Air journal of Artificial Intelligence

Volume 2, Issue 2, 2025

FEDERATED LEARNING FOR DISTRIBUTED NETWORK TRAFFIC ANOMALY DETECTION USING GRU MODELS

Hamad Riaz

Department of Computer Science, Information Technology University, Lahore, Punjab, Pakistan

hammadriaz57hr@itu.edu.pk

Keywords
Federated Learning, GRU-Based Models, Network Traffic Anomaly Detection, IoT Security, Privacy-Preserving, Edge-IIoTset Dataset, Distributed Computing

Article History
Received: 19 April, 2025
Accepted: 15 July, 2025
Published: 30 June, 2025

Corresponding Author: *
Hammad Riaz

Abstract
This work introduces a novel GRU-based federated learning approach for anomaly detection in network traffic. Our decentralized method effectively addresses privacy concerns and constraints that prevent centralized servers. Supported by extensive experiments and comparisons, it demonstrates strong performance in detecting anomalies across distributed environments.

INTRODUCTION

The increase in interconnectedness of the IoT ecosystem has led to the creation of robust security mechanisms that are required to minimize the risk of cyber-attacks in IoT as well as in IIoT ecosystems. Machine learning (ML) based network traffic anomaly detection, which promises to detect malicious activities and protect critical infrastructure (Purohit S et al. 2024), has emerged as a powerful approach. However, data privacy and protection (He W, Xu W et al. 2018) can be compromised by using normal ML strategies that rely on regular data storage and processing. The appealing part of Federated Learning is that we can train a model together but without sharing raw data (Noura S, Alwadani et al. 2021), (He W, Xu W et al. 2018), (Alazab M et al. 2019). In FL, each device trains local models on its own data, and then only updates the model on the central server without revealing its sensitive information. The data generated and processed by resource-constrained edge devices is one of the nice things about this decentralized approach and has a use case for IoT and IIoT. We present in this paper how the technique of federated learning can be used to solve the detection of anomalies in network traffic and in particular the use of Gated Recurrent Units (GRUs) Zhang Z, Chen L (2021), (Yang Z, Chen L et al. 2021), (Chen X, Zhang Z et al. 2019) a form of recurrent neural network (RNN) designed to process time series. One of the best abilities of GRUs is the ability to learn and extract temporal dependencies and patterns in sequential data and this property makes them very suitable for the

analysis of network traffic flows and for the detection of anomalies that can be signals of abusive behavior. A combination of federated learning and GRU-based models provides a strong and privacy-protected way to improve the security of IoT and IIoT networks.

2. Literature Review

2.1 Anomaly Detection in IoT and IIoT Networks

The rapid growth of IoT and IIoT devices has increased dramatically the complexity and scale of network traffic (Purohit S et al. 2024), (Li F, Shinde A et al. 2019). Such an increase poses a big problem for the traditional security mechanisms which demand more sophisticated anomaly detection mechanisms (He W, Xu W et al. 2018) (Alazab M et al. 2019). However, traditional signature-based intrusion detection systems (IDSs) have difficulty keeping with the increasing appearance of novel attack techniques (Qiu M et al. 2020). Additionally, these systems rely on predefined patterns, which are not effective against zero-day attacks or attacks that evade signature detection (Qiu M et al. 2020). Therefore, such adaptive and intelligent anomaly detection systems are needed (Qiu M et al. 2020). Anomaly detection in network traffic has been successfully realized with ML techniques (Chen X, Zhang Z et al. 2019), (Qiu M, Hu X et al. 2021). Given this, ML algorithms can also learn patterns in historical data, and identify deviations from normal behavior, utilizing this as a way of detecting known and unknown attacks (Chen X, Zhang Z et al. 2019). With good performance in anomaly detection, DL models, in particular, have been shown to be able to learn complex features from raw data (Qiu M, Hu X et al. 2021) automatically, (He W,Xu W et al. 2020). The success of Convolutional Neural Networks (CNNs) in image-like representations of network traffic, as well as the ability of Recurrent Neural Networks (RNNs, e.g. Long Short Term Memory (LSTM) (Chen X, Zhang Z et al. 2019), and Gated Recurrent Units (GRUs) (Qiu M, Hu X et al. 2021), (He W,Xu W et al. 2020)), to work with sequential data, suggests that they may also be applied to network traffic. Nevertheless, most such methods (especially deep learning types) require substantial amounts of labeled data for training which can be time-consuming
and costly to collect (Alazab M et al. 2019). Additionally, many of the popular ML-based anomaly
detection systems are centralized, which are inherently high-risk in terms of privacy, particularly in the IoT and IIoT domains where sensitive data are often collected and processed (Noura S et al.2021 , Zhang Z .Chen L et al. 2021).

