central server, outperforming centralized ML methods. However, the study highlights only a limited number of attacks as compared to large datasets containing high number of malicious traffic.

Shukla *et al*. (2024) published their research on FEDHNN, a federated learning-based hybrid neural network combining CNN and LSTM for real time intrusion detection in wireless sensor network. Using the NSL-KDD datasets, it attained higher accuracy of 97.68% and low loss of 0.1568 outperforming many other methods. The method is designed for binary classification; however, future studies could expand the classification into several classes.

Li *et al*. (2023) used a dynamic weighted aggregation federated learning system, DAFL, for network intrusion detection. The methodology used federated learning to enhance data privacy, utilizing a dynamic aggregation approach that filters and weights local models based on their performance. They used the CSE-CIC-IDS2018 dataset, demonstrating good detection performance in terms of metrics such as accuracy, precision, recall and an F1 score with multiple communication rounds of 3,5 and 7. The study highlighted improved communication efficiency, reducing overhead by 33-71% compared to other methods. However, the paper lacks details on implementing federated learning using advanced frameworks like PyTorch and Flower.

Qazi *et al.* (2022) presented an intelligent and efficient network intrusion detection system (NIDS) using deep learning, specifically focusing on a stacked non-symmetric deep auto encoder (S-NDAE) combined with a support vector machine. The methodology used TensorFlow for implementation and evaluates performance on the KDD CUP '99 dataset. The proposed system achieves higher accuracy of 99.65%, a precision of 99.99% and a recall of 99.85%. The study lacks exploration of real time data processing and diverse datasets beyond KDD Cup'99 as compared to other datasets.

Zhai *et al*. (2023) published their research on federated learning-based intrusion detection systems for smart grids, combining convolutional neural network and gated recurrent units (GRUs) to address privacy and security challenges. The methodology used an attention mechanism to enhance feature extraction and introduces a trust-based node selection mechanisms for improved convergence. The system utilizes the NSL-KDD dataset demonstrating accuracy of 78.79%, a recall of 64.15 5 and an F1- Score of 76.90 %. Secondly, the dataset used here is NSL-KDD which is an extension of KDD CUP'99 where the traffic was generated by simulation over a virtual computer network, it could have been replaced by real time traffic datasets for better results.

Das and Brunschwiler (2019) researched federated learning (FL) conducted on edge devices, particularly focusing on privacy preserving methods of digital health applications. The study utilized CNN, LSTM and MLP models trained on the MINST dataset using Raspberry Pi-4 devices, highlighting the performance and latency of these models under IID and non-IDD data distributions. The methodology included federated averaging (FEDAvg) for model updates, achieving up to 85% accuracy while minimizing data transfer. The paper used the pysftp method for executing the experiment.

Ahanger *et al*. (2022) explored a federated learning FL technique for attack classification in IOT networks, utilizing long short-term memory (LSTM) and gated recurrent unit (GRU) models. The methodology includes decentralized training on the edge devices to preserve data privacy, with model updates shared with a central server. The study uses the TON_IOT dataset, demonstrating enhanced statistical performance in accuracy, precision, recall, and F1- measure compared to conventional method. However, the author suggested these techniques can be further improved by using real time information concerning known and unknown vulnerabilities in IOT devices.

Hou, Liu and Zhuang (2019) put forward an intrusion detection model built on gated recurrent units (GRUs) and a salient feature selection technique. Using the NSL-KDD dataset, the study implements feature selection to decrease data dimensionality, ensuring essential information for various intrusion types is maintained. The GRU model, refined with adaptive moment estimation (Adam) and cross-entropy loss functions, demonstrated notable accuracy and computational efficiency. Performance metrics were impressive, boasting a precision rate of 99.95% and an F1 score of 91.60% for various attack types. Nonetheless, the study falls short in exploring real-time data processing and does not compare its model with other advanced frameworks such as PyTorch and Flower, which are critical for practical federated learning implementations.

