Enhancing Intrusion Detection: Leveraging Federated Learning and Hybrid Deep Learning Models

By

Faiza Naeem

Fall-2019-MS-CS SEECS

Supervisor

Dr Safdar Abbas

Department of Computer Science
School of Electrical Engineering Computer Science (SEECS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan

August 2023

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Enhancing Intrusion Detection: Leveraging Federated Learning and Hybrid Deep Learning Models" written by FAIZA NAEEM, (Registration No 00000318806), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: ____________

Name of Advisor: Dr. Safdar Abbas Khan

Date: 03-Aug-2023

HoD/Associate Dean: ________________________

Date: ________________________

Signature (Dean/Principal): ________________

Date: ________________________

Approval

It is certified that the contents and form of the thesis entitled "Enhancing Intrusion Detection: Leveraging Federated Learning and Hybrid Deep Learning Models" submitted by FAIZA NAEEM have been found satisfactory for the requirement of the degree

Advisor : Dr. Safdar Abbas Khan

Signature: ______________________

Date: 03-Aug-2023

Committee Member 1: Dr. Muhammad Khuram Shahzad

Signature: Khurram.

03-Aug-2023

Committee Member 2:Dr Farzana Jabeen

Signature: Farzana

Date: 04-Aug-2023

Signature: ______________________

Date: ______________________

Dedication

This thesis is dedicated to *my beloved Mamu* ***Mr Zia ul Haq*** *and Mumani* ***Ms Rashda Parveen****.*

I want to express my gratitude towards them for their love, support and motivation.

Certificate of Originality

I hereby declare that this submission titled "Enhancing Intrusion Detection: Leveraging Federated Learning and Hybrid Deep Learning Models" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: FAIZA NAEEM

Student Signature: ______________

Acknowledgments

All praises are for Almighty Allah, the creator of the universe. I thank Him for all of His blessings at every step of my life.

I am very thankful to my supervisor, **Dr Safdar Abbas**, especially for his helpful guidance, continual support, patience and tolerance during my study. I would like to thank and express gratitude to **Dr Asad Waqar Malik** for his fruitful knowledge and experience have always motivated and emboldened me all the time in my research work and academic studies. In accompany with that, I am also grateful to **Dr Farzana Jabeen** and **Dr Khuram Shahzad** for their kind assistance and support, which have made my complete work and life magnificent at NUST.

I would also be grateful to my parents as without their support and prayers it would be impossible to complete the work during the time frame. Their extreme encouragements always motivate me in life.

Contents

List of Abbreviations and Symbols

Abbreviations

FL	Federated Learning
QoS	Quality Service
IoT	Internet of Things
IIoT)	Industrial Internet of Things
FedAvg	Federated Average
SGD	Stochastic Gradient Descent
RF	Random Forest
DT	Decision Tree
TPR	True Positive Rate
FPR	False Positive Rate
Non-IID	Non-Identical Independent
SVM	Support Vector Machine

List of Tables

List of Figures

Abstract

An efficient and productive IoT application may ease some real-time tasks however, it is at risk of cyber-attack. Intrusion Detection Systems (IDS) are of significant importance for the security measures of IoT applications. Anomaly-based intrusion detection systems perform more efficaciously than other methods. IoT/IIoT devices that deal with large data volumes are at risk of malicious attacks and as a result, anomaly-based IDS are developed. But, the question that arises is whether the performance of models meets the required standards and accuracy. For research the Telemetry data of IoT/IIoT services from the ToN_IoT dataset collected at UNSW Canberra Cyber IoT lab, SEIT (Australia), is used. It includes data about seven IoT/IIoT sensors. Federated Learning based on Deep auto-encoder is adopted to efficiently identifying attacks while solvingthe issue of data leakage and the privacy of users . Federated models handle the non-IID data efficiently. Hybrid models use Machines and Deep Learning algorithms for efficient model design with increased detection rates. The algorithms used for the Hybrid model are Random Forest, Decision Tree and XGBoost. The XGBoost algorithm improves the accuracy of the Hybrid model with better predictions. Both Federated and Hybrid model ensures efficient pre-processing and feature selection. The results of the Federated model are dependent on device datasets while the Hybrid model outperforms on the same data.

Keywords: *Internet of Things (IoT), Industrial Internet of Things (IIoT), cyber-security, intrusion detection systems (IDSs),Machine Learning, Federated Learning, XGBoost*

CHAPTER 1

Introduction

In the advanced era of development, IoT devices are widely used to improve different tasks and aid humans in daily practices. These wireless IOT devices work with a connection to the internet and are managed remotely for real-time decision-making and providing information dissemination.[1] These IoT devices are helpful in daily life in every domain such as industrial control applications, health care and for environment monitoring like weather updates. [2] Every device senses its environment and records events and enables communication between machines without human intervention. [3] The resources of IoT sensors are limited and they are vulnerable to security attacks which is a major challenge in the current era. [4] IoT devices are vulnerable to different types of attacks and the reasons for them are viruses, malicious software and hackers. These attacks aim to breach privacy and data integrity.[5]As a result security measures requires more attention and for the stated purpose Intrusion Detection Systems are designed to examine data and identification of anomalous attacks.

The intrusion detection systems monitor the network traffic and scan it for malicious activities continuously and record it for further processing and administration. The recorded data is analyzed using signs and patterns for abnormal behavior. Then it is compared with some predefined rules for identifying the attack and notifying the administrator to take required preventions. Thus IDS prevents IoT devices from unauthorized access. Intrusion detection systems belongs to cryptography domains but require machine learning for their working as high cryptographic computational capabilities ar not applicable on Iot/IioT dataset.

Intrusion detection systems work using two methods i.e. signature-based model and anomaly-based model. In the signature-based method, the IDS detects malicious activities based on patterns of bytes in the network traffic.[6] These IDS also uses the sequences that are known before hands for intrusion detection which are called signatures. Using this system the signatures which are already in the system can be easily detected but detection of new anomalous activities is very difficult in it.

Therefore, anomaly-based intrusion detection systemswere introduced for unknown malicious attack detection. Machine learning models are used for anomaly-based attack detection. All the data is compared with the model and if it is not found then it is identified as attacked one. Anomaly-based IDS are better than signature-based because they are trained and designed in accordance with application and hardware.[7]

In general traditional IT-based system uses different tools like firewalls and cryptographic solution but IoT and IioT devices require comprehensive security measures because of their lightweight communication protocols and low storage capacity. For these devices, it is required to design specific intrusion detection systems in case of cyber attacks.[8]

The research thesis is based on an anomaly-based IDS. The dataset is classified into normal and anomalous one and then the unusual pattern is identified as anomalous. In recent research, different anomaly-based systems are designed to improve the efficiency of systems. The principle of all anomaly-based intrusion detection systems are:
i) defining parameters,
ii) training of system on given data and
iii) detecting anomalies.

The system is designed by defining the baseline behaviour of data for training purposes and then analyzing test data. Deep Auto-encoder and KitNET are famous anomaly-based intrusion detection algorithms. Both algorithms first extract features and then accurately identify anomalies from the dataset by training the dataset using an auto-encoder[9]. In research thesis for intrusion detection Federated Learning and Hybrid models are used for model development. Federated Learning is used in previous studies for intrusion detection in an efficient way. Federated learning plays an efficient role to support applications which are sensitive to privacy. Earlier Federated Learning is applied on various fields and some of its application are stated for background knowledge. Smartphones are used for learning user behavior using statistical model for applications for next word prediction. Users' privacy is a major concern as they are not willing to share their personal details. Other concern of users is to save their phones battery power and bandwidth. Federated learning enables predictions of words on mobile phones and different applications without affecting the user experience and preserving privacy [10].

Organizations like hospitals use federated learning to record patients' data to predict health care in accordance with preserving privacy of patients as they may have to face consequence legally and ethically. Federated learning is a solution for this type of problem where privacy policies are needed to be maintained while learning process [11]. Nowadays Internet of things: wearable devices, smart vehicles and homes have sensors for data collection and processing in real time. Autonomous vehicles require data real time traffic data which is difficult as it requires connectivity with each device. Federated Learning also help in this type of model training with efficiency to adapt change maintaining user privacy.

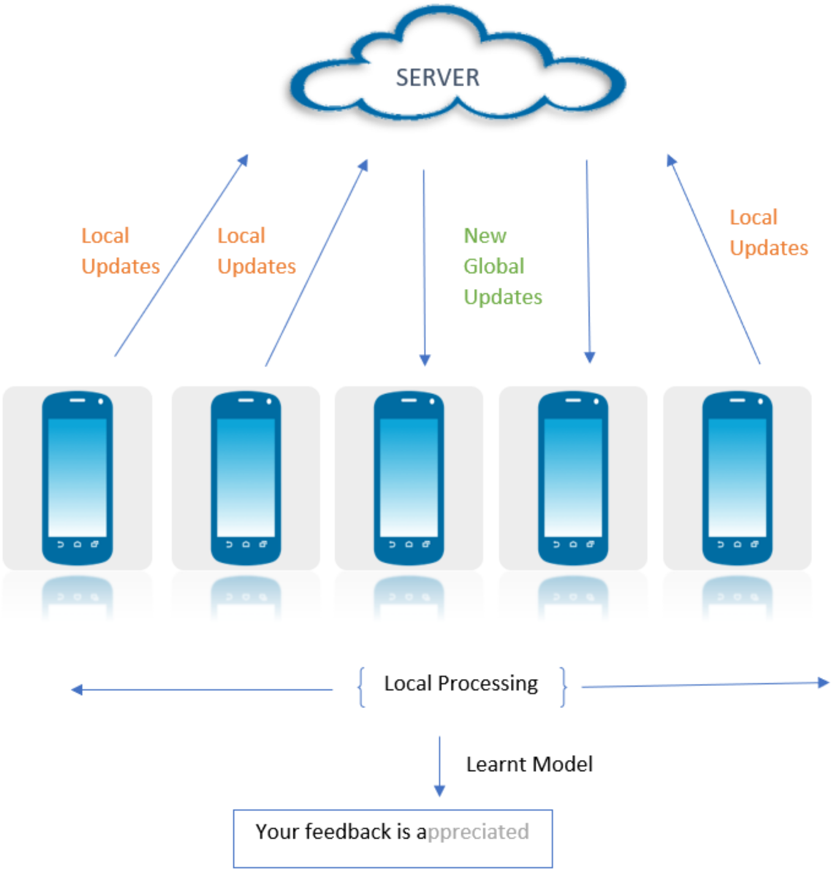


Figure 1.1: Federated Learning Text Predictive Model

Decisions Tree Random Forest and XGBoost are supervised machine learning algorithms. They are used for classification and segregation problems for regression or numerical output. Classification is a predictive modeling task where the goal of model is to assign input data values to predefined categories. Regression is another type of predictive modeling task where the aim of model is to predict a continuous numeric value or a quantity based on input features.

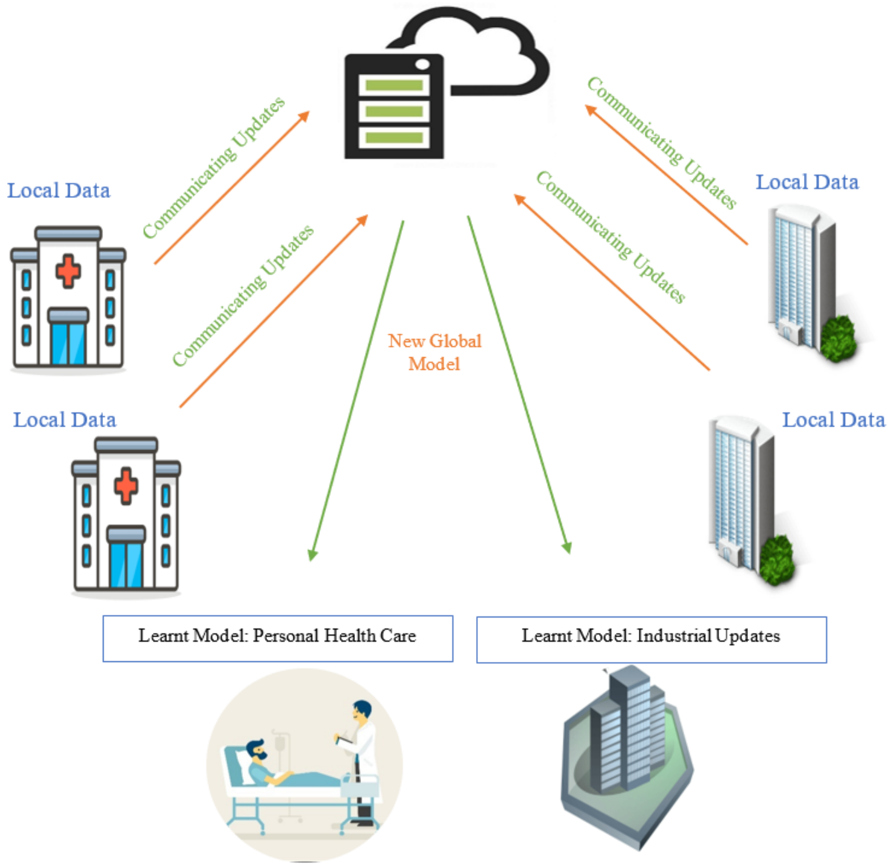


Figure 1.2: Federated Learning in Hospitals and Organizations

1.1 Research Problem

1.1.1 Problem statement

IoT devices are heterogeneous and relatively weak in terms of security, Machine Learning algorithms are widely used for intrusion detection. Federated learning is used with the advantage of a secured model. The main research problem is that how use of Federated learning and machine learning based Hybrid models enhance the performance of intrusion detection system. The designed models are evaluated and analyzed by accuracy, F1 score, precision and recall. Another question for the stated problem is that what are the limitations and challenges of both models over each other?