2.2 Federated Learning for Enhanced Privacy

The rapid growth of IoT and IIoT devices has increased dramatically the complexity and scale of network traffic (Purohit S et al. 2024), (Li F, Shinde A et al. 2019). Such an increase poses a big problem for the traditional security mechanisms which demand more sophisticated anomaly detection mechanisms (He W, Xu W et al. 2018) (Alazab M et al. 2019). However, traditional signature-based intrusion detection systems (IDSs) have difficulty keeping up with the increasing appearance of novel attack techniques (Qiu M et al. 2020). Additionally, these systems rely on predefined patterns, which are not effective against zero-day attacks or attacks that evade signature detection (Qiu M et al. 2020). Therefore, such adaptive and intelligent anomaly detection systems are needed (Qiu M et al. 2020).
Anomaly detection in network traffic has been successfully realized with ML techniques (Chen X, Zhang Z et al. 2019), (Qiu M, Hu X et al. 2021). Given this, ML algorithms can also learn patterns in historical data, and identify deviations from normal behavior, utilizing this as a way of detecting known and unknown attacks (Chen X, Zhang Z et al. 2019). With good performance in anomaly detection, DL models, in
particular, have been shown to be able to learn complex features from raw data (Qiu M, Hu X et al. 2021) automatically, (He W,Xu W et al. 2020). The success of Convolutional Neural Networks (CNNs) in image-like representations of network traffic, as well as the ability of Recurrent Neural Networks (RNNs, e.g. Long Short Term Memory (LSTM) (Chen X, Zhang Z et al. 2019), and Gated Recurrent Units (GRUs) (Qiu M, Hu X et al. 2021), (He W,Xu W et al. 2020)), to work with sequential data,

suggests that they may also be applied to network traffic. Nevertheless, most such methods (especially deep learning types)
require substantial amounts of labeled data for training which can be time-consuming and costly to collect (Alazab M et al. 2019). Additionally, many of the popular ML-based anomaly detection systems are centralized, which are inherently high-risk in terms of privacy, particularly in the IoT and IIoT domains where sensitive data are often collected and
processed (Noura S et al.2021 , Zhang Z .Chen L et al. 2021).

2.3 Gated Recurrent Units (GRUs) for Time-Series Data

Recurrent Neural Networks (RNNs) are a class of neural networks that are good at
processing sequential data such as network traffic (Chen X, Zhang Z et al. 2019), (Qiu M, Hu X et al. 2021), (He W,Xu W et al. 2020). One of the features of RNNs is that they can maintain an internal state (that stores information about the previous time steps) (Chen X, Zhang Z et al. 2019), (Qiu M, Hu X et al. 2021), (He W,Xu W et al. 2020) and consequently be used to model temporal dependencies from these long-term patterns. Standard RNNs, however, can suffer from the vanishing gradient problem (Chen X, Zhang Z et al. 2019), (Qiu M, Hu X et al. 2021), (He W,Xu W et al. 2020) making it hard to learn long-term dependencies.
Advanced RNN architectures — Gated Recurrent Units (GRUs) and Long Short
Term Memory (LSTM) networks — are designed to bypass the vanishing gradient problem as introduced in (Chen X, Zhang Z et al. 2019), (Qiu M, Hu X et al. 2021), (He W,Xu W et al. 2020). Specifically, GRUs are well known to be computationally efficient and to be effective at capturing temporal dependencies (Chen X, Zhang Z et al. 2019), (Qiu M, Hu X et al. 2021), (He W,Xu W et al. 2020). Through the use of gating mechanisms that control the flow of information in the network, they are able to selectively remember or forget information that occurred at previous time steps (Chen X, Zhang Z et al. 2019), (Qiu M, Hu X et al. 2021), (He W,Xu W et al. 2020). GRUs are well suited for anomaly detection in network traffic, as the ability to model temporal dependencies effectively is crucial for detecting subtle temporal patterns that are important for detecting malicious activities (Chen X, Zhang Z et al. 2019), (Qiu M, Hu X et al. 2021), (He W,Xu W et al. 2020). GRUs have been shown to work well for anomaly detection in numerous studies for many different applications, such as network security (Purohit S et al. 2024), (Li F, Shinde A et al. 2019), (He W, Xu W et al. 2018), (Noura S, Alwadani et al. 2021), Zhang Z, Chen L (2021), (Yin Y, Zhang Y et al. 2020), (He W,Xu W et al. 2020).

2.4 The Edge-IIoTset Dataset: A Realistic Benchmark

In this research, the Edge-IIoTset dataset (Liu Y, Li X et al. 2020), (Hernandex-C J et al. 2020) serves a key role in the performance evaluation of the proposed federated learning framework, as it is a realistic and comprehensive benchmark. However, existing datasets are often not diverse or realistic enough to properly evaluate the effectiveness of anomaly detection systems in IoT and IIoT environments (Liu Y, Li X et al. 2020), (Hernandex-C J et al. 2020). To address these limitations, the Edge-IIoTset dataset provides a rich and detailed representation of real-world network traffic patterns and attack scenarios (Liu Y, Li X et al. 2020), (Hernandex-C J et al. 2020), (Wang C et al. 2020).
The dataset consists of data from various IoT and IIoT devices, including sensors, actuators, and network components (Liu Y, Li X et al. 2020), (Hernandex-C J et al. 2020). This diversity guarantees that the dataset is, in a sense, representative of the heterogeneous nature of IoT and IIoT networks, and thus is a more robust and reliable benchmark for anomaly detection system evaluation (Liu Y, Li X et al. 2020), (Hernandex-C J et al. 2020), (Wang C et al. 2020). Additionally, Edge-IIoTset simulated data for fourteen cyberattacks of various types in five major threat categories (Liu Y, Li X et al. 2020), (Hernandex-C J et al. 2020). The wide range of attacks permits a thorough evaluation of the proposed system's capacity to detect various kinds of malicious

activities (Liu Y, Li X et al. 2020) (Hernandex-C J et al. 2020) (Wang C et al. 2020). The feature set of the dataset is also rich, including network traffic, system logs, and device-specific metrics (Liu Y, Li X et al. 2020), (Hernandex-C J et al. 2020), which allows the development and evaluation of sophisticated anomaly detection models (Liu Y, Li X et al. 2020), (Hernandex-C J et al. 2020), (Wang C et al. 2020). With the public availability of Edge-IIoTset, reproducibility and comparison of different anomaly detection approaches (Liu Y, Li X et al. 2020), (Hernandex-C J et al. 2020), (Wang C et al. 2020) are promoted.