2.3 Summary of the related research work

Sr#	Authors	Algorithms used	Dataset	Method	Shortcomings
1	Wang, Li and Wu (2022)	FLTRELM	NSL-KDD, KDD'99, ISCX2012	Method leverages federated learning to ensure privacy while aggregating data from different organizations and utilizes transfer learning to address distribution disparities.	Experimental results shows that algorithm solves IDS for small samples but not used for big samples
2	Li *et al.* (2021)	Convolution neural Network (CNN), a gated recurrent unit (GRU), and a multi perceptron (MLP).	CPS Dataset	Deep federated learning Scheme with Paillier based-secure communication.	F-IDS used for same domain industrial CPS. Different domain industrial CPS were not used.
3	Alazab *et al.* (2023)	Federated averaging	NSL-KDD	Methodology includes federated averaging, differential privacy, and secure aggregation techniques to maintain data privacy and security.	Different testing results for deep learning and federated learning were analysed in Epochs rounds 1,5,10,20. All the results are from the NSL-KDD dataset which is old and does not contain diverse DDOS and DOS attacks.
4	Huang *et al.* (2024)	They proposed the DVACNN fed model that combines convolutional neural networks (CNNs) with attention mechanisms and variational autoencoders to enhance data privacy and detection accuracy.	TON_IOT, BOT_IOT	Methodology involves differential privacy, federated averaging, and secure aggregation techniques.	Mechanisms to shorten the training time, and enhancement of scalability were not explored in the industrial IOT field.

5	Rahman *et al*., (2020)	Federated averaging	NSL-KDD	Proposed scheme involves local training and inference on devices, sharing only model updated with a central server for aggregation.	Some issues related to FL include: device dropout, slower response time and clients with low frequent data were not discussed.
6	Chen *et al*. (2020)	Presented (FedAGRU), federated learning-based attention gated recurrent unit for WSN networks.	Combined KDD CUP99, CICIDS 2107, WSN-DS.	FedAGRU improves detection accuracy by 8% compared to centralized algorithms and reduces communication costs by 70%. It is also highly robust to poising attacks.	Model used old datasets. More focus was done in prediction of poisoning attacks as compared to other attacks in the dataset.
7	Attota *et al*. (2021)	Federated AGRU, with GRU and SVM	MQTT	MV-FLD, a federated learning-based intrusion detection model for IOT network that enhances attack identification accuracy by using multi-view learning with BI-flow, UNI-flow and packet data partitions.	Poisoning attacks where malicious data could gradually degrade model performance, and the computational overhead for real-time IoT devices needs consideration.
8	Shukla *et al*. (2024)	CNN, LSTM using federated learning	NSL-KDD	Real-time updates using distributed training based on federated learning on the NSL-KDD dataset but the testing is done centrally.	The method is designed for binary classification; however, future studies could expand the classification into several classes.
9	Li *et al*. (2023)	Dynamic weighted aggregation federated learning system, DAFL, for network intrusion detection.	CSE-CIC-IDS2018	Methodology used federated learning to enhance data privacy, utilizing a dynamic aggregation approach that filters and weights local models based on their performance	Paper lacks details on implementing federated learning using advanced frameworks like PyTorch and Flower.
10	Ahanger *et al*. (2022)	GRU model with federated learning	TON_IOT	Method employs the NSL-KDD dataset to compare the efficacy of FL models against centralized and on-device learning methodologies. Model updates were performed using techniques such as neural network architecture with federated averaging	Techniques can be further improved by using real time information concerning known and unknown vulnerabilities in IOT devices.

3. Research Methodology

3.1.0 Introduction

The approach that is used in this study aims at solving these problems through application of FL to improve the performance of NIDS. This chapter describes the methodological approach used in this study's context with emphasis on the evaluation of FL when using the TII-SSRC-23 dataset. The approach covers data preprocessing, model designing, the method of federated learning, and evaluation criteria to give an all-embracing analysis of NIDS enhancement.

The objective of the study can be explained in the light of several main goals that the work is based on. First, the goals of the study are to investigate the effectiveness of federated learning in increasing NIDS reliability and authenticity using the TII-SSRC-23 database. Second, it aims at comparing the performance difference between federated learning and centralized learning. The methodological approach covers data preprocessing so as handling missing values, labeling the data, data under sampling like NearMiss.

The implementation phase entails creating a model that would be interpreting network traffic data details; this model is a gated recurrent unit (GRU) model. Based on the federated learning setup, the model is trained and tested in Flowers, the federated learning library for Python. It helps ensure that data is never centralized as it is in many other distributed learning approaches while at the same time allowing for the joint training of models across the nodes. The evaluation parameters are, first and foremost, accuracy, followed by a confusion matrix and a classification report.