1.1.2 Proposed Solution

An architecture for the problem's solution is proposed, which implements Federated Learning based on Deep Auto-Encoder and Hybrid Model using Random Forest, Decision Tree and XGBoost. Hybrid approach is applied with efficient preprocessing and feature selection to get optimal IDS model while Federated model solves the issues of non-IID for real time usage.

1.1.3 Research Objective

Nowadays Internet of things: wearable devices, smart vehicles and homes have sensors for data collection and processing in real-time. These sensors reads data in real-time that is difficult to process as it requires connectivity with each device for its proper working. This real-time data is a source of collecting data in huge volume. Machine learning algorithms are designed to handle this type of data. Data collection is an easy task in comparison to data handling as many organizations and individuals are not comfortable sharing their private data because of privacy concerns, government and trade secrets. Privacy concerns hinder researchers to carry out their research as they do not have complete data access only due to data leakage.Intrusion detection systems (IDS) are essential for network usability. The challenges while developing IDS are large amounts of data and sophisticated attackers. In machine learning, Federated Learning is a method that makes sure data security by uploading local model parameters to different clients and building up a digital model. The study involves two parts one using the federated learning for IoT/IIoT devices using the Deep encoder model. The second part of the research involves the evaluation of the Dataset using Hybrid Models.

Chapter 2

Literature Review

In past, many researches have tried to resolve the issue of anomaly detection. Anomalies are major hindrance for experts to perform their tasks in the computer science and IT field. The behavior of anomalies are needed to be detected efficiently to take adequate measures. In previous years, different techniques and measures are proposed with efficacy to identify intrusions and prevent them. In our research, we discuss use of machine learning algorithms and federated learning based anomaly detection methods. Machine learning algorithms have high classification accuracy to detect intrusions so they have gained great momentum in recent years. However, federated learning preserves privacy of users and provide data security.

2.1 Application of Federated learning for IDS:

Federated learning is a decentralized technique to ensure security and privacy. The methodology trains the model locally and parameters are send to the the centralized server for aggregation and then server transfer back the aggregated parameters to iterate the training process [12]. The challenges of FL are false alarms, high latency and poisoning attacks. These challenges are examined for future research. The study also presents solutions for federated learning based challenges like IDS handling, FL heterogeneity and interoperability to build efficient models.

In research, federated learning based technique to ensure privacy is proposed with local training of model on IoT. The benefit of proposed study is that devices take advantage of aggregated results of others and in result the anomaly detection model is improved [13]. The dataset used in this study is NSL-KDD. The accuracy of centralized model that trained over the entire dataset is 83.09%. A comparison between FL based model and self-learning in which model updates are not shared is done. The results of FL model outperforms the self-learning model.

In the study of wireless edge networks and their intrusion detection, the Federated Learning-based model is designed using Attention Gated Recurrent Unit (FedAGRU). The FedAGRU updates on the universal learning method while the centralized learning model updates on raw data shared among edge devices and the central server. The proposed methodology efficiently reduces the communication overhead with an assurance of converged learning. The datasets used for research are KDD CUP 99, CICIDS2017 and the WSN-DS [14]. The results of the model show that the accuracy is increased by 8% with the use of FedAGRU than the centralized learning algorithms, and the model is also 70% cost-effective in comparison to different federated learning methodologies. The accuracy of the FedAGRU model on IID data is 99.28% and on non-IID data, it is 98.82%.

A multi-classifier approach is used for developing FL based IDS model to evaluate the CIC-ToN-IoT dataset [15]. To model the IDS system three different settings are adopted by dataset partitioning relative to IP address and malacious attack types. These data partitioning types are basic, balanced and mixed. Basic data partitioning contains data about the network traffic of each IoT device. Balanced data contains portions of data from each IoT device with the same number of samples while mixed data is a tradeoff of basic and balanced data in which a balanced number of samples are taken but each IoT device maintains them. The model aggregation is carried out using FedAvg and Fed+ aggregating functions and the IBMFL framework. The study also discusses the challenges of efficacious FL implementation. The results of Fed+ are better than FedAvg on mixed data or balanced data.

The research of Davy Preuveneers gives a solution to the challenge where an adversary may poison the ML models with malicious samples of training. For the solution of the problem, "a block-chain-based federated learning model" is designed to detect intrusions on the CICIDS2017 dataset with a chained distributor ledger [16]. The proposed methodology audits the models without centralized data training. The accuracy of the model according to results is 97% for both training and validation data. The study reveals that while the use of auto-encoder with federated learning, has an inauspicious impact on the performance of the model differs between 5-15% but it provides transparency in distributed training of the model. The model can be generalized and applied to similar use cases.

In research Liu et al. proposed an anomaly detection method in which training of given model is carried out using "FL framework and Attention Mechanism-based Convolutional Neural Network Long Short-Term Memory (AMCNN-LSTM)"[17]. The efficiency of the model is improved by implementing a gradient compression mechanism. Application of all ML techniques gives an accuracy of 92% on power demand, ECG, space shuttle and engine. The Root Mean Squared Error (RMSE) is comparatively lower than state-of-the-art methodologies: CNN-LSTM[18], Stacked Autoencoders [19], GRUs [20] and LSTM[21]. Lin et al., in their research on the dataset provided by Virustotal, implement a combination of FL with LSTM and SVM for malware classification and achieve an accuracy of 91.67% [22]. Mothukuri et al. in their research for intrusion detection system introduce FL with an ensembler to detect anomaly applying a decentralized model on the Modbus network dataset [23]. The accuracy achieved using implemented model is 90.25%.

In a study for anomaly detection in IoT devices, an autonomous FL-based self-learning distributed system named DIOT is presented by Nguyen et al.. The model also makes use of Gated Recurrent Units (GRU). The dataset was generated by collecting network activity in a lab using Kali Linux and Hostapd to develop a gateway for WiFi and Ethernet to connect IoT devices [24]. The model utilizes federated learning for efficient aggregation of behaviour profiles. The designed model is capable of identifying unknown attacks. The accuracy rate of the system is 95.6%.

Qin et al. in their study, develop anomaly-based IDS using real-time high-dimensional time series and NSL-KDD dataset [25]. The model is designed for resource-limited embedded systems by implementing "Sequential Extreme Learning Machine(OS-ELM)" to get "on-device sequential learning neural network (ONLAD)". For threat detection, OS-ELM is used with an auto-encoder. For the dimensions of the dataset greedy algorithm is used to select features. Federated averaging algorithms help in grouping similar target attacks but it has a disadvantage in that it only selects devices with the same data features while developing the global model. The designed model has many input and output layers according to selected features and 64 neurons as hidden layers. The accuracy achieved using feature selection is 70.4% and the worst is 25.7% when features are not selected.

The study [26] addresses the heterogeneity challenge of IoT devices for anomaly detection and uses federated learning on the LSTM model on simulated datasets of general electric current smart buildings . For model training, the federated averaging algorithm is used in iterations until completion of defined training rounds and the model is converged. The accuracy achieved by the proposed model is 90%.

A resource-efficient approach for anomaly detection in IoT is proposed by implementing Federated aggregation-based BIRCH K-means to develop site-invariant IoT µS models [27]. The dataset is collected by 7 different VSL service types. The behaviour of the model is evaluated on common IoT attacks. The accuracy achieved by the designed model is 99% on the test dataset.

Schneble et al. [28] implement Federated Learning for efficient communication and reduce the cost of the machine learning model. The MIMIC patient dataset is used for identification of attacks. The model achieves an accuracy of 99%. An advantage of research is that it handles the uneven distribution of data scaled up for mobile devices for security.

Zhao et al. [29] in their research for intrusion detection, proposed "a federated learning-based multi-task deep neural network(MT-DNN-FL)". Their research is on VPN traffic recognition and classification tasks on CICIDS2017, ISCXTor2016 and ISCXVPN2016 datasets. The achieved accuracy is 97.97% which is better than the centralized training architecture. In a study on generated IIoT device dataset for intrusion detection a blockchain-based federated learning approach is proposed by Zhang, Lu, et al.[30]. For model designing "centroid distance weighted federated averaging (CDW FedAvg)" is used to resolve data heterogeneity issues and distinguish positive and negative classes in the dataset. The proposed approach is efficient and feasible with an accuracy of 89%.

Authors in their research applied "Convolutional Neural Networks (CNN) along with federated learning" to develop efficient intrusion detection systems. Fan et al. [31]make use of Federated parameter aggregation on CICIDS2017, NSL-KDD and self-generated datasets. The proposed model achieves an accuracy of 91%. Sun, Ochiai, et al. [32]also develop a model using the same ML techniques on the network dataset LAN-Security Monitoring Project with an accuracy of 87.10%. Li, Zhou, et al. [33] model based on CNN and federated Homomorphic parameter addition has an accuracy of 81% on self-generated terrestrial and satellite network datasets.

In the classification-based research of Al-Marri et al. [34], Artificial Neural Network is applied using a federated averaging algorithm on the NSL-KDD dataset and attains an accuracy of 98.12%. Popoola et al. [35] secures an accuracy of 99.39% by implementing the same ML algorithms to evaluate Bot-IoT and N-BaIoT datasets.

An architecture is proposed for network intrusion detection using unsupervised stacked federated learning. The Federated learning flower framework evaluates the network datasets: Bot-IoT, TON-IoT, UNSW-NB15 and CSE-CIC-IDS-2018 [36]. The results of the model are remarkable on non-IID data. The model reaches an accuracy of 93% on Bot-IoT, 74% on TON-IoT, 97% on UNSW-NB15 and 98% on CSE-CIC-IDS-2018. With the help of research, federated learning is considered a promising approach to generalising diverse networks.

2.2 Hybrid Approach for IDS:

IoT systems capture real-time data continuously to process it and transfer information to a global server. Hackers are ready to spot weak systems and attack them. So, intrusion detection systems are vital for identification and security of attacks. A model is designed by Souza et al. [37]using a fog computation layer to identify data as attacked or benign to take countermeasures. An efficient Hybrid classification approach using DNN-kNN is implemented to design an IDS system. The model is applied to NSL-KDD and CICIDS2017 datasets for evaluation. The selection of attributes from the dataset is carried out using the rate of information gain. The designed model has less operational cost and memory usage with an accuracy of 99.77% on NSL-KDD and 99.85% on CICIDS2017 datasets.

In research, Rashid et al. [38]designed an anomaly-based intrusion detection system by incorporating self-learning classification algorithms along with feature selection and ranking. The ML algorithms used for Hybrid architecture design are Neural Network, DNN, SVM, Naïve Bayes along with Deep Auto-encoder. To evaluate the designed model, authors selects NSL-KDD and CIDDS-001 datasets. To evaluate model results, performance metrics: Accuracy, F1 Score, Precision and Recall are used. k-NN, SVM, NN and DNN algorithms are applied on the NSL-KDD dataset and attain 100% accuracy and CIDDS-001 has approximately 99% accuracy rate by applying k-NN and Naïve Bayes algorithms.

In research by Ding et al. [39], three datasets are used for evaluating their network intrusion system. The research focus is on balancing sample data using KNN for under-sampling in an efficient way. TACGAN- IDS framework is applied for iterative training of data. TACGAN is derived from ACGAN and has properties of three types of networks which are conditional generative adversarial network (CGAN), semi-supervised generative adversarial network (SGAN) and information maximizing generative adversarial network (infoGAN). The datasets used for training and testing of the model are KDDCUP99, CICIDS2017 and UNSW-NB15 datasets. The model achieves 93.53% accuracy on the KDDCUP99 while accuracy score on CICIDS2017 is 95.86% and the UNSW-NB15 dataset has an accuracy of 92.39%.

Machine learning algorithms are applied mostly for intrusion detection but in the study, Parsaei et al. [40] applied a data mining approach to the NSL-KDD dataset. They work to detect R2L and U2R classes from imbalanced data using the Hybrid model by applying the "synthetic minority oversampling technique (SMOTE) and cluster centre and nearest neighbour (CANN)". The accuracy of the designed model for detecting low-frequency attack U2R is improved by 94% and R2L by 50%. In base research by Lin et al. [41] the accuracy achieved using combining Cluster Centers and Nearest Neighbors (CANN) on KDD-Cup 99 dataset for U2R is 28.7% and for R2L 61.92%.