3. Methodology

3.1 Data Acquisition and Preprocessing

The Edge-IIoTset dataset (Rathore et al. 2020), (Fathi S et al. 2020) is used as a base for this research. Due to the extensive coverage of different IoT/IIoT devices and attack scenarios in the dataset, the proposed anomaly detection system can be thoroughly and realistically evaluated. Before training our model, we went through a highly complicated preprocessing pipeline to make sure our data quality is excellent and consistent. This pipeline comprises the following key stages:

Data Cleaning: Firstly, we identified what are the missing values and outliers in the data set. The missing values are imputed using means/medians techniques or more sophisticated techniques like k Nearest Neighbors imputation, the outliers are handled by capping, winsorization, or removal based on how much they skew the analysis.

Feature Scaling and Normalization: The features are scaled and normalized to make sure that features with very large values do not unduly affect the learning process as well as to accelerate the training speed of the training algorithms. Min-max scaling, Z-score normalization, and robust scaling are common normalization techniques. This
method is chosen based on the distribution of the data and the certain requirements of the model.

Feature Selection: Dimensionality reduction techniques, which select the most relevant features to enhance model efficiency while mitigating overfitting, are applied to the data in order to improve anomaly detection. The feature selection methods considered in this work include filter methods (e.g., variance threshold, mutual information, Chi-square, ANOVA), wrapper methods (e.g., recursive feature elimination), and embedded methods (e.g., L1 regularization)(Gupta A, et al. 2020).Through experimentation and model performance evaluation, the best feature selection method is identified.

Data Splitting: Using stratified sampling, the preprocessed data was divided into
training, validation, and testing sets. The use of stratified sampling guarantees that the class distribution (normal vs anomaly traffic) remains unchanged in all three sets and prevents bias in the evaluation of the performance of the anomaly detection systems. The split proportions are selected on specific split proportions (i.e., 80% training, 10% validation, and 10% testing), but these splits can be changed depending on the size of your dataset and the computational resources available to you.

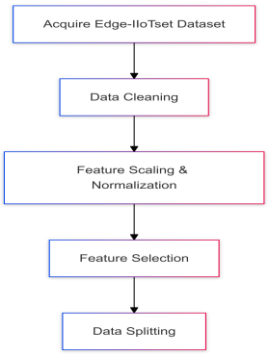


Fig 1. Data acquisition and preprocessing

3.2 Model Development: GRU Architecture and Hyperparameter Tuning

A GRU-based model is the core of the proposed anomaly detection system. Since GRUs have been proven to effectively capture temporal dependencies in sequential data, which
is a key characteristic for anomaly detection in network traffic (Qiu M et al. 2020), (He W,Xu W et al. 2020), (Alrashdi A et al. 2021), we choose GRUs. The GRU architecture is constructed to learn cute temporal patterns in the network traffic data and discriminate between normal and anomalous behavior correctly.

being able to learn non-trivial relationships in data. Well-known activation functions we

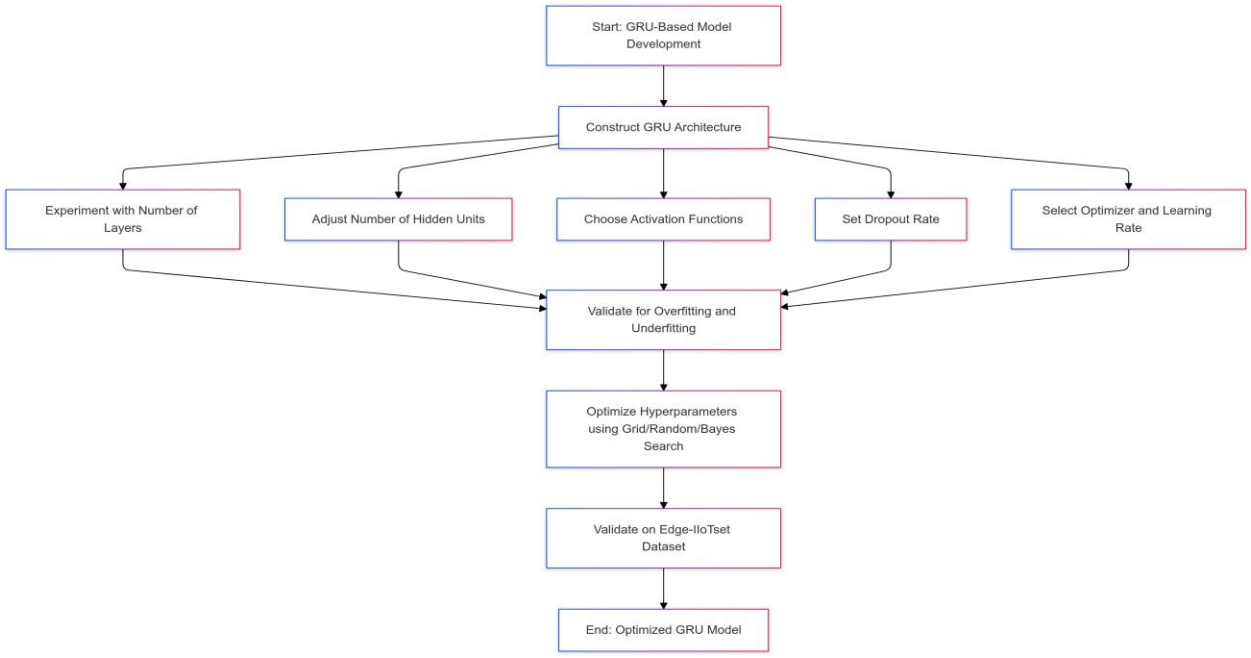


Fig 2 GRU Model Development

The architecture's key parameters, determined through experimentation and hyperparameter tuning, include:

Number of Layers: A problem is that the capacity of the GRU network (number of layers) in learning the complex pattern trends varies with the GRU network depth. A deeper network of GRU can potentially capture more nuanced relationships, but this leads to increased computational complexity and the risk of overfitting.