Custom Methodology

In this methodology a step-by-step process is outlined how a data set is prepared and preprocessed to create a federated learning model and assess it.

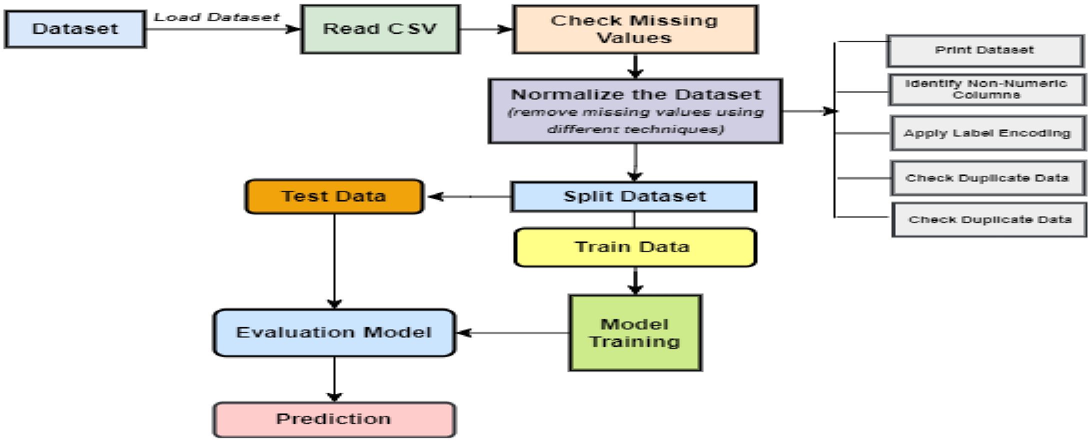


Figure 1: Methodology Flow

3.2.1 Data Collection

The first phase deals with the preparation of data, which includes gaining access to the dataset. Data has been pulled from Kaggle (Herzalla, 2023) based cyber-attacks dataset and have been used for preparing this report and enhancing the performance of NIDS system.

Dataset Description:

- The TII-SSRC-23 dataset is used for a wide selection of network traffic patterns. It was carefully designed for Intrusion Detection Systems (IDS) development and research.
- The csv file with the final dataset is 5.02 GB.
- Total number of Columns in Dataset: 86
- Traffic Count in Dataset:

Sr#	Traffic Type	Count
1	DoS	7,490,929
2	Information Gathering	1,038,363
3	Mirai	91,002
4	Brute force	35,172
5	Video	870
6	Text	209
7	Audio	190
8	Background	32
Number of Attack Instances:		85,655,466
Number of Non-Attack Instances:		1301

Table 1 Traffic Count Classification

3.2.2 Data Pre-Processing

In the data pre-processing stage, this step was to assess and possibly deal with missing values. This was important since the absence of data could result in negative impacts on the blueprint of the model.

- No missing values were found in the dataset.

- Categorical features were distinguished and transformed:

Flow ID	Src IP	Dst IP	Timestamp	Label	Traffic Type	Traffic Subtype

Table 2 Categorical Features in Dataset

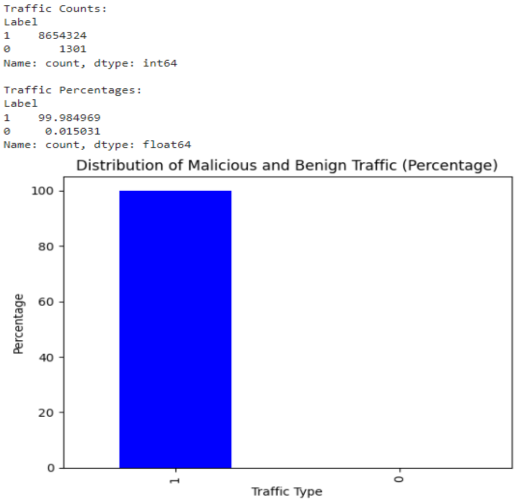
- In the case of non-numeric columns, the type of feature transformation that took place was label encoding to convert them in numeric form for machine learning algorithm.
- The next step of data pre-processing includes deletion of records which were duplicate to avoid skewing probabilities during computation.
- Following duplicate values were found in the dataset:

Duplicate Rows	1142

Table 3 Number of Duplicate Rows in Dataset

- Dataset Malicious and Benign Traffic count and Percentage:

Label	Percentage	Class
1	99.98%	Malicious samples
0	0.02%	Benign samples



```
Traffic Counts:
Label
1    8654324
0       1301
Name: count, dtype: int64

Traffic Percentages:
Label
1    99.984969
0     0.015031
Name: count, dtype: float64
```

Table 4 Percentage of Label Traffic in Dataset

- Balancing of Dataset was performed using undersampling Technique (Imbalanced learn, 2024)

3.2.3 Data Training

In the data training process 80% data used, the trained dataset prepared in the previous step was input into the model. The model was fitted using required techniques to understand the relation between the variable within training data set and to build the requisite features for prediction. This phase was to develop a model that could give proper predictions given the patterns identified from the training set.

3.2.4 Classification Algorithms

In the training process, the following algorithms were used.

- Algorithms: Gated Recurrent Unit (GRU)
- Federated learning Framework: Flower

3.2.5 Model Evaluation

- The evaluation of the model entailed using the testing data set in order to determine the performance of the trained model while making the predictions. In order to quantify the success made by the model, the essential evaluation metrics including
- Accuracy: $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$.
- Precision: $P = \frac{TP}{TP+FP}$
- Recall:$= \frac{TP}{TP+TN}$
- F1-score: $\boldsymbol{F1} = \frac{\boldsymbol{2*Precision*Recall}}{\boldsymbol{Precision+Recall}}$
- Loss function in centralized Model.
- Number of Hidden dimensions in GRU Model: 128
- Number of Epochs in Centralized Model: 50, 100, 150
- Number of Epochs in Federated Model:50, 100, 150.
- Confusion matrices and classification report were utilized.

The last task was to show the results on new data with the help of the established model, thereby providing a full cycle from raw data to the model's prediction, checking its consistency and relevance for real-world use.

CHAPTER 4: Design Specification

The applied technique was data preprocessing and federated learning; thus, modifying the data preprocessing and enhancing the deep learning algorithms in analyzing the network intrusion data was conducted. The study employed data preprocessing, feature selection, under sampling, and neural network as the main methodologies of the investigation. First, data cleaning involved dealing with the missing data, encoding of the data, and dealing with the duplicate data using pandas. To deal with the imbalanced class distribution, NearMiss under sampling was used. For feature selection, SelectKBest was employed to reduce the overall number of features to be used to the most important ones. The main of the technique was to train a novel classifier through creating and using a type of neural network known as Gated Recurrent Unit (GRU) neural network which additionally included the dropout regularization method to present network intrusions. In this study, the use of FL based on Flower was considered for the training of the GRU model and centralized model, which enables decentralization of training among clients. The results of the data distributions and the models were explained and graphically demonstrated by utilizing techniques like PCA and Matplotlib to visualize plots, confusion matrices and T-SNE to measure and compare predictions and accuracy with regards to the results. The current approach entailed the integration of preprocessing, sophisticated methods, and federated learning for accurate network intrusion detection.

Steps Involved in the Model

4.1 Data Preparation

The dataset collected from Kaggle (Herzalla, 2023). Data preprocessing included data loosening, missing data, and duplicates, encoding of nominal variables, and NearMiss undersampling techniques. The features that were chosen were done using the SelectKBest function, followed by Standard Scaler. The cleaned dataset was further divided into training and testing dataset for training of the GRU neural network and for assessment of federated learning.