Gautam et al.[42]proposed a Hybrid model using the CICIDS2017 dataset. The model architecture is based on "Bidirectional Recurrent Neural Network using Long Short-Term Memory and Gated Recurrent Unit". The results of the model predict attacks with an accuracy score of 99.13%. The performance of the model is improved because instructions flow in bi-direction and concatination is applied to combine sequence.

Hybrid learning for intrusion detection is considered an innovative perspective. In research by Emec et al. [43], a BGH Hybrid model is constructed and compared with BLSTM-GRU algorithms. The evaluation of the model is carried out using CIC-IDS-2018 and BoT-IoT datasets to detect intrusions. To use the datasets comprehensive feature selection is needed and for it, seven ML algorithms are used to get two types of feature variants: Full and ten best features. The BGH model outperforms the given datasets with an accuracy of 98.78% on CIC-IDS-2018 and 99.99% on BoT-IoT datasets.

Research by Ethala et al. [44] proposed a new model by combining "Spider Monkey Optimization (SMO) and Hierarchical Particle Swarm Optimization (HPSO)". The dataset used for intrusion detection is NSL-KDD and UNSW-NB 15 datasets. Optimal and important feature selection is carried out by using the Rosenbrock function. The score of accuracy using these datasets are 99.175% and 99.18% respectively.

A hybrid intrusion detection system is designed using "Support Vector Machine (SVM) and combined with Adaptive Neuro-Fuzzy System (ANFIS)" to deal with imprecise information [45]. The fuzzy model is like ANN and detects R2U, U2R, and DDOS attacks. 99.3% accuracy is achieved after applying Fine Gaussian SVM (FGSVM) algorithm on projected NSLKDD dataset.

Jadhav et al.[46], in their research on Hybrid modelling of anomaly-based IDS, uses KDDCUP99 and NSLKDD datasets. The architecture of the model is designed by applying RNN-LSTM machine learning algorithms which outperform other algorithms SVM, RF, J48 and Naive Bayes. The accuracy result of the model on the projected dataset is 96%. From the beginning of intrusion detection to the current research many efficient IDS are available but, there is always room available to improve them. Recurrent neural networks (RNNs) based Hybrid model is used to evaluate on IoT-23 dataset and the UNSW-NB15 dataset [47]. Optimal features from the dataset are selected using Harris Hawk optimization. LSTM and GRU algorithms are used to detect attacks from the dataset along with RNN. Testing of modelshow that its accuracy on the IoT-23 and UNSW-NB15 datasets are 98.12% and 99.98% respectively. Hence, from accuracy score of the model on both projected datasets reveals its efficacious behaviour. The proposed architecture of IDS is capable of detecting known and unknown attacks.

Machine Learning algorithms Logistic Regression (LR), KNN, Random Forest (RF), Linear Discriminant Analysis (LDA), Classification and Regression Tree (CART), SVM AND LSTM are projected on the ToN_IoT dataset for evaluation. The results of Machine Learning algorithms on the ToN_IoT dataset are shown in the table below depicting minimum and maximum accuracy [48].

Dataset	Minimun Accuracy	Maximum Accuracy
Garage Door	-	100% Accuracy of all algorithms
Weather	LR 58%	CART 87%
Motion Light	KNN 54%	LSTM 59%
Thermostat	CART 59%	Others 66%
	KNN 60%	
GPS Tracker	CART and NB 84%	KNN 88%
Modbus	NB, SVM, LR, LDA 67%	CART 98%
Fridge	NB 50%	LSTM 100%

Table 2.1: Performance summary of ML Algorithms on ToN_IoT Dataset

The minimum accuracy of binary classification on the combined dataset when evaluated using Logistic Regression (LR) and SVM is 61% and its maximum is 88% by using CART. The minimum accuracy of multi-class classification on a combined dataset when evaluated using NB is 54% and its maximum is 77% by using (CART).

Sr No	Authers	Year	Dataset	Algorithm	Accuracy
1	[13]	2020	NSL-KDD	FL	77.79%
2	[14]	2020	KDD CUP 99, CICIDS2017 and WSN-DS	FedAGRU	99.28%
3	[15]	2022	CIC-ToN-IoT	IBMFL	Above 80%
4	[16]	2018	CICIDS2017	Centralized Auto-encoder FL based Auto-encoder and FL based Blockchain	97% vary 5-15%
5	[17]	2021	Power demand, ECG, space shuttle and engine	FL based AMCNN-LSTM	92%
6	[22]	2020	Virustotal'S dataset	FL with LSTM and SVM	91.67%
7	[23]	2021	Modbus Network Dataset	FL with an ensembler	90.25%
8	[24]	2019	Network Traffic	GRU and FL	95.6%
9	[25]	2021	NSL-KDD	FL based OS-ELM	70.4%
10	[26]	2021	electric current smart buildings	FL on LSTM	90%

Table 2.2: Performance summary of proposed works in literature review

Sr No	Authers	Year	Dataset	Algorithm	Accuracy
11	[27]	2018	Generated	FL based BIRCH K-means	99%
12	[28]	2019	MIMIC	FL	99%
13	[29]	2020	CICIDS2017 ISCXTor2016 and ISCXVPN2016	FL based DNN	97.97%
14	[30]	2020	Generated	CDW FedAvg	89%
15	[31]	2020	CICIDS2017, NSL-KDD and Generated	CNN and FL parameter aggregation	91%
16	[32]	2020	LAN-Security Monitoring Project	CNN and FL parameter aggregation	87.10%
17	[33]	2020	Generated	CNN and Homomorphic parameter addition	81%
18	[34]	2020	NSL-KDD	FL based ANN	99.12%
19	[35]	2020	Bot-IoT and N-BaIoT	FL based ANN	99.39%
20	[36]	2023	Bot-IoT TON-IoT UNSW-NB15 CSE-CIC-IDS-2018	FL Flower Framework	93% 74% 97% 98%

Sr No	Authers	Year	Dataset	Algorithm	Accuracy
21	[37]	2020	NSL-KDD CICIDS2017	DNN-kNN	99.77% 99.85%
22	[38]	2020	NSL-KDD CIDDS-001	k-NN, SVM, NN,DNN NB and AutoEncoder	100% 99%
23	[39]	2022	KDDCUP99 CICIDS2017 UNSW-NB15	TACGAN-IDS framework	93.53% 95.86% 92.39%
24	[40]	2016	NSL-KDD to identify U2R and R2L	SMOTE and CANN	U2R 94% R2L 50%
25	[42]	2022	CICIDS2017	Bidirectional RNN (LSTM + GRU)	99.13%
26	[43]	2022	CIC-IDS-2018 BoT-IoT	BGH Model using BLSTM-GRU	98.78% 99.99%
27	[44]	2022	NSL-KDD UNSW-NB 15	SMO and HPSO	99.175% 99.18%
28	[45]	2022	NSL-KDD	FGSVM and ANFIS	99.3%
29	[46]	2023	NSL-KDD	RNN-LSTM	96%
30	[47]	2023	UNSW-NB15	RNN based LSTM-GRU	99.98%

Chapter 3

System Architecture

3.1 Dataset Description

The TON_IoT dataset is comprised of Internet of Things (IoT) and Industrial IoT (IIoT) datasets to evaluate the efficiency and accuracy of cybersecurity-based IoT devices using Artificial Intelligence. The ToN_IoT dataset is based on data collected from different heterogeneous data sources. This dataset includes Internet of Things (IoT) and Industrial Internet of Things (IIoT) Telemetry datasets, Network Traffic datasets, Windows datasets and Linux Datasets. The IoT and IIoT dataset from main ToN_IoT contains data about IoT devices; Fridge, Garage Door, Weather, GPS Tracker, Modbus, Motion Light and Thermostat. The ToN_IoT Linux dataset is about Ubuntu 14 and 18 TLS. The Windows dataset is about Windows 7 and 10. The source of data collection was a large-scale realistic network at UNSW Canberra Cyber IoT lab in the School of Engineering and Information Technology (SEIT). The data about normal and cyber attacks were collected from the network using parallel processing and the testbed was developed by connecting physical systems, fog and cloud platforms, virtual machines and hacking applications. IoT and IoT sensors were also used for industrial IoT in a testbed to impersonate the complexity and scalability. For research on defined architectures Processed and Train-Test IoT and IIoT devices, dataset is selected.

DoS, DDoS and ransomware data hacking techniques were used for computers, web applications and IoT gateways for normal and anomalous data collection.

3.2 Imprtance of ToN_IoT Dataset

In comparison to previous datasets, the sensors readings of IoT are not included in it like in UNSW-IoT and Bot-IoT. But, the ToN_IoT dataset has the sensors readings. It includes both normal and anomalous data. The architecture of testbed is real with IoT communicating layers i.e. Edge, Fog and Cloud layers. The Testbed includes hetrogeneous data sources.

3.3 Processed Dataset

The data collected from different sensors is processed and saved in CSV format. Data is processed to filter and convert it into standard features and labels. The processed data is considerably large and its total count is given below.

Dataset	Total Records Count
Garage Door	571205
Weather	650238
Motion Light	452263
Thermostat	442229
GPS Tracker	595687
Modbus	287195
Fridge	587077

Table 3.1: Record Count of Processed Dataset.

3.4 Train_Test_datasets

Train_Test dataset of Iot and IIoT stores information in CSV format. This dataset is used to evaluate accuracy and efficiency of Cybersecurity based IoT devices using Artificial Intelligence. Listed below is the count of data records.

Dataset	Total Records Count
Garage Door	59588
Weather	59261
Motion Light	59489
Thermostat	52775
GPS Tracker	58961
Modbus	51107
Fridge	59945

Table 3.2: Record Count of Train_Test_Dataset

3.5 Types of Attacks in Dataset

3.5.1 DoS/DDoS Attacks:

Daniel of the system (DoS) and Distributed Daniel of the system (DDoS) is a system flooding in which network and service overcome enormous traffic access, making it unavailable to the authorized user and diverting their attention from other malicious activities[49]. It is launched using bots and botnets. The storage and memory capacity of IoT/IIoT devices is very low so these devices become easy and weak victims of DDoS attacks. For dataset generation, IoT/IIoT devices are attacked by using a Python script coded with the help of the Scapy package and UFONet toolkit. Different IP addresses of offensive Kali Linux are used for the purpose and are listed below:

1. For DoS:

a) 192.168.1.30

b) 192.168.1.31

c) 192.168.1.39

2. For DDoS:

a) 192.168.1.30

b) 192.168.1.31

c) 192.168.1.34

d) 192.168.1.35

e) 192.168.1.36

f) 192.168.1.37

g) 192.168.1.38

3.5.2 Password attack:

They are types of intrusions which intervene and focus on attaining unauthorized access to a system, network, or account of IoT/IIoT device through weaknesses in password utilization[50]. The IoT/IIoT devices are attacked by dictionary attacks and brute force attacks using the IP method Common password attack is Brute and phishing force attack. Bash scripts for attacks are coded using the CeWL toolkit and the Hydra toolkit and Hydra toolkit for dictionary and brute force attacks respectively. The IP addresses for Password cracking using offensive Kali Linux are listed below:

a) 192.168.1.30

b) 192.168.1.31

c) 192.168.1.3

d) 192.168.1.35 and

e) 192.168.1.38

3.5.3 Ransomware Attack:

A ransomware attack is a type of cyber attack in which an attacker infiltrates a system or device or network and encrypts the data, making it inaccessible. Then ransomware is adopted by an attacker, generally in cryptocurrency, which gives the key to decode encrypted files. Ransomware attacks have become soared, which have a significant threat to businesses, individuals, organizations and for smart devices. Ransomware detection and prevention is critical to save the organization from data loss, financial detriment and potential operational intrusion[51][52]. In the case of IoT/IIoT ransomware deny access. For dataset generation, the Metasploitable3 framework is used to attack the designed system as a ransomware attack. The IP addresses for ransomware attacks using offensive Kali Linux are listed below:
a) 192.168.1.33
b) 192.168.1.37
IDS intrusion detection systems can play a major role for IoT/IIoT in diagnosing ransomware attacks early and respond timely

3.5.4 Injection Attacks:

These are types of cyber attacks in which the involvement of malicious and vulnerable data or code is inserted into an application, database or smart device[53]. These attacks utilize software vulnerabilities that allow input of the user to be executed without any proper validation with serious harmful results. Injection attacks can pose unauthorized access, theft of data, manipulation of data and even whole system compromise. For telemetry data bash codes are written using vulnerable public PHP and DVWA web applications. The IP addresses for an attack using offensive Kali Linux are listed below:
a) 192.168.1.30
b) 192.168.1.31
c) 192.168.1.33

d) 192.168.1.35

e) 192.168.1.36

f) 192.168.1.38

3.5.5 Cross-Site Scripting (XSS):

It is a type of vulnerability of web security that occurs when a malicious script (usually Java Script) is inserted by an attacker into web pages viewed by other users[53]. When an application fails to validate properly data provided by the user before rendering it back to other users can lead to Cross-site scripting XSS. Attacks of XSS are considered as most common web application security threats and can have serious consequences which includes data theft, hijacking of session and unauthorized access to user account. HTML and Javascript codes are used to inject malicious data on web pages for authentication between the web server and IoT device. The IP addresses for an attack using offensive Kali Linux are listed below:

a) 192.168.1.32

b) 192.168.1.35

c) 192.168.1.36

d) 192.168.1.39

3.5.6 Backdoor Attack:

A backdoor attack is a type of threat to cyber security in which an unauthorized and hidden access point is intentionally inserted by an attacker in a computer system, network and application[53]. The main theme of this attack is to grant normal authentication and security mechanisms, which allows the attacker to achieve unauthorized access or control over a system without being diagnosed easily. For dataset generation, the Metasploitable3 framework[] is used to attack the designed system as a ransomware attack. The IP addresses for ransomware attacks using offensive Kali Linux are listed below:

a) 192.168.1.33

b) 192.168.1.37

3.5.7 Scanning Attack

A scanning attack is a cyber security attack in which systematic scanning of a network, system or application is carried out by an attacker for vulnerabilities, weaknesses or for open ports that can be used to attain unauthorized access or incept further attacks. The first step in the cyber attack process is scanning attacks which allow attackers to recognize potential targets and weaknesses before initiation of more targeted and focused attacks Scanning attacks can be authorized and malicious both, scanning activities which are performed by authorized professionals to recognize vulnerabilities fixing before exploitation by malicious attackers often include ethical hacking or penetration testing.