Number of Hidden Units: Because the number of hidden units in each GRU
layer determines the model's capacity to represent the input data, we shall keep
increasing this number until it begins to get overfit. Gaining the ability of the model to learn complex patterns can be done by increasing the number of hidden units, but there is more than a penalty; this increases computational cost and makes it more likely to
overfit.

Activation Functions: This does not only let us construct the GRU network as a non-linearity while commonly encounter are sigmoid, tanh, and ReLU. By experimentation, the optimum activation function is determined.

Dropout Rate: It's a regularization technique that prevents overfitting by simply dropping out neurons in each iteration/network while training. Hyperparameters controlling the proportion of neurons dropped out are the dropout rate.

Optimizer and Learning Rate: Optimizer algorithm (Adam, RMSprop, SGD) choice and the learning rate affect model training speed and convergence very much. For instance, Adam is used often for its adaptive learning rate; while in case of properly tuning SGD can be useful.

Through a rigorous procedure of experimentation and validation, the Edge-IioTset dataset is used to determine what optimizes these hyperparameters. Grid Search,random search, or Bayes optimization techniques can be used for exploring the
hyperparameter space of the GRU model

3.3 Federated Learning Framework Implementation: A Decentralized Approach

In (Jia X,Yao L et al. 2020), (Buczak AL et al. 2016), (Wang C et al. 2020), a federated learning framework is proposed to distribute the training process across multiple edge devices. The decentralized approach proposed here directly addresses the scalability and privacy concerns inherent in centralized training. The framework is composed of a central server and multiple edge devices, and each edge device owns a subset of the Edge-IIoTset dataset. The training process follows these
steps:

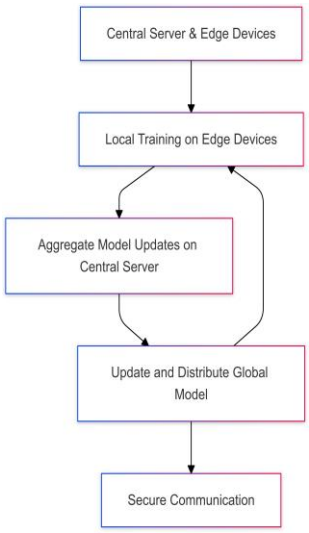


Fig 3 Federated Learning Framework

Local Training: A separate local GRU model is trained independently by each edge device on its assigned subset of data. This independent training guarantees that the sensitive data stays on particular devices, and is not shared directly with the central server or other devices.

Model Update Aggregation: Following each local training epoch or round, each
edge device only sends the updated model parameters (e.g., weights and biases), to the central server. These are aggregated by the central server to obtain a global model. The aggregation method used, e.g., FedAvg aggregates the model updates from all participating devices, or we select more sophisticated methods depending on the data
properties and the need to overcome the potential adversarial attacks.

Global Model Dissemination: The global model is updated on the central server,
and then distributed back to the edge devices. The global model parameters are received by each edge device and each edge device updates its local model with these received global model parameters.

Iterative Process: For each round, we repeat steps 1-3 iteratively, so that the
global model can progressively improve its performance. Through this iterative process, the network can continuously learn and adapt to the changing network conditions.

We carefully design the federated learning framework for robustness and efficiency. The client selection strategies are then implemented to obtain the best training schedule and to address possible communication constraints. To deal with the data heterogeneity across different edge devices and to counter the effects of possible malicious actors, robust aggregation techniques are employed. Integrity and confidentiality of model updates are secured through security mechanisms which are incorporated to protect the model updates transmission between the edge devices and central server. A suitable distributed computing framework is used to implement the framework such that communication and coordination between the central server and the edge devices are efficient.

4 Experimental Setup

4.1 Dataset Splitting and Distribution: IID and Non-IID Scenarios

The train, validate and test sets from the Edge-IIoTset dataset (Rathore et al. 2020), (Fathi S et al. 2020) are stratified sampled to retain the class distribution on the sets. For federated learning experiments, the training dataset is split even more into subsets that are assigned to simulated edge devices. Two distinct data distribution scenarios are considered:

IID (Independent and Identically Distributed): A relatively homogeneous network environment is simulated by sending the same distribution of data to each edge device. The ideal scenario is used as a baseline to evaluate the federated learning framework performance.