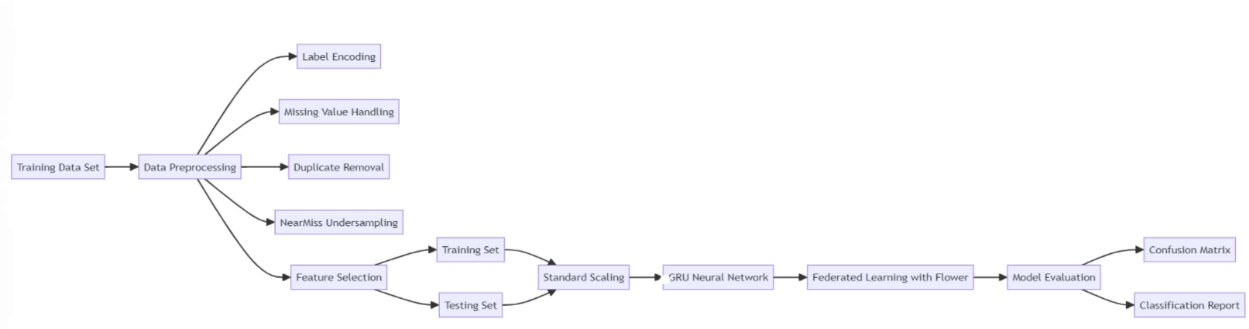


Figure 2 : Data Preparation Workflow

4.2 Data Classification

- The prepared dataset was fed to the GRU model for the classification of network traffic into benign and malicious.
- After that, the dataset passed through a training and testing stage using federated learning with Flower to encourage decentralized training of clients ranging from **2 to 5.**
- The performance of the model was assessed using metrics including accuracy, precision, recall and F1-score.

4.4 Algorithms Chosen for This Research

The models used for the research are the centralized model and federated learning with Flower.

4.4.1 Centralized Model

- Centralized system is a system where all the processing as well as decision making is done at a central point such as server. The model collects information from multiple sources and forwards it after passing through certain centralized components of analysis.

4.4.2 Federated Learning with Flower

Federated learning is a subtype of collaborative learning that enables submission of model updates from various clients while keeping their raw data private.

- Flower is a tool that helps in the implementation of FL as it handles the distribution and collection of model updates between clients.
- Every client uses its local set of data to train the parameters of a GRU neural network and then sends the model parameters to a master established server.
- These updates are then jointly accumulated on the server to come up with a global model that is then broadcast to the clients. This process improves the privacy of the data, as data remains in local domains, while still reaping from training, which acts as a form of learning.

5 Implementation

This section of the report includes a description of the process conducted to reach the research result, with references to the specific codes and the configuration notes.

5.1 Software and Hardware Used

The integrated development environment software adopted for this project is Google Collaboratory, commonly called Collab.

5.2 Dataset Used for The Analysis

The dataset used for this research has been used from Kaggle (Herzalla, 2023). The binary datasets were in the CSV format, which was also compatible with Collab.

5.3 Data Pre-Processing

The data pre-processing steps in the code are as follows:

- **Read the CSV file:** Read in the dataset into a panda data frame.
- **Check for Missing Values:** Point out any gaps in the data as far as is possible in the format of the data.
- **Label Encoding:** The non-numeric columns should be converted to numerical via applying the label encoder.
- **Check for Duplicates:** It is necessary to eliminate the repeating rows from the dataset.
- **Train-Test Split:** Select data split into a training (80%) and a test data (20%).
- **Feature Scaling:** All the features should be scaled down to be of similar range, using the StandardScaler.
- **Feature Selection:** Pass the features to SelectKBest with ANOVA F-test, then choose the first 20 features on this list.
- **Undersampling:** To do that NearMiss undersampling should be used to balance it properly.
- **Principal Component Analysis (PCA):** Lower the dimensionality to visualize.

5.4 Methodology used for Algorithms

This section is made up of the running of two models to determine performance metrices.

- **Centralized Model:** Centralized model is incorporated using a single-layer GRU (gated recurrent unit) neural network. In this model, the entire data set is run on a single computer and the given data set is used for training and testing of the model for prediction of network intrusion instances. **In our scenario only one GRU layer was used. 50, 100, 150 epochs were used to test the evaluation metrics results.**
- **Federated Learning with Flower**: This model uses federated learning with the help of the Flower framework. The training of the model is done across multiple decentralized devices, and each of them has data subset. The Flower client helps the global model of the client be better by accumulating the parameters from these devices. **In our scenario we used 2 to 5 clients.**

6 Evaluations

This section of the report presents the implication of the different experiments conducted to determine accuracy, precision etc. The models explained in the sections above. The training of the model has been done using 80 % of the entire data and the performance of the algorithm was carried out using the remaining 20 % has applied the following statistics on the test dataset to find out the following statistics.