3.6 Dataset Statistics

3.6.1 IoT Garage Door

3.6.1.1 Features of IoT Garage Door

1. date: Date

It is the logging Date of IoT telemetry data.

2. time: Time

It is the time logging of IoT telemetry data.

3. door_state: Boolean

This feature represent the state of door which is linked with designed network using a sensor to check either the door is open or closed.

4. sphone_signal: Boolean

This feature represent the signals received on phone which are either true or false.

5. label: Number

It is a tag for identifying normal and attack records. For identification 0 is for normal and 1 specify attacked record.

6. type: String

The "type" feature enlists the tags of attacks. Attacks are divided into categories, like normal, DoS, DDoS, password and backdoor attacks.

Listed Below is the count of attacks on the IoT Garage Door dataset.

Labels	Train-Test Dataset	Processed Dataset
normal	35000	495201
ddos	5000	35568
backdoor	5000	19287
injection	5000	10230
password	5000	6331
ransomware	2902	2902
xss	1156	1156
Scanning	529	529

Table 3.3: Statistics of IoT Garage Door

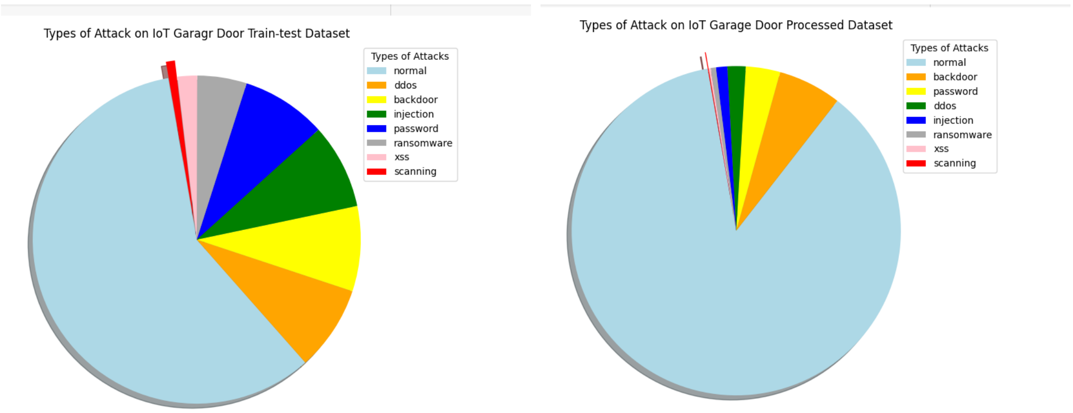


Figure 3.1: Statistics of IoT Garage Door

Important Features of IoT Garage Door Dataset

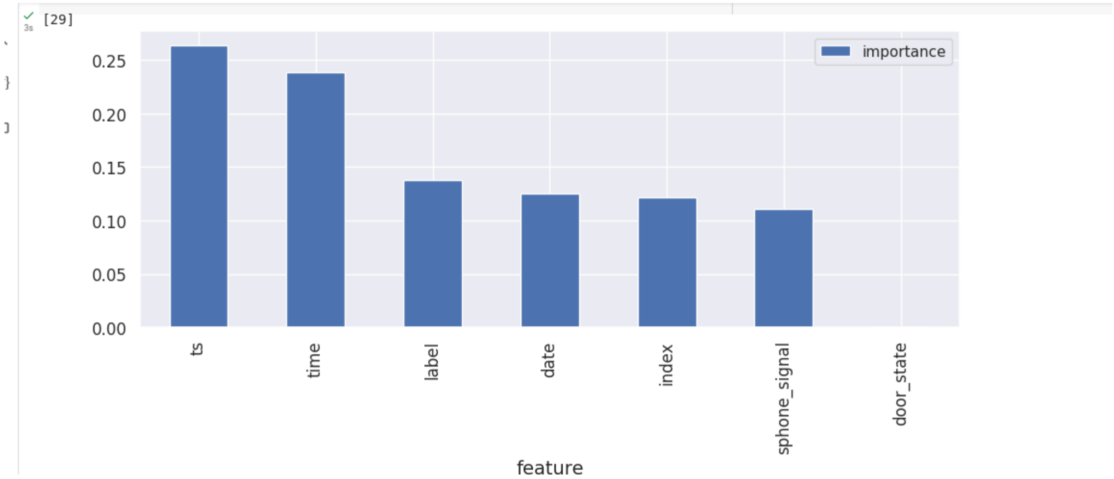


Figure 3.2: Important Features of Garage Door

3.6.2 IoT Weather

3.6.2.1 Features of IoT Weather

1. date: Date

It is the logging Date of IoT telemetry data.

2. time: Time

It is the time logging of IoT telemetry data.

3. temperature: Number

This feature is numeric recording of weather temperature using sensor connected with the network.

4. pressure: Number This feature is reading pressure value of weather using sensor connected with the network.

5. humidity: Number This feature has data about humidity of weather recorded using sensor connected with the network. Humidity readings of a weather sensor linked to the network

6. label: Number

It is a tag for identifying normal and attack records. For identification 0 is for normal and 1 specify attacked record.

7. type: String The "type" feature enlists the tags of attacks. Attacks are divided into categories, like normal, DoS, DDoS, password and backdoor attacks.

Listed Below is the count of attacks on the IoT Weather dataset.

Labels	Train-Test Dataset	Processed Dataset
normal	35000	559718
ddos	5000	35641
backdoor	5000	25715
injection	5000	15182
password	5000	9726
ransomware	2865	2865
xss	866	866
Scanning	529	529

Table 3.4: Statistics of IoT Weather

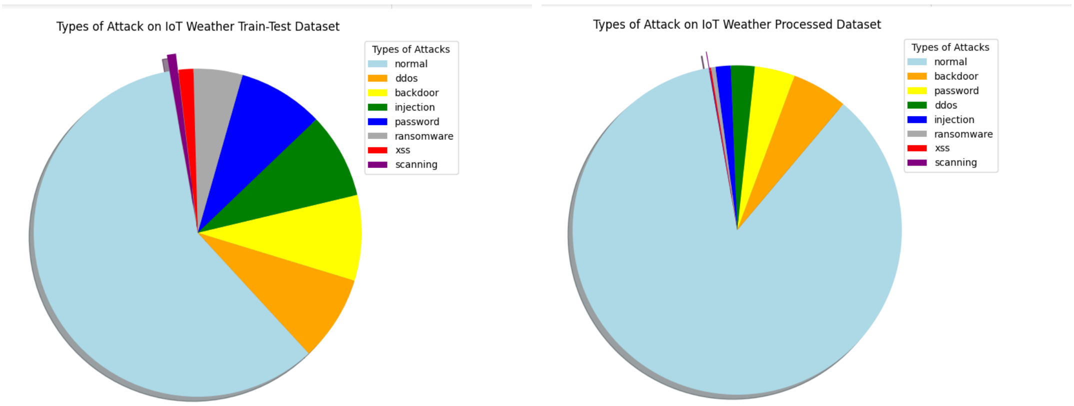


Figure 3.3: Statistics of IoT Weather

Important Features of IoT Weather

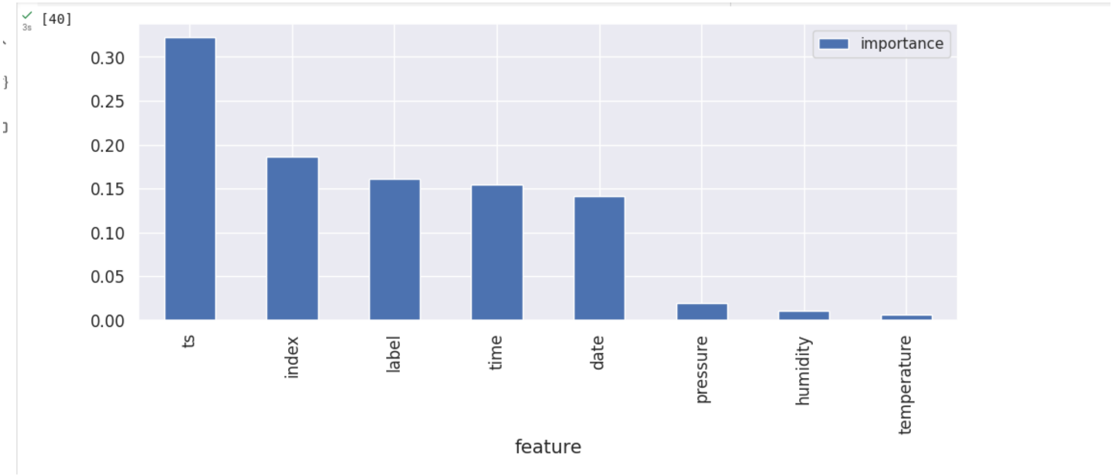


Figure 3.4: Important Features of Weather

3.6.3 IoT Motion Light

1. date: Date

It is the logging Date of IoT telemetry data.

2. time: Time

It is the time logging of IoT telemetry data.

3. motion_status: Number

This feature represents the motion sensor status which is either on (1) or off (0).

4. light_status: Boolean

This feature represents the motion sensor status which is either on or off.

5 label: Number

It is a tag for identifying normal and attack records. For identification 0 is for normal and 1 specify attacked record.

6. type: String

The "type" feature enlists the tags of attacks. Attacks are divided into categories, like normal, DoS, DDoS, password and backdoor attacks.

Listed Below is the count of attacks on the IoT Motion Light dataset.

Labels	Train-Test Dataset	Processed Dataset
normal	35000	388328
ddos	5000	8121
backdoor	5000	28209
injection	5000	5595
password	5000	17521
ransomware	2264	2264
xss	449	449
Scanning	1775	1775

Table 3.5: Statistics of IoT Motion Light

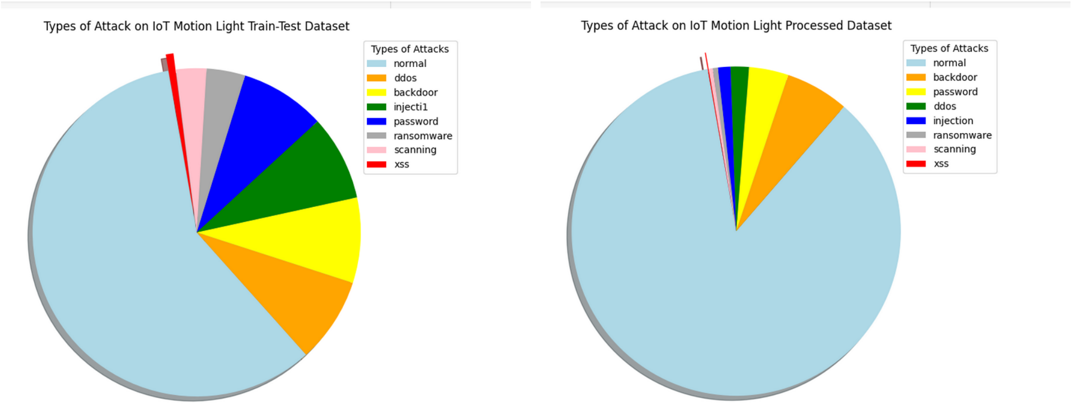


Figure 3.5: Statistics of IoT Motion Light

Important Features of IoT Motion Light

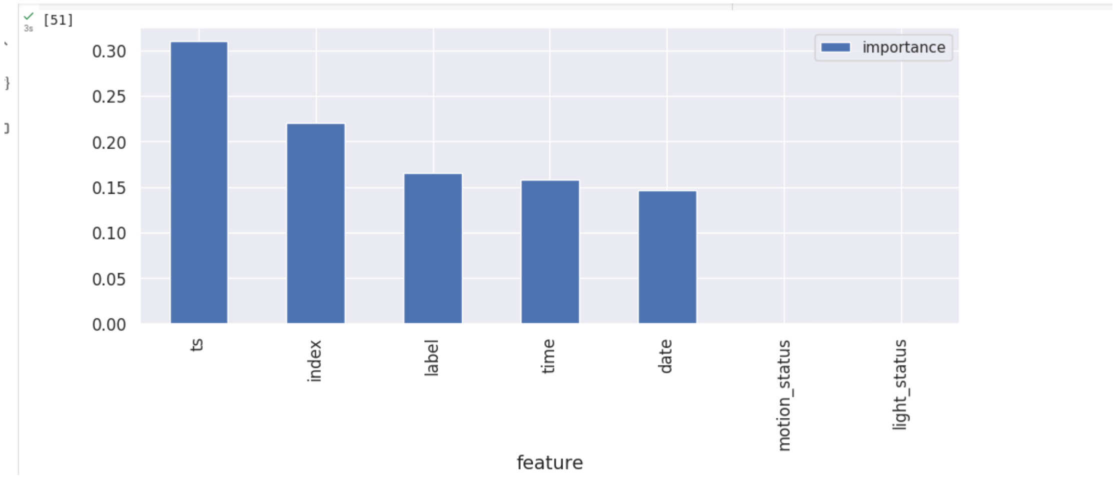


Figure 3.6: Important Features of Motion Light

3.6.4 IoT Thermostat

3.6.4.1 Features of IoT Thermostat

1. date: Date

It is the logging Date of IoT telemetry data.