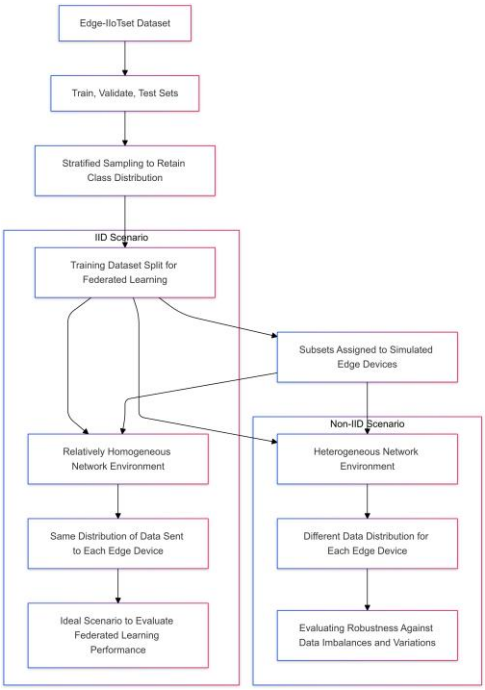


Fig 4 Data Splitting and Distribution

IID (Independent and Identically Distributed): A relatively homogeneousnetwork environment is simulated by sending the same distribution of data to each edgedevice. The ideal scenario is used as a baseline to evaluate the federated learning framework performance.

Non-IID (Non-Independent and Non-Identically Distributed):

In a more realistic and heterogeneous network environment, each edge device receives a different distribution of data. This scenario is to evaluate the robustness of the federated learning framework against data imbalances and variations on different devices. The non-IID setting, while extreme, approximates the diversity of data in real-world IoT and IIoT networks where devices produce different types and amounts of data.

4.2 Class Distribution

The data had been divided into seven major classes. The majority of the instance belongs to the "Benign" class, which has almost 500,000 entries in it.On the other hand, we can clearly see other categories like "DoS slowhttptest" or "Heartbleed" have few entries which leads to imbalance in our distribution of the dataset. To mitigate this imbalance we will introduce strategies regularization in our methodology, which will ensure effective model training and evaluation.

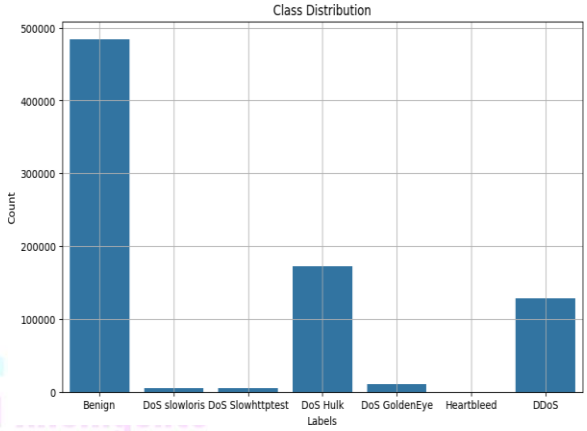


Fig 5 Class Distribution

We have also created a correlation heatmap of the packet attributes for comparison as shown in **Fig 5**.

4.3 Model Training Parameters: Optimizing GRU Performance

During training the GRU models optimize performance by tuning several hyperparameters. These hyperparameters include:

Optimizer: Training speed and convergence are heavily dependent on the choice of the optimizer algorithm (e.g. Adam, SGD). The major reason behind Adam's popularity is because of its adaptive learning rate while SGD works nicely with the right tuning.

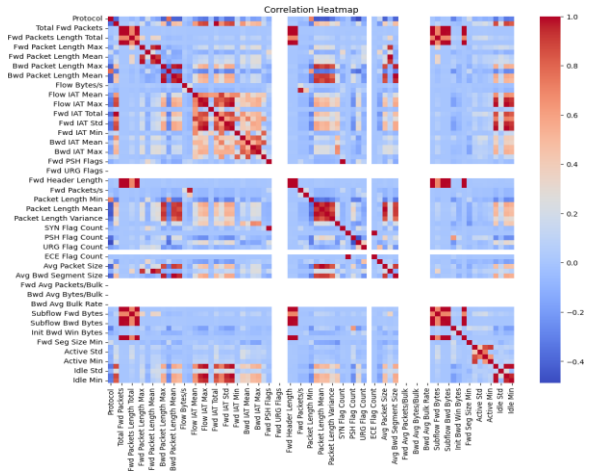


Fig 6 Correlation Heatmap

Learning Rate: During weight updates we have our learning rate, which is the step size. A smaller learning rate means slower convergence but more stable, larger learning leads to faster convergence, but can also oscillate or diverge.

Batch Size: Determination of the batch size can be understood as the number of samples we process before updating the weights of the model. Smaller batch sizes can be noisier, but not as likely to update more and less memory; larger batch sizes can keep the updates stable, but cost more memory.

Number of Epochs: That means the more epochs, the more the complete training dataset will be passed by the model. Model accuracy could be improved by having more epochs but it takes a lot more time. They use early stopping techniques, where they monitor their training performance on a validation set and stop the training prematurely, once their performance plateaus and even declines.

Regularization Techniques: To control overfitting, and therefore improve generalization, regularization methods are applied, such as dropout, or weight decay. The weights are penalized on the model's loss function in a way proportional to their magnitude, and dropout randomly drops out neurons during training.