6.1 Experiments Done Using Centralized Model

The first model of the annotation has been trained by centralized learning with 50,100,150 epochs. The Centralized model with 150 Epochs showed best performance metrices as compared to 50 and 100 epochs. The following evaluation parameters have been obtained:

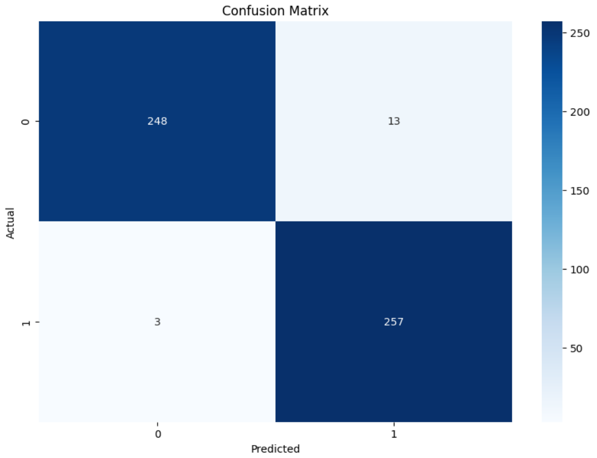


Figure 3: Confusion matrix for Centralized Model with 150 Epochs

	Evaluation Metrices				
Centralized Learning Model	**Number of Epochs**	**Precision**	**Accuracy**	**Recall**	**F1-score**
	50	94.89%	94.63%	94.63%	94.62%
	100	95.28%	95.01%	95.01%	95.005
	150	96.93%	97.00%	96.93%	96.93

Table 5: Evaluation Metrices

6.2 Experiment Done Using Federated learning with Flower

The second model of the annotation has been trained by federated learning with Flower (Epochs 50,100,150) and the acceptable levels of scores. Multiple clients were used for analysis. The following evaluation parameters have been obtained:

Federated Learning with Flower	Precision	Accuracy	Recall	F1-score
	100%	90%	100%	90%

Table 6 Evaluation Metrices with Single Client

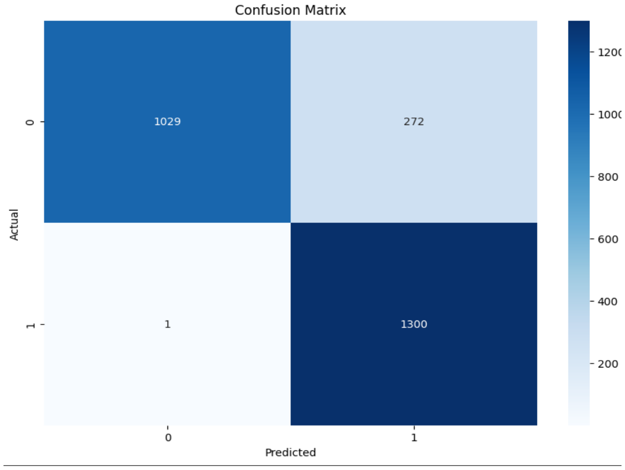


Figure 4: Confusion matrix for Federated Learning with Flower

Evaluation Metrices						
Federated Learning with Flower	Number of Clients	Epochs	Precision	Accuracy	Recall	F1-score
	2	50	88.98%	88.68%	88.68%	88.65%
	2	100	85.03%	85.04%	85.03%	85.03%
	2	150	81.49%	81.38%	81.37%	81.38%
	4	50	85.10%	80.23%	80.23%	79.53%

Federated Learning with Flower	4	100	76.79%	84.24%	76.97%	75.69%
	4	150	65.64%	78.01%	65.64%	61.42%
	5	50	78.54%	77.54%	77.54%	77.44%
	5	100	86.39%	86.18%	86.18%	86.16%
	5	150	80.85%	80.81%	80.80%	80.80%

Table 7 Evaluation Metrices FL Model with Multiple Clients

6.3 Discussion

The assessment of the models was to check their performance in terms of accuracy, precision, recall and F1-score using the training and testing set split 80:20. The first model that was based on centralized learning provided high results with accuracy above 94% with multiple epochs of 50,100 and 150. The centralized machine learning model with 50 epochs yielded accuracy of 94.63%. precision of 94.89%, recall of 94.63%, and an F1-score of 94.62%. Similarly, for 100 epochs the accuracy and evaluation metrices accuracy of 95.01%, 95.28% of precision, 95.01% of recall, and 95% of F1-score. In the final evaluation of 150, the accuracy and evaluation metrices increased more than 96 % with accuracy of 96.93%, precision of 97%, recall of 96.93% and f1-score of 96.93%. The number of epochs were not increased more than 150, as increasing number of epochs can lead to overfitting. This suggests, since centralized learning where all the data is processed at single point the accuracy and robustness of the model will be higher as compared to FL.