2. time: Time

It is the time logging of IoT telemetry data.

3. current_temperature: Number

This feature is for reading sensor current temperature of thermostat connected with the network.

4.thermostat_status: Boolean

This feature represents the thermostat sensor status which is either on or off.

5. label: Number

It is a tag for identifying normal and attack records. For identification 0 is for normal and 1 specify attacked record.

6 type: String The "type" feature enlists the tags of attacks. Attacks are divided into categories, like normal, DoS, DDoS, password and backdoor attacks.

Listed Below is the count of attacks on the IoT Thermostat dataset.

LABELS	TRAIN-TEST DATASET	PROCESSED DATASET
normal	35000	385953
ddos	-	-
backdoor	5000	35568
injection	5000	9498
password	5000	8435
ransomware	2264	2264
xss	449	449
Scanning	61	61

Table 3.6: Statistics of IoT Thermostat

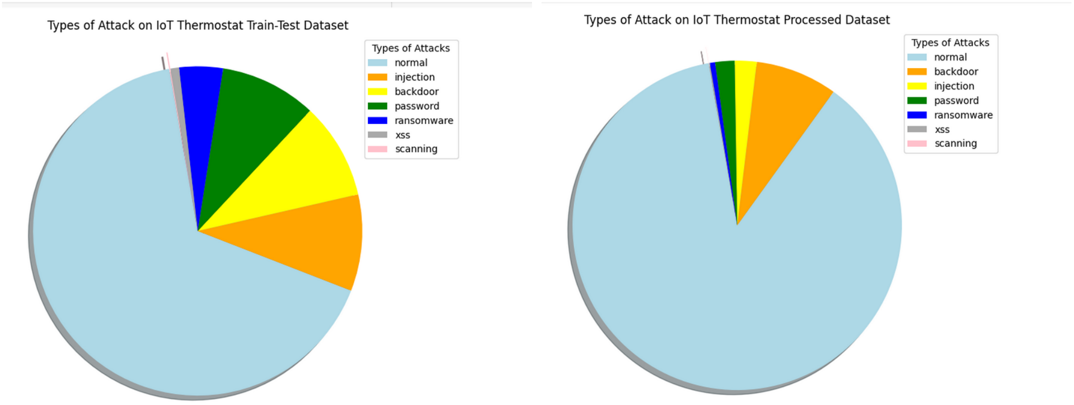


Figure 3.7: Statistics of IoT Thermostat

Important Features of IoT Thermostat

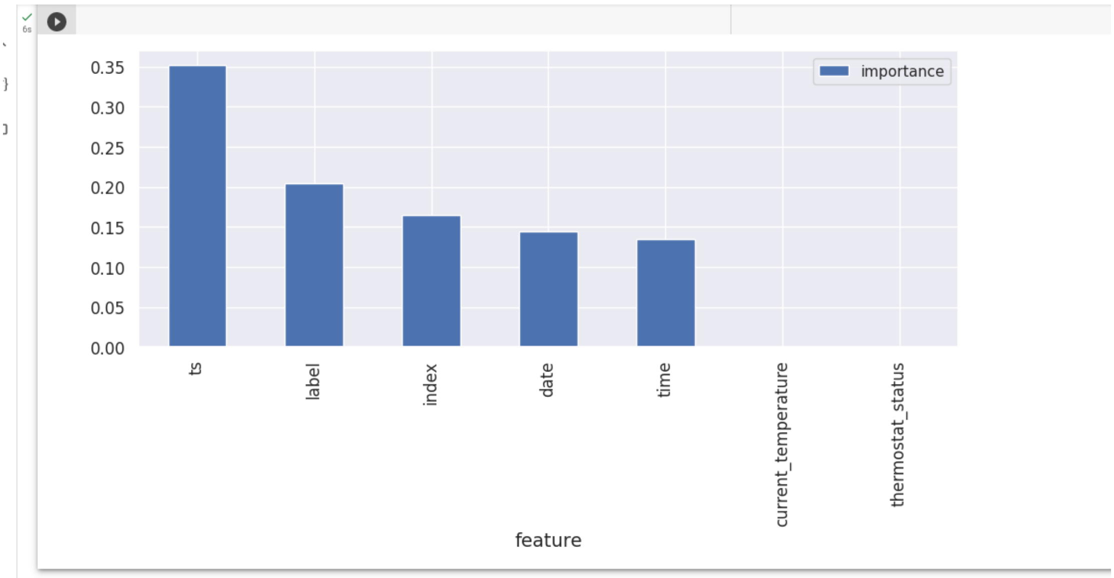


Figure 3.8: Important Features of Thermostat

3.6.5 IoT GPS Tracker

3.6.5.1 Features of IoT GPS Tracker

1. date: Date

It is the logging Date of IoT telemetry data.

2. time: Time

It is the time logging of IoT telemetry data.

3. latitude: Number

It is the measured Latitude value of GPS sensor attached with network.

4. longitude: Number

It is the measured Longitude value of GPS sensor attached with network.

5. label:

It is a tag for identifying normal and attack records. For identification 0 is for normal and 1 specify attacked record.

6. type: String

The "type" feature enlists the tags of attacks. Attacks are divided into categories, like normal, DoS, DDoS, password and backdoor attacks. Listed Below is the count of attacks on the IoT GPS Tracker dataset.

Labels	Train-Test Dataset	Processed Dataset
Normal	35000	513849
DDos	5000	10226
Backdoor	5000	35571
Injection	5000	6904
Password	5000	25176
Ransomware	2833	2833
XXS	577	577
Scanning	550	550

Table 3.7: Statistics of IoT GPS Tracker

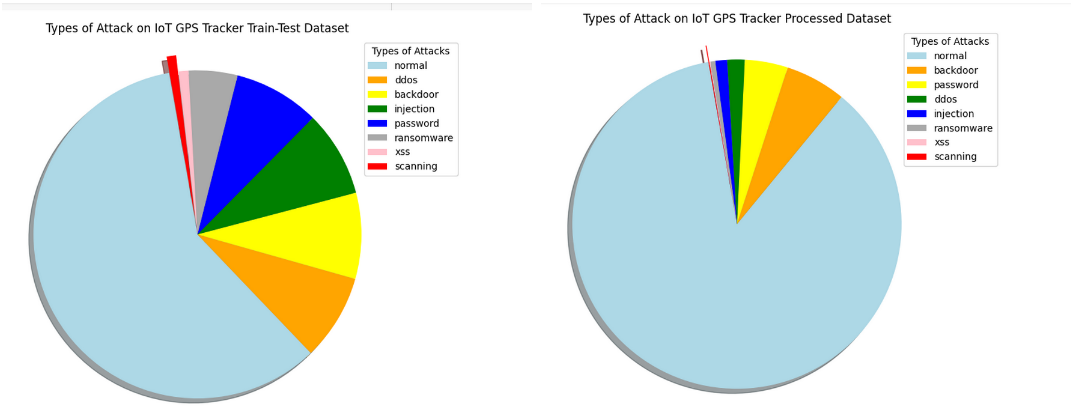


Figure 3.9: Statistics of IoT GPS Tracker

Important Features of IoT GPS Tracker

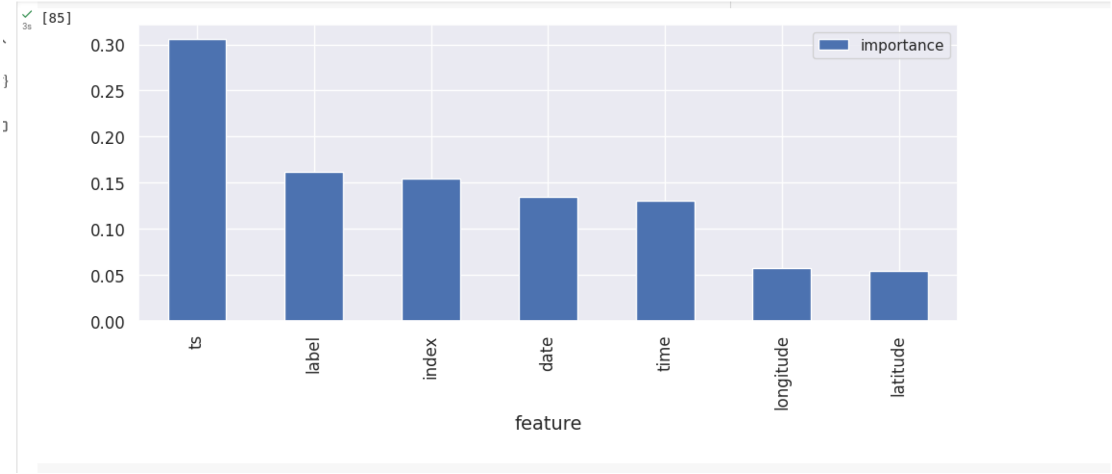


Figure 3.10: Important Features of GPS Tracker

3.6.6 IoT Modbus

3.6.6.1Features of IoT Modbus

1. date: Date

It is the logging Date of IoT telemetry data.

2. time: Time

It is the time logging of IoT telemetry data.

3. FC1_Read_Input_Register: Number

This feature is for reading an input register of Modbus function code.

4.FC2_Read_Discrete_Value: Number

This feature is for reading a discrete value of Modbus function code.

5. FC3_Read_Holding_Register: Number

This feature is for reading a holding register value of Modbus function code.

6. FC4_Read_Coil: Number Modbus function code that is responsible for reading a coil

7. label: Number

It is a tag for identifying normal and attack records. For identification 0 is for normal and 1 specify attacked record.

8. type String The "type" feature enlists the tags of attacks. Attacks are divided into categories, like normal, DoS, DDoS, password and backdoor attacks.

Listed Below is the count of attacks on the IoT Modbus dataset.