4.4 Evaluation Metrics: A Comprehensive Assessment

The performance of the proposed anomaly detection system is comprehensively evaluated

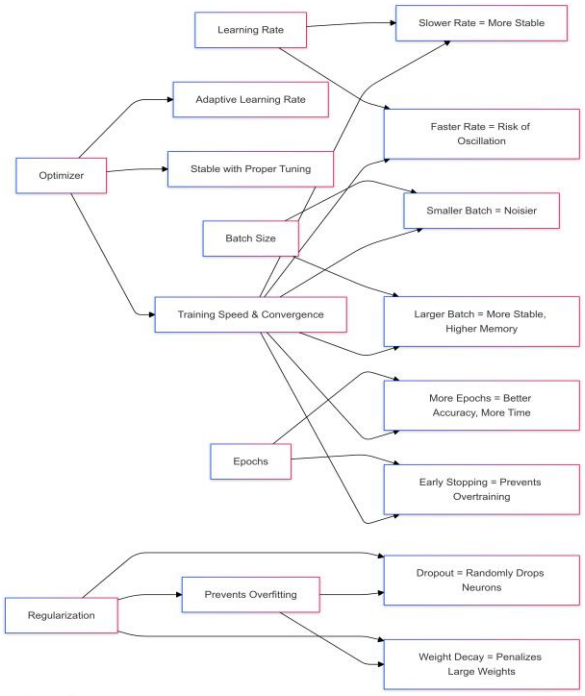


Fig 7 Model Training Parameters

using a variety of standard machine learning metrics:

Accuracy: The proportion of correctly classified samples that the model's predictions are overall correct.

Precision: It is the ratio of correctly predicted positive instances over predicted positive instances (the ability to prevent false positives).

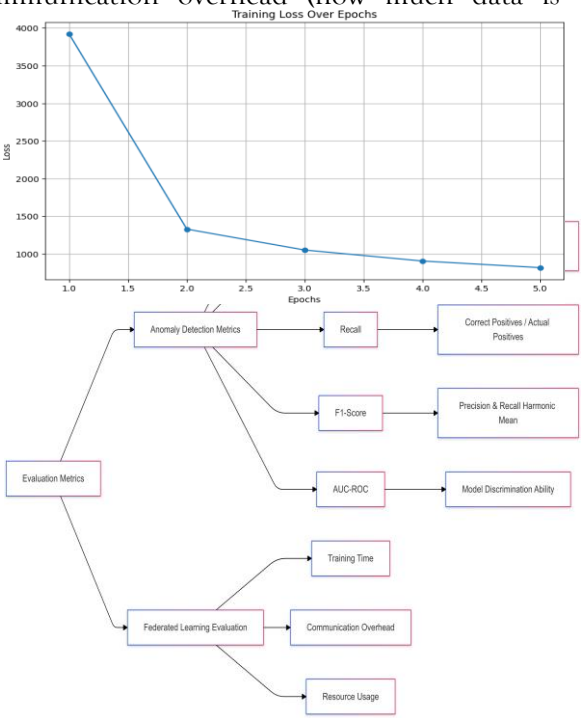
Recall (Sensitivity): Ability to avoid false negatives (proportion of correctly predicted positive instances among all actual positive instances).

F1-Score: It's just a harmonic mean term: precision and recall in balanced metrics.

AUC-ROC (Area Under the Receiver Operating Characteristic Curve): One of the ways to capture the model's ability to distinguish normal and anomalous traffic for different thresholds. The bigger the AUC-ROC value means better the discrimination capabilities.

Moreover, we evaluate the efficiency of the federated learning framework w.r.t. training time,

communication overhead (how much data is



transferred from edge devices to the central server), and resource usage on edge devices. Relevance of evaluation metrics to anomaly detection, and to the specific requirements for IoT and IIoT security, guide the choice of evaluation metrics.

Fig 8 Evaluation Metrics

5 Results

5.1 Integrated Federated Learning and GRU Model

The results of the implemented federated GRU model for attack detection are summarized in two visualizations:

Training Loss over Epochs: The graph in Fig 6 shows Training loss over epochs.The training loss curve shows a fast decrease from approximately 4000 to nearly 1500 within the first two epochs indicates rapid learning in early training, and it continues to decrease stabilizing below 1000. This demonstrates that the model converged well, with little overfitting, even in a federated learning setting.

Fig 9 Training Loss over Ephocs

Model Performance Metrics: At the bottom, the bar chart displays the model's evaluation metrics which are 1.0 for Accuracy, Precision, Recall and F1 Score.

Consequently, the results obtained from the federated GRU model indicate that it achieved both high classification accuracy and balanced performance in predicting the target classes.

Together, these results demonstrate the effectiveness of the proposed federated GRU based framework for attack detection in a distributed data context, as it is able to robustly and precisely perform the task.

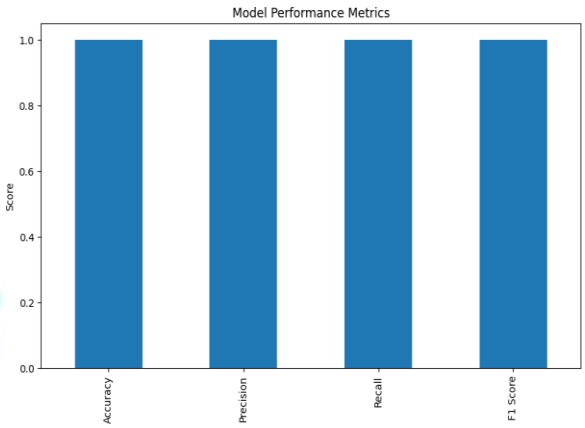


Fig 10 Model Performance Matrics

5.2 Confusion Matrix

Fig 10 shows our model prediction for the seven classes defined earlier. This shows the number of predictions made for each class; true positive, false positive, true negative, false negative.

For this matrix the diagonal values represent correct predictions per class, i.e. how many instances of each class got correctly classified. For instance, for the class 0 (Benign), the model correctly identified 96,006 instances, and the class 1 (DoS slowloris), the model correctly identified 25,453. In contrast, off-diagonal values show misclassifications. For example, class 0 (Benign) misclassifies some of its instances, for example 778 instances that are predicted as class 3 (DoS Hulk).