On the other hand, the second model, which is federated learning with Flower, achieved perfect precision and recall at 100%, but a lower accuracy and F1 score at 90%, as seen in Table 6 and Figure 4. In this FL model, the high precision value means that model has correctly identified most of the attacks as an intrusion. This is crucial in minimizing false positives which can be costly and time consuming to investigate. Similarly, a recall of 100% means that model successfully detected intrusion in the dataset. Sometimes it is critical that no intrusion goes undetected, which is vital for maintaining network security. Accuracy of 90% means that 90% of the model predictions are correct. While this is a reliable performance indicator, it may not fully capture the model's effectiveness due to high class imbalance issues in large datasets as in our case.

In the last experiments of the federated learning model, multiple clients Table 7 were used to check the model's performance. The main purpose was to check by increasing number of clients can model performance improves? The model was trained with 2,4 and 5 clients. With 2 clients and 50 epochs, higher precision and recall was obtained, indicating balanced and accurate intrusion detection. However, on increasing the epochs to 100, 150 there was drop in evaluation metrics which is due to overfitting. With 4 clients and 50 epochs there was a good balance of precision and recall, indicating robustness with more clients. But increasing epochs there was drop in performance indicating potential model degradation. With 5 clients and 100 epochs, suggested better generation as compared 50 and 100 epochs. In short, issues like data heterogeneity and communication overhead occur when scaling the number of clients in FL. Ensuring robust aggregation mechanisms in the flower framework can be used to handle client's data effectively.

7 Conclusion and Future Work

This research's objective was to increase the efficiency of the network intrusion detection systems (NIDS) by implementing federated learning (FL). Thus, using the TII-SSRC-23 dataset, we illustrated that FL could be a solution to the problem that is caused by the central model. The results showed that the centralized models had high accuracy and FL with flowers had moderate performance measures; FL models preserved the privacy of data and had high precision and recall. Thus, the primary disadvantage of the FL approach is its higher computational complexity and the difficulty of synchronizing the updates across the different systems; however, the FL approach was effective in the intrusion detection task. This indicates that FL could be a vital approach for creating efficient and private NIDS solutions in the current and future environment of cybersecurity threats. In conclusion, one limitation of the research work was high class imbalance issue in terms of attack in the dataset. The attempt was made to address the class imbalance using Near miss under sampling method, but it was not done completely. This limitation should have been addressed using other techniques.

Future work will include the fine-tuning of the FL model proposed in this work using the Flower framework to overcome the issues and improve the system performance. Some of the techniques like differential privacy and secure multiparty computation will be considered to enhance the protection of model updates during the transmission. Furthermore, attempts will be made to enhance the management of the update process in systems that are independent to minimize time delays when implementing the real-time NIDS. However, using more comprehensive and diverse datasets will enable the assessment of the FL model's viability and flexibility concerning numerous cyber threats. Thus, through enhancing the FL approach, we strive to enhance the development of NIDS that can provide adequate protection to the network infrastructures.

References

Ahanger, T. A., Aldaej, A., Atiquzzaman, M., Ullah, I. and Yousufudin, M. (2022) 'Federated learning-inspired technique for attack classification in IoT networks', *Mathematics*, 10(12), 2141. doi: 10.3390/math10122141.

Alazab, A., Khraisat, A., Singh, S. and Jan, T. (2023) 'Enhancing privacy-preserving intrusion detection through federated learning', *Electronics*, 12(16), 3382. doi: 10.3390/electronics12163382

Attota, D. C., Mothukuri, V., Parizi, R. M. and Pouriyeh, S. (2021) 'An ensemble multi-view federated learning intrusion detection for IoT', *IEEE Access*, 9, pp.117734–117745. doi: 10.1109/access.2021.3107337.