Labels	Train-Test Dataset	Processed Dataset
Normal	35000	222855
DDOS	-	-
Backdoor	5000	40011
Injection	5000	5186
Password	5000	18115
Ransomware	-	-
XXS	577	498
Scanning	529	529

Table 3.8: Statistics of IoT Modbus

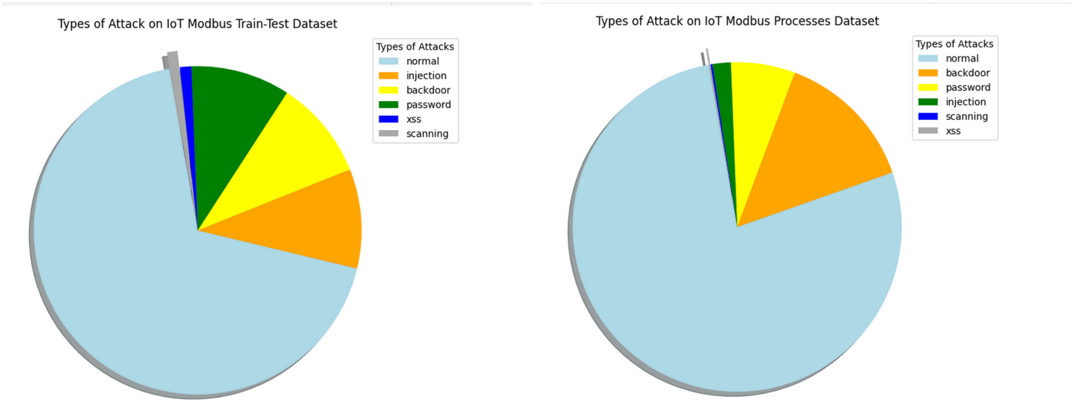


Figure 3.11: Statistics of IoT Modbus

Important Features of IoT Modbus

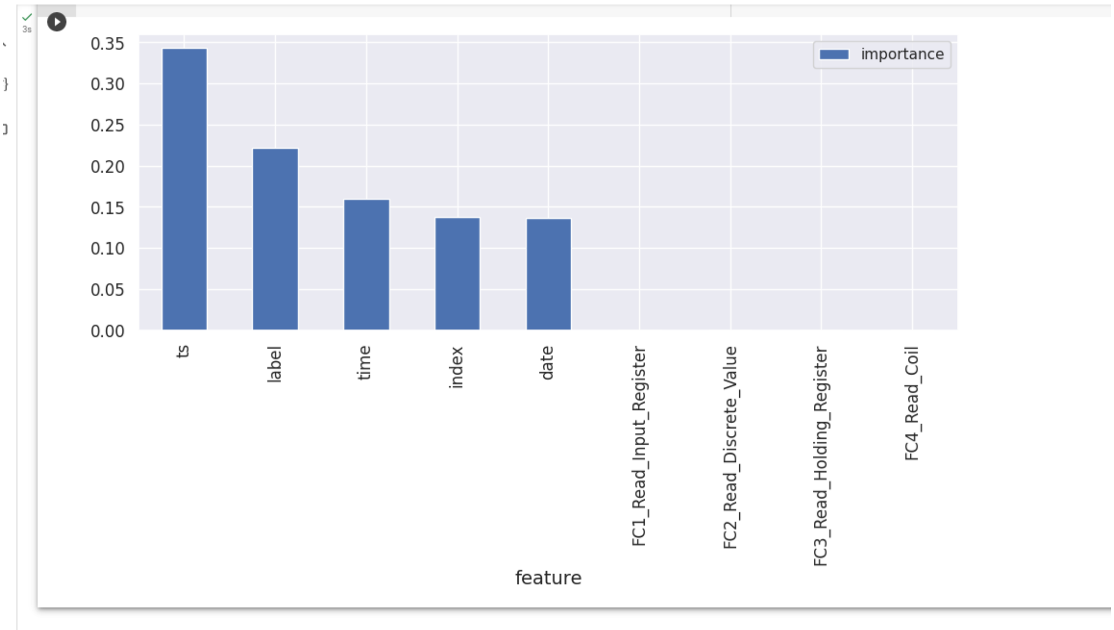


Figure 3.12: Important Features of Modbus

3.6.7 IoT Fridge

3.6.7.1 IoT Fridge Features

1. **date:** Date it is the logging Date of IoT telemetry data.

2. **time:** Time it is the time logging of IoT telemetry data.

3. **fridge_temperature:** Number It is the Numeric value for measuring temperature with linked sensor of IoT Fridge to the network.

4. **temp_condition:** String The data type of this feature is string. It is for measuring temperature with linked sensor of IoT Fridge to the network. The temperature is set to high of low based and threshold value is predefined.

5. **label:** Number

It is a tag for identifying normal and attack records. For identification 0 is for normal and 1 specify attacked record.

6. type: String The "type| feature enlists the tags of attacks. Attacks are divided into categories, like normal, DoS, DDoS, password and backdoor attacks.

Listed Below is the count of attacks on the IoT Fridge dataset.

Labels	Train-Test Dataset	Processed Dataset
Normal	35000	500827
DDOS	5000	35568
Backdoor	5000	28425
Injection	5000	10233
Password	5000	7079
Ransomware	2902	2902
XSS	2042	2042

Table 3.9: Statistics of IoT Fridge

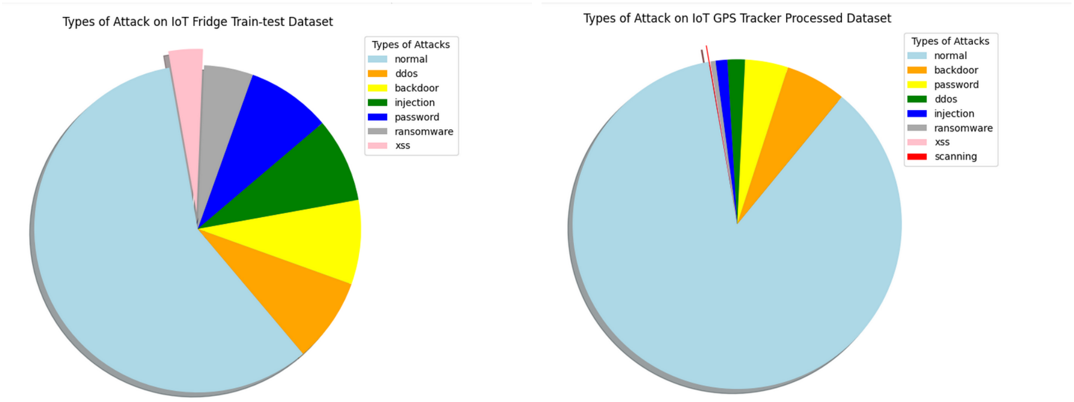


Figure 3.13: Statistics of IoT Fridge

Important Features of IoT Fridge

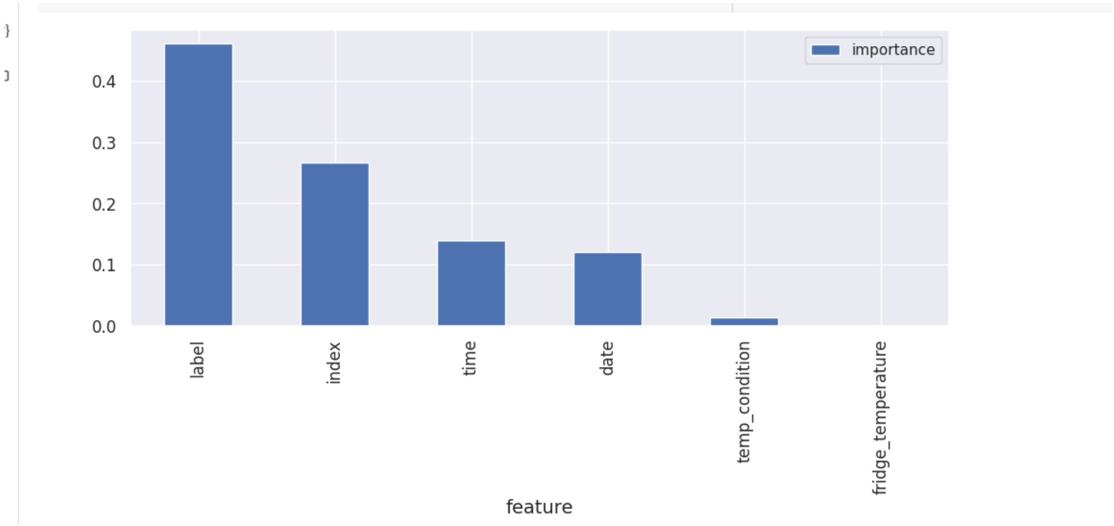


Figure 3.14: Important Features of Fridge

3.7 EXPERIMENTAL METHODOLOGY

In machine learning, it is a good practice to prepare and clean data to have an efficient model with good accuracy. The data preparation involves discarding data features that are not important and adversely affect the model performance. In data preparation missing values are also replaced or these data elements are also removed and non-numerical features of data into numerical ones. Data preparation is carried out by pre-processing and normalisation of data.

3.7.1 Required Resources

Listed below are the resources and libraries required for implementation of research models:

1. Lenovo Ideapad 320
2. Google Colab
3. Python

4. Pandas
5. Numpy
6. Seaborn
7. Matplotlib
8. Sklearn
9. Pytorch

3.7.2 Data Pre-processing

In data pre-processing, some features of data that are in string data type are converted into numeric values for applying the machine learning Hybrid Model algorithms. For instance, all the 'high' and 'low' or 'on' and 'off' values are converted into '0' and '1'. The second step of data pre-processing involves the omission of labels 'date', 'time' and 'timestamp'. These features are omitted because they are the root cause of data over-fitting. Important features are selected for Hybrid Model using Random Forest classifier and depicted in figures in dataset description section.

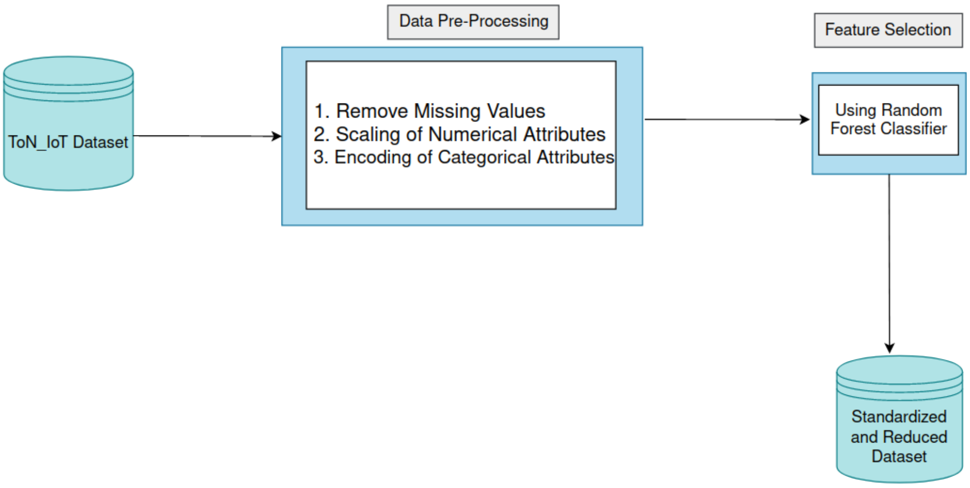

Data Pre-Processing
Feature Selection
ToN_IoT Dataset
1. Remove Missing Values
2. Scaling of Numerical Attributes
3. Encoding of Categorical Attributes
Using Random Forest Classifier
Standardized and Reduced Dataset

Figure 3.15: Pre-Processing of Data

In data pre-processing for Federated Learning Based Deep Auto-encoder, the dataset is divided into two parts based on the "type" label used for identifying normal and attacked data. All the normal data and anomalous data are separated. The normal data i.e. not affected by any kind of attack is divided into three parts one for deep auto-encoder training to extract essential information from it, the second, is used as an input to the trained model along with the mean square error of auto-encoder for computing the threshold to detect the anomalous data. The third one is combined with anomalous data and named mix_data for testing purposes. The loss values from the mix_data are compared with the threshold, if the loss is higher then it is detected as attacked and vice versa.

3.7.3 Federated Learning Based Deep Auto-Encoder Model

From the literature review, we came to know that different anomaly detection techniques are used. In this research, the federated learning-based deep auto-encoder model is used for intrusion detection. The characteristic of a Deep Autoencoder is that it learns all features and observations with different layers and neurons

3.7.4 Federated learning

Federated learning is a machine learning algorithm for training of highly centralized model trained over distributed clients that are connected with unreliable and relatively slow network. This learning algorithm have rounds where clients computes independently on local data to update the current model. The updated data is communicated to the central server for aggregation of client-side updated model to get a global model. In most of the cases clients are mobile phones or concerning applications where communication efficiency is necessary [54]. Steps followed by federated learning process are:

The basic infrastructure of Federated learning algorithm is based on two approaches of either asynchronous or synchronous training algorithms. Asynchronous

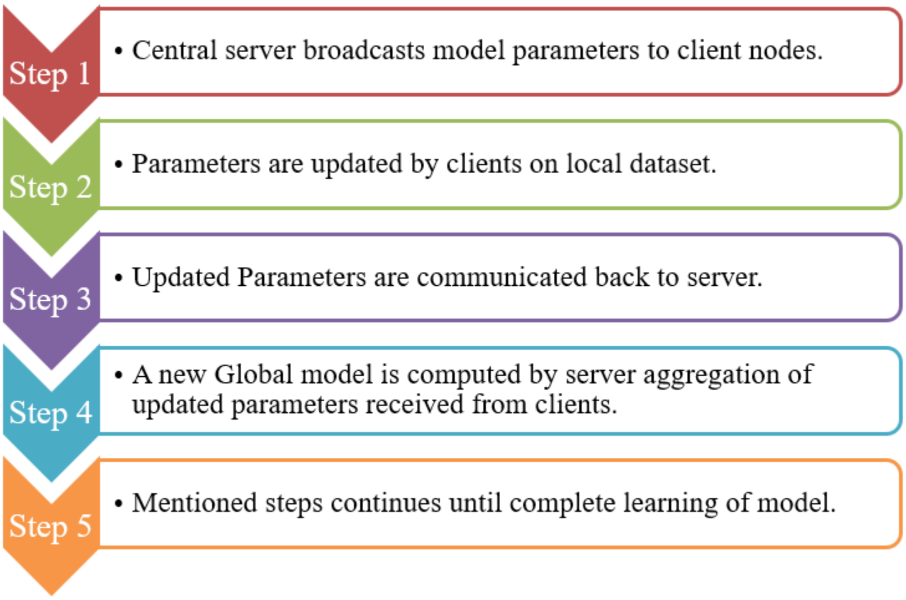

Step 1
• Central server broadcasts model parameters to client nodes.
Step 2
• Parameters are updated by clients on local dataset.
Step 3
• Updated Parameters are communicated back to server.
Step 4
• A new Global model is computed by server aggregation of updated parameters received from clients.
Step 5
• Mentioned steps continues until complete learning of model.