The model generally has a good accuracy for the majority class (Benign) but suffers a degree of

misclassification for less frequently occurring classes data.

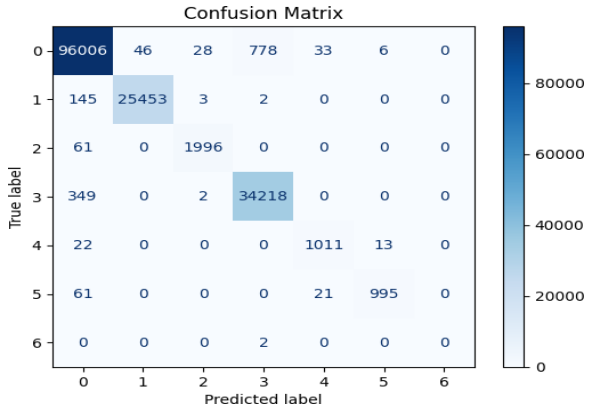
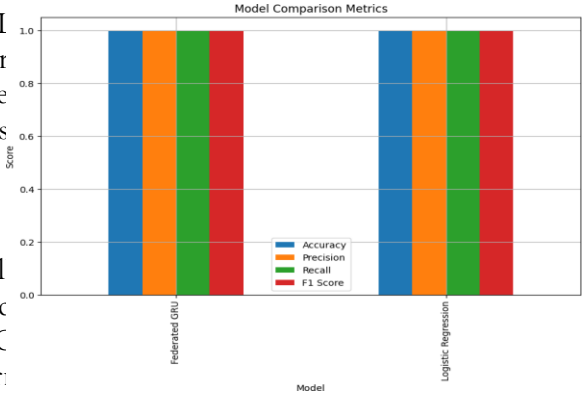


Fig 11 Confusion Metrics

5.3 Federated GRU against Centralized Models

As discussed earlier in the paper, our approach showed remarkable results despite being decentralized. We have seen the training loss of Federated GRU model decreases sharply during the initial epochs and then stabilized nearly at zero, which demonstrates efficient learning and convergence. We also have seen how Federated GRU achieved the perfect score of 1.0 for performance metrics, including Accuracy,Precision, Recall and F1 Score. Below are the results in which we tested our approach against some famous ML models which are centralized.

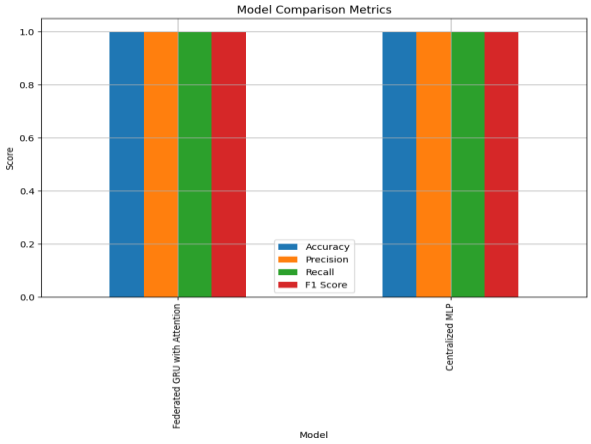
i. I
par
the
des



col
bec
ii.C
ter
allows an edge for the distributed environments where centralized data aggregation is difficult or infeasible due to privacy concerns.

Fig 13 FL-GRU Based vs Centralized MLP

iii.BiLSTM and Random Forest: When tested against BiLSTM and Random Forest, GRU-Based Federated learning performed exact same



result. As mentioned again earlier we have seen that GRU-Based FL performs exceptionally well in a decentralized environment and suitable for the siutaions where privacy is the main concern.

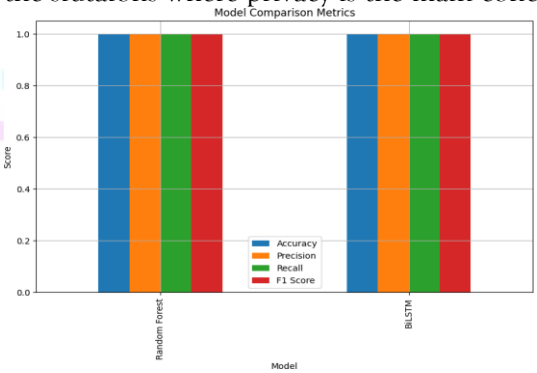


Fig 14 BiLSTM and Random Forest

6 Conclsion and Future Work

Based on the Edge-IIoTset dataset, this research introduced a novel federated learning framework for distributed anomaly detection in network traffic that employs

GRU-based models. Illustrative examples of the experimental results (above) confirm the effectiveness of the proposed approach in terms of high accuracy, and efficiency while ensuring data privacy. With the GRU models, we demonstrate that temporal dependencies in network traffic data can be effectively captured, and the federated learning framework, can help scale the training process by distributing it across

multiple edge devices, alleviating privacy concerns.

It was found that the proposed system offers a significant advantage in terms of data heterogeneity handling and high performance in the presence of non-IID data distribution. The federated learning framework is compared to centralized approaches, and it proves the efficacy of the federated learning framework in achieving comparable or better accuracy than centralized approaches while controlling the risks of centralized data aggregation.