Bag, S. (2024) 'A beginners guide to federated learning', *Analytics Vidhya*, 28 May.
Available at: https://www.analyticsvidhya.com/blog/2021/05/federated-learning-a-beginners-guide/
[Accessed 1 August 2024].

Chen, Z., Lv, N., Liu, P., Fang, Y., Chen, K. and Pan, W. (2020) 'Intrusion detection for wireless edge networks based on federated learning', *IEEE Access*, 8, pp. 217463-217472. doi: 10.1109/ACCESS.2020.3041793.

Das, A. and Brunschwiler, T. (2019) 'Privacy is what we care about: Experimental investigation of federated learning on edge devices', in *Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*. New York, NY, USA, 10-13 November 2019, pp.39–42. doi: 10.1145/3363347.3363365.

Herzalla, D. (2023) *TII-SSRC-23 Dataset.*
Available at: https://www.kaggle.com/datasets/daniaherzalla/tii-ssrc-23
[Accessed 1 August 2024].

Hou, J., Liu, F. and Zhuang, X. (2019) 'A new intrusion detection model based on GRU and salient feature approach', in *International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications*. Guangzhou, China, 12-15 November 2019. doi: 10.1007/978-981-15-1304-6_32

Huang, J., Chen, Z., Liu, S.-Z., Zhang, H. and Long, H.-X. (2024) 'Improved intrusion detection based on hybrid deep learning models and federated learning', *Sensors*, 24(12), 4002. doi: 10.3390/s24124002.

Imbalanced learn (2024) *Under-sampling*. Available at: https://imbalanced-learn.org/stable/under_sampling.html [Accessed 1 August 2024].

Jin, Y., Liu, Y., Chen, K. and Yang, Q. (2023) *Federated learning without full labels: A survey.* doi: 10.48550/arXiv.2303.14453

Li, L., Fan, Y., Tse, M. and Lin, K.-Y. (2020) 'A review of applications in federated learning', *Computers & Industrial Engineering,* 149, 106854. doi: 10.1016/j.cie.2020.106854.

Li, B., Wu, Y., Song, J., Lu, R., Li, T. and Zhao, L. (2021) 'DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems', *IEEE Transactions on Industrial Informatics*, 17(8), pp. 5615-5624. doi: 10.1109/TII.2020.3023430

Li, J., Tong, X., Liu, J. and Cheng, L. (2023) 'An efficient federated learning system for network intrusion detection', *IEEE Systems Journal*, 17(2), pp. 2455–2464. doi: 10.1109/JSYST.2023.3236995

Qazi, E.-H., Imran, M., Haider, N., Shoaib, M. and Razzak, I. (2022) 'An intelligent and efficient network intrusion detection system using deep learning', *Computers and Electrical Engineering*, 99, 107764. doi: 10.1016/j.compeleceng.2022.107764.

Rahman, S. A., Tout, H., Talhi, C. and Mourad, A. (2020) 'Internet of things intrusion detection: Centralized, on-device, or federated learning?', *IEEE Network*, 34(6), pp. 310–317. doi: 10.1109/mnet.011.2000286.

Shastri, Y. (2023) 'A step-by-step guide to federated learning in computer vision', *V7labs*, 3 February. Available at: https://www.v7labs.com/blog/federated-learning-guide
[Accessed 1 August 2024].

Shukla, S., Raghuvanshi, A. S., Majumder, S. and Singh, S. (2024) 'FedHNN: A federated learning based hybrid neural network for real-time intrusion detection systems', in *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*. Dehradun, India, 15-16 March 2024, pp.693–697. doi: 10.1109/DICCT61038.2024.10533096.

Wang, K., Li, J. and Wu, W. (2022) 'An efficient intrusion detection method based on federated transfer learning and an extreme learning machine with privacy preservation', *Security and Communication Networks*, 2022, 2913293. doi: 10.1155/2022/2913293.

Zhai, F., Yang, T., Chen, H., He, B. and Li, S. (2023) 'Intrusion detection method based on CNN–GRU–FL in a smart grid environment', *Electronics*, 12(5), 1164. doi: 10.3390/electronics12051164