Figure 3.16: Steps of federated learning algorithm

online federated learning training has working principle where the edge nodes streams local data continuously with online learning and the central server is assigned with the task to aggregate parameters of model from client [55]. Recent research has trends to implement synchronous training of large batch at even data center [56, 57]. The McMahan's Federated learning algorithm has similar approach [58]. Research of Sin Kit Lo contributed in collection of architectural patterns for designing and development of real-world federated learning system [59].

In case of synchronized Federated learning algorithm, a round has following steps [54]:

1. A subset of existing clients downloads the current model.
2. Every client device performs computations on local data to train the model.
3. eights from trained model are communicated from the selected clients to the server.
4. The task of server is to aggregate the weights comes from communication round by averaging and send back the updated weights for next iteration and construction of an improved version of model.

3.7.4.1 Working of Deep Auto-encoder

Hinton and Zemel are the first developers of auto-encoder. The proposed model of auto-encoder is unsupervised which uses recognition weights for compressing input vectors into code vectors. The model uses generative weights to generate output by reconstructing the input vector[60]. Deep auto-encoder can work on linear feature compression [61]. The deep auto-encoder is used for detecting anomalous activities from the ToN-IoT dataset for IoT/IIoT devices . The Deep Auto-encoder model used in the research is illustrated in Figure. It works on three main layers named as encoder layer, the hidden layer and the decoder layer. Deep auto-encoder has the same size of target values and input values because it helps it in learning the representation of input data. The training of deep auto-encoder is carried out using the normal traffic data from the dataset to learn the features and essential characteristics. From Median et al [9] the reconstructed error is used for identifying the attacked data.

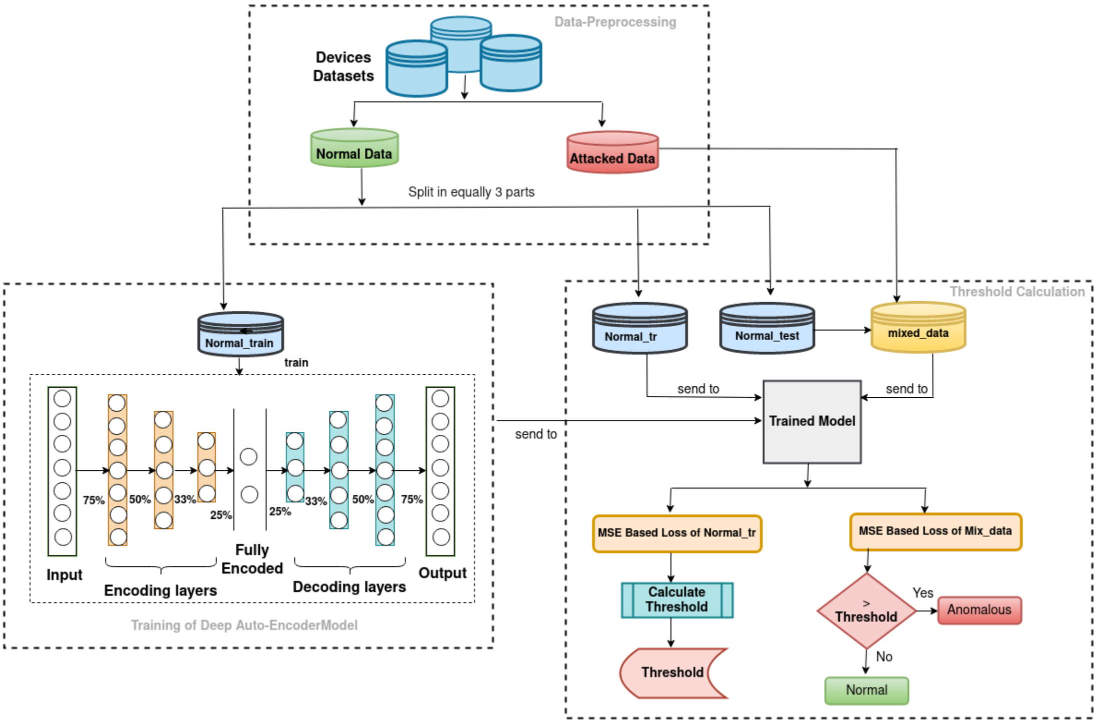


Figure 3.17: Working of Deep Auto-encoder Model

Like other Neural Networks, the Output of the encoding layer is input to the hidden layer and the output from the hidden layer is input to the decoder layer. Deep auto-encoder, in its training on normal data, uses the four convolutional neural network layers for input i.e. 75%, 50%, 33% and 25%. Due to the symmetrical nature of the Deep auto-encoder model, the decoder decodes it layer by layer after final compression as 25%, 33%, 50%, and 75%. The decompression is done through linear layers starting from the lowest encoding layer for data reconstruction [62].

3.7.4.2 Threshold Calculation

The threshold value is calculated for the purpose to identify the anomalous data. It is calculated after training of model and computing the mean square error. The threshold value is the summation of the mean square error and its standard deviation that comes from the training of the model on normal data. The instance when compared with the threshold value is less than it is normal data but if it is greater then the instance is identified as anomalous data. The equation below is for computing Threshold

tr = MSE normal_tr + std(MSE normal_tr)

3.7.4.3 Working of Federated Learning based Deep Auto-Encoder

Federated learning along with Deep Auto-encoder is used to achieve project goals. As mentioned in the literature Review, the algorithm of Federated Learning is helpful when training is performed on a large number of devices and solves the issues of privacy concerns. The primary advantage of training a model on federated Learning is that it does not share and store personal or sensitive information to global servers and hence results in ensuring privacy issues.

In the training process "clients" are the IoT/IIoT devices. The pre-processing of data is defined above where normal data is divided into three parts. The training of Federated Learning is carried out on all devices at the same time. The global server receives the weights of all devices that are the output of the Deep Auto-encoder. These weights are processed through the federated aggregation function. The Deep Auto-encoder receives the updated aggregated weights shared by the global server and proceeds with its training process.

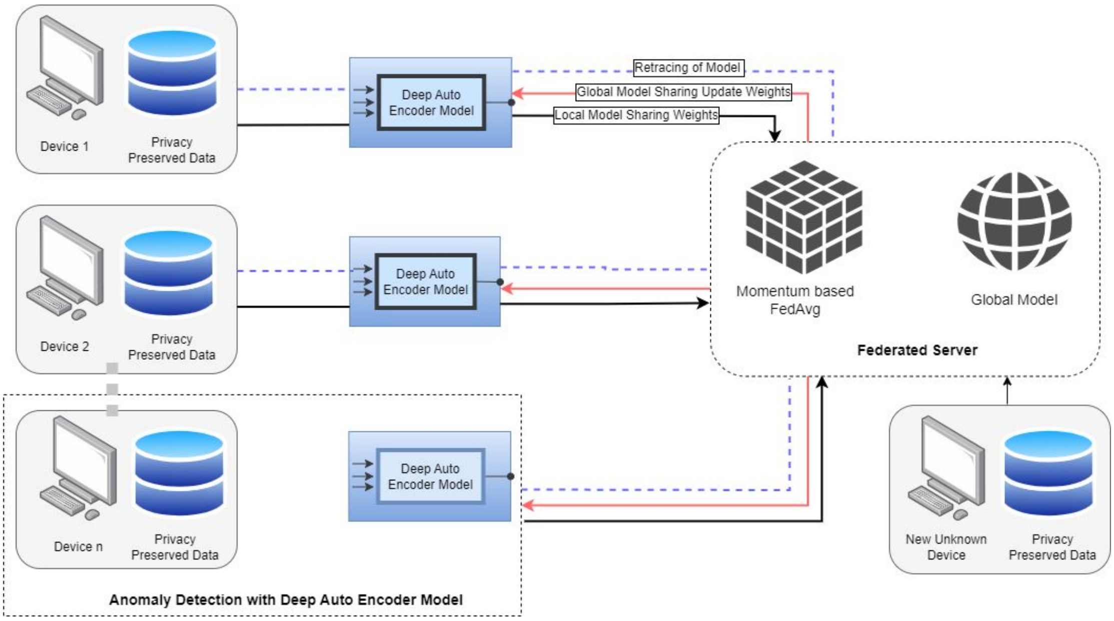


Figure 3.18: Federated Deep Auto-encoder Model for Anomaly Detection

The main process of the Federated communication round is a feed-forward process in which weights are sent to the global server after aggregation and the updated weights are propagated back. The count of communication rounds updates the weights for improving model performance. In the model, retraining is carried out for handling real-world scenarios and Independent and Identically Distributed data. In the retraining process, shown in fig 3.16, the client model from the Deep Auto-encoder is trained again on the random training data before it is sent to the global server for aggregation of weights [62].

The performance of the model is enhanced by partially selecting the clients. In partial selection, the federated training model randomly chooses the devices among all for its communication round. The trained federated model is useful for devices involved in the training process and any new one.

3.7.4.4 Criteria for client selection

The ToN-IoT devices dataset used in the research is heterogeneous and its distribution is non-Identical Independent in short non-IID. The federated model is data sensitive and dependent on features and classes of client devices data. The efficiency of the model is improved by client selection in comparison to using all clients. The client selection method minimizes the impact of non-IID data. If the model is trained on randomly selected five devices then from these five two devices are selected for training the global model. In each communication round, the global weights are assigned to these selected devices.

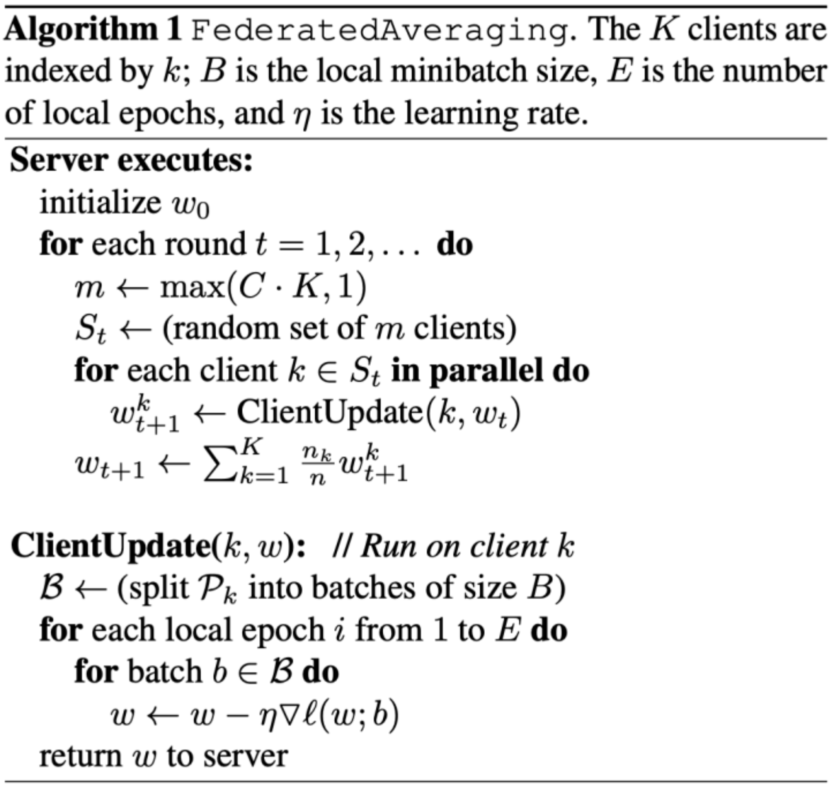
3.7.4.5 Retraining of Model

The retraining of the model is carried out to further improve the federated model performance. It also aids in resolving the usage of Non-IID data for real-time usage. For retraining of the model, 1000 random instances are used from the normal_train data. The global server aggregates the weights from communication rounds and weights from retaining round. These eights from the global server are then assigned to the local model to proceed with the training process and after completion of model training, it is used for testing to detect anomaly[62].

3.7.4.6 Momentum-based Federated Averaging Algorithm

The Federated averaging algorithm based on Momentum is used in this research. The algotithm is inspired by Hsu et al. [63]. The study elaborates that the performance and efficiency of the model are improved by around 35% to 75% when the averaging algorithm with momentum is used on the server side.

Momentum-based federated algorithm is the same as the federated averaging algorithm but momentum is used on top of it.



Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

```
Server executes:
  initialize $w_0$
  for each round $t = 1, 2, \dots$ do
    $m \leftarrow \max(C \cdot K, 1)$
    $S_t \leftarrow$ (random set of $m$ clients)
    for each client $k \in S_t$ in parallel do
      $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$

ClientUpdate($k, w$):  // Run on client k
  $\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size $B$)
  for each local epoch $i$ from 1 to $E$ do
    for batch $b \in \mathcal{B}$ do
      $w \leftarrow w - \eta \nabla \ell(w; b)$
  return $w$ to server
```

Figure 3.19: Federated Averaging Algorithm

Simple federated averaging uses the stochastic gradient descent (SGD) with the properties listed below:
1. it is used for efficiently solving federated problems by calculating the single batch gradient in every communication round.
2. It improves the algorithm as it is computationally efficient.
3. For achieving a good federated model, a large number of training rounds are required.