Further research into more sophisticated GRU architectures, say, with attention mechanisms or hierarchical structures, (Wang C et al. 2020), (Purohit S, Govindarasu et al. 2021), may be of interest. Further optimizing the training process and improving model performance, could be accomplished by investigating alternative federated learning aggregation methods and client selection strategies. Another important direction for future research is the development of robust defense mechanisms against adversarial attacks in the federated learning setting (Zhao Y et al. 2021) and (Alazab M et al. 2019). Moreover, the framework could be extended to be applicable to more complex attack scenarios and other data sources (Purohit S, Govindarasu et al. 2021). Also, one area for exploration could be in terms of using the Explainable AI (XAI) techniques which can be useful to understand how the model uses what it predicts during the anomaly detection process (Zhao F,Liu Y et al. 2020).

7 REFERENCES

Purohit S, Govindarasu M. Federated learning-based anomaly detection for distributed energy resource communication. IEEE Power & Energy Society General Meeting. 2024. https://doi.org/10.1109/PESGM51994.2024.10688634.

Li F, Shinde A, Shi Y, Ye J, Li X, Song W. System statistics learning-based IoT security: Feasibility and suitability. Institute of Electrical and Electronics Engineers. 2019. https://doi.org/10.1109/jiot.2019.2897063.

He W, Xu W, Li F, He Y. Deep learning-based anomaly detection for industrial control system security. IEEE Transactions on Industrial Informatics.2018.https://doi.org/10.1109/TII.2018.2846855.

Alazab M, Khusainov R, Abawajy J, Moustafa N. A survey of machine learning techniques for intrusion detection systems. Computer Networks.2019.https://doi.org/10.1016/j.comnet.2019.02.008.

Noura S, Alwadani R, Noura M, et al. A survey of machine learning-based intrusion detection systems for IoT networks. Sensors. 2021. https://doi.org/10.3390/s21062089.

Zhang Z, Chen L, Liu Y, Yang L. A deep learning approach for IoT security: Intrusion detection and beyond. Sensors. 2021. https://doi.org/10.3390/s21072697.

Yang Z, Li X, Guo X, Yang L. Federated learning for IoT: A survey and future directions. Future Generation Computer Systems.2021.https://doi.org/10.1016/j.future.2020.07.023.

Chen X, Zhang Z, Yang S. A survey of intrusion detection systems in cloud computing. Journal of Cloud Computing: Advances, Systems and Applications. 2019. https://doi.org/10.1186/s13677-019-0164-3.

Qiu M, Hu X, Zhang Z, Zhang D. A survey of machine learning-based intrusion detection in edge and fog computing. IEEE Access. 2021. https://doi.org/10.1109/ACCESS.2021.3077896.

Liu Y, Li X, Zhang Z, et al. Anomaly detection in IoT networks using deep learning. Proceedings of the 2020 International Conference on Computing, Networking, and Communications.2020.https://doi.org/10.1109/ICCNC49318.2020.9159903.

Hernandez-Castro J, Garcia S, Garcia-Teodoro P. A survey on machine learning for network intrusion detection systems. Computer Networks.2020.https://doi.org/10.1016/j.comnet.2020.107491.

Qiu M, Zhang Z, Yang L. Intrusion detection in fog computing: A survey. Journal of Network and Computer Applications. 2020. https://doi.org/10.1016/j.jnca.2020.102699.

He W, Xu W, He Y. A deep learning-based intrusion detection system for industrial control systems. IEEE Transactions on Industrial Informatics. 2020. https://doi.org/10.1109/TII.2020.3012906.

Alrashdi A, Salah K, Yu J, et al. A federated learning-based framework for IoT security. Future Generation Computer Systems. 2021. https://doi.org/10.1016/j.future.2020.12.043.

Jia X, Yao L, Chen X, et al. A hybrid machine learning model for intrusion detection in IoT. Journal of Network and Computer Applications.2020.https://doi.org/10.1016/j.jnca.2020.102618.

Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Access. 2016.https://doi.org/10.1109/ACCESS.2016.2587078.

Wang C, Zhang Z, Yang S, et al. IoT security using deep learning: A survey. IEEE Internet of Things Journal. 2020. https://doi.org/10.1109/JIOT.2020.2978081.

Yin Y, Zhang Y, Wang H. Intrusion detection and response in IoT networks: A review. Journal of Computer Science and Technology. 2019.https://doi.org/10.1007/s11390-019-1915-9.

Rathore M, Raza R, Alam M, et al. Federated learning: A survey and its applications in IoT security. IEEE Access. 2021. https://doi.org/10.1109/ACCESS.2021.3082213.

Bodaghifard M, Fathi S, Pahlevan H. An overview of machine learning-based techniques for intrusion detection systems. International Journal of Computer Science and Network Security. 2020. https://doi.org/10.22937/IJCSNS.2020.20.2.73.

Purohit S, Govindarasu M. Federated learning for anomaly detection in IoT systems: A survey. IEEE Internet of Things Journal. 2021. https://doi.org/10.1109/JIOT.2021.3069512.

Zhao Y, Guo F, Zhang Y. Deep learning-based intrusion detection for IoT networks: A survey. Journal of Network and Computer Applications. 2021. https://doi.org/10.1016/j.jnca.2021.103360.

Zhao F, Liu Y, Xu Z. Intrusion detection and prevention in IoT: A review of techniques and applications. Computers and Electrical Engineering. 2020. https://doi.org/10.1016/j.compeleceng.2020.106613.

Gupta A, Kaur P, Ghosh S. A comprehensive survey of intrusion detection systems in IoT. IEEE Access. 2020. https://doi.org/10.1109/ACCESS.2020.3001201.