The working of federated learning is based on selecting a fraction of clients according to global batch size for computing the gradient of the loss on every communication round. Traditionally one client is selected for implementing it with a fixed learning rate. Weights are updated by averaging the gradients of the local model with momentum vector on stochastic gradient descent on the global server side. The steps of gradient descent are in the way that every client takes one step locally on the local model and the server is responsible for taking the average weights of the resulting model with the addition of the momentum vector. The process iterates for local updates by adding computations on each client for multiple times before calculating federated averaging. The computations for local updates are dependent on the fraction of clients, the batch size for client updates and training passes for clients in every communication round. It is seen that if data is non-IID then simple federated averaging algorithms diverge empirically. But in the case of momentum based federated Averaging the results show that it converges to the directions of curvatures with efficiency and efficient learning. According to the literature review, the momentum value is set as 0.9 because with this value convergence of accuracy and loss function soared up.

3.7.5 Hybrid Model Using Deep Learning Algorithms

For developing an efficient model for the stated issue, the hybrid approach is applied to the dataset to get optimal results. The hybrid approach first preprocesses the data then train the model and tests it. The algorithms used for hybrid modelling are Decision Tree, Random Forest and XGBoost.

3.7.5.1 Random Forest (RF)

The Random Forest algorithm is an ensemble of decision trees that enhances the accuracy of the model by averaging all trees. It builds multiple decision trees on different subsets of the training data and combines their predictions through voting or averaging. Random forests reduce over-fitting and improve generalization by introducing randomness into the tree-building process. Every tree and class of the RF algorithm computes a class that the majority of trees forecast about and results in the prediction of the model. The algorithm generates multiple independent trees from the training data and combines them for developing a single model by voting process.

3.7.5.2 Decision Tree (DT)

Decision trees are simple yet powerful models that partition the data based on the values of different features to make predictions. They are easy to interpret and can handle both numerical and categorical data. DT are applicable in image processing, machine learning models and pattern recognition. It is widely used for grouping data. Data mining uses a decision tree for the classification model. It is also used for regression tasks. The main components of Decision trees are the root node i.e. the whole dataset, branches that define the features of the dataset and leaf nodes are outputs for possible outcomes of the dataset. DT has a broad range of applications because it applies to all data types with precision and its analysis is easy.

3.7.5.3 XGBoost Algorithm

Extreme Gradient Boost (XGBoost) is an optimized algorithm for scalable model training. The learning algorithm combines predictions of multiple weak models to generate a strong one. It is considered as state of the art algorithm for classification and regression on large datasets. It efficiently handles the missing values from the dataset. It has the feature of parallel processing to train large datasets in a short period. It is widely used in recommendation systems and for fine-tuning parameters to get the efficient performance of the resultant model. For model training requirements adjust hyper-parameters, such as the learning rate, number of trees, or maximum depth, to optimize the model.

Models of each algorithm is designed to check their particular accuracy on every device dataset of ToN_IoT dataset and afterwards Hybrid Model is implemented.

3.7.5.4 Hybrid Approach

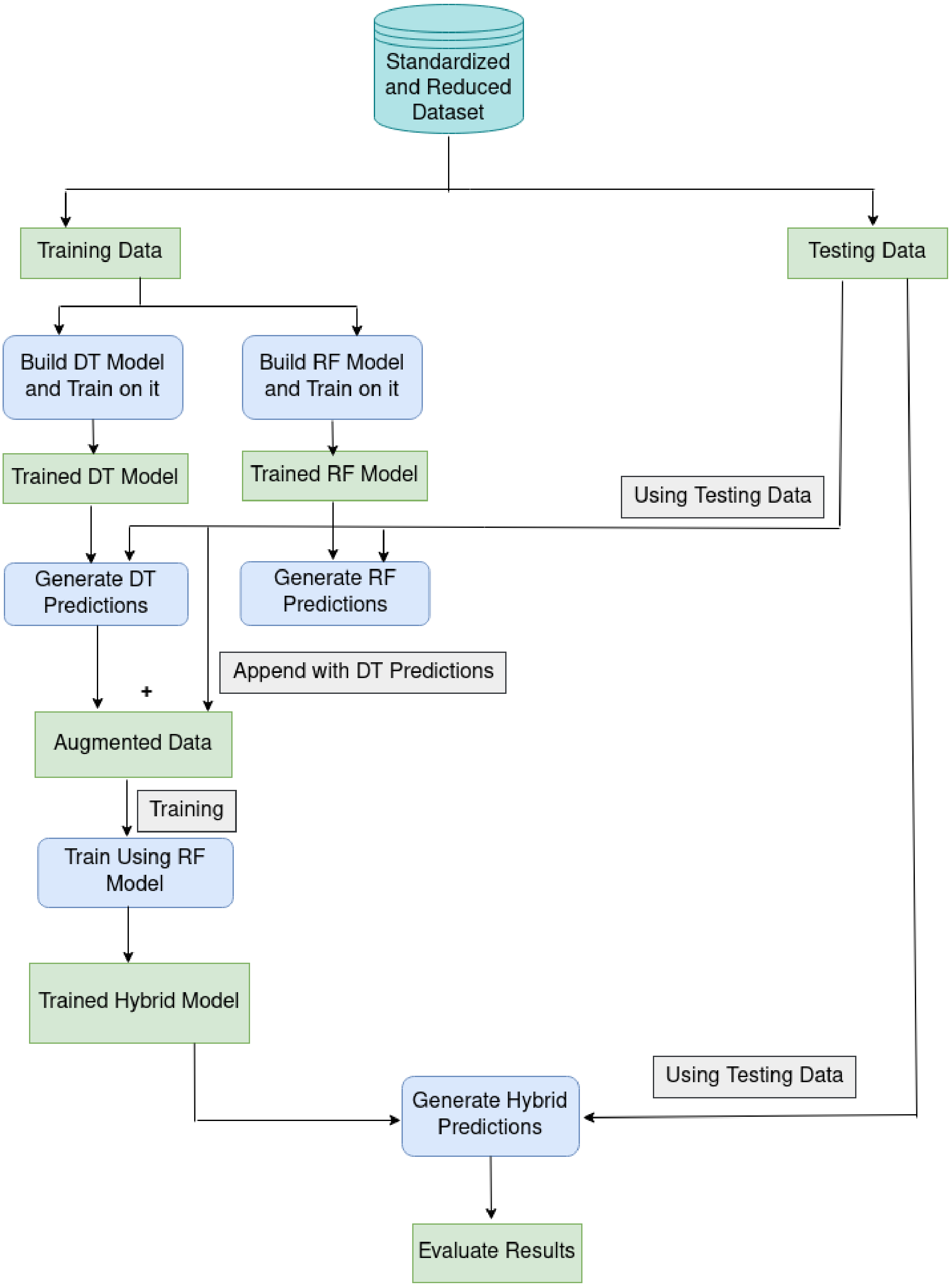


Figure 3.20: Hybrid Model for Anomaly Detection

Listed below are the steps to implement the Deep learning-based Hybrid approach for intrusion detection.

1. Process the datasets to remove missing values, scale numerical features and encode categorical features.

2. Important Features are selected on the bases of class labels and less correlated are excluded. A Random Forest classifier is applied to the dataset for feature selection.

3. Dataset is split into two parts for training and Testing.

4. On the mentioned machine learning algorithms, the dataset is trained using the training data. The trained model is tested to get predictions using testing data.

5. Any two algorithms are selected. Instead of using the original input features, append the predictions from the first algorithm as additional features to the training data. Then, train using the second algorithm on this augmented dataset. The trained model will learn to utilize both the original features and the predictions from the first model. For instance, append the predictions from the decision tree model as additional features to the training data and train the random forest on this augmented dataset. The random forest will learn original features and the decision tree predictions.

6. To make predictions using the hybrid model, pass the test data through the first model to get its predictions. Append these predictions as features to the test data and use the trained model of the second algorithm to make the final predictions.

A hybrid model helps in interpretability and yields an efficient predictive model using either algorithm where a single one is not able to capture complex relationships in the data.

Chapter 4

Results

4.1 Evaluation Metrics

In this section of research, the performance measures of all algorithms are discussed according to the listed parameters and used in literature for first-hand evaluation and analysis of federated learning-based deep auto-encoder and hybrid models for intrusion detection on the Ton-IoT dataset. Both models are evaluated using Accuracy, F1-score, Precision and Recall. These measures are performance analyzing metrics. The terms used for the purpose are:

True Positive (TP): It is the count of occurrences through which we accurately identify attacks from dataset.

True Negative (TN): It is the count of occurrences through which we accurately identify normal data.

False Positive (FP): It is the count of occurrences in which normal instances are classified as attacks.

False Negative (FN): It is the count of occurrences in which attacked instances are classified as normal.

4.1.0.1 Performance Metrics

1. Accuracy

Accuracy is defined as the proportional detection of accurately identified instances to total count of instances.

$$Accuracy = \frac{TP + TN}{(TP + FP + FN + TN)}$$

2. F1-Score

F1 Score is representation of the Harmonic mean between Precision and Recall.

$$F1score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

3. Precision

Precision is defined as the proportion of True Positives to overall positively predicted results.

$$Precision = \frac{TP}{(FP + TP)}$$

4. Recall

It corresponds to the percentage of predicted positive instances out of the total positive instances. It is also known as True Positive Rate (TPR).

$$Recall/TPR = \frac{TP}{(FN + TP)}$$

5. False Positive Rate

The False Positive Rate (FPR) is defined as the percentage of incorrect detection of positive instances to the actual number of negative instances. It is also called False Alarm Rate.

$$FPR = \frac{FP}{(FP + TN)}$$

Mean Square Error (MSE)

In machine learning MSE is used to measure the error rate on average squared difference between the actual values and the predicted values of a model. Performance of a regression model is measured using MSE and its goal is to predict continuous numerical values.

The formula for calculating Mean Square Error is:

$$MSE = \frac{1}{n} * \sum (actual - predicted)^2$$

where:

n is the total sample count of dataset.
actual also known as ground truth is the actual target value.
predicted is the count of target variable which are predicted by using regression model.

4.2 Federated learning-based deep auto-encoder model results

According to the method discussed in the methodology section, the experiment is set up to test Federated Learning Performance using a Deep auto-encoder. The test setup includes FedAvg for aggregation of global and client model results. To evaluate the model results on the Ton-Iot datasets for seven IoT devices, the accuracy, F1 score, precision, recall, TPR and FPR were compared.

The Hyper-Parameters used for experimental setup are given below:

Parameters	Values	Meaning
lr	0.012	Learning rate
No_of_clients	7	Number of all clients
Selected_no	7	Number of clients used for training purpose
Batch size	128	It is the size of data for each iteration of training
Baseline_no	1000	This indicates the selected data from trained one in order to retrain the already trained model
No_of_rounds	6	Number of rounds for training the global model
No_of_Epochs	5	Total number for training the clients.
No_of_retraining_Epochs	12	Total number for global server retraining after weights are assigned

Table 4.1: Parameters for Federated Learning Model

4.3 Results of Train-Test datasets

Dataset Devices	Accuracy %	F1 score %	Precision	Recall	TPR	FPR
Garage Door	**96.916**	**97.777**	0.957	1.000	1.00000	0.09583
Weather	51.418	50.853	0.800	0.371	0.37370	0.19371
Motion Light	**87.852**	**91.770**	0.848	1.000	1.000	0.37645
Thermostat	41.259	18.386	0.570	0.110	0.10960	0.12582
GPS Tracker	50.119	46.865	0.826	0.327	0.32709	0.14125
Modbus	55.342	48.405	0.733	0.361	0.36123	0.18128
Fridge	53.410	52.539	0.859	0.378	0.37849	0.13320

Table 4.2: Result Table for Train-test datasets

4.4 Results of Processed Datasets

Dataset Devices	Accuracy %	F1 score %	Precision	Recall	TPR	FPR
Garage Door	53.531	11.881	0.148	0.099	0.09936	0.26397
Weather	**70.694**	45.734	0.579	0.378	0.37801	0.13347
Motion Light	54.142	35.033	0.329	0.374	0.37396	0.37586
Thermostat	67.969	40.552	0.466	0.359	0.35901	0.18003
GPS Tracker	44.129	0.162	0.002	0.001	0.00140	0.34853
Modbus	64.989	54.005	0.692	0.443	0.44286	0.17079
Fridge	68.585	49.417	0.547	0.450	0.45048	0.19254

Table 4.3: Result Table for Processed Datasets

4.5 Results of Single ML Algorithm

Dataset	DT	RF	XGBoost
Garage Door	100	100	100
Weather	58	84	69
Motion Light	59	58	88
Thermostat	59	66	66
GPS Tracker	84	85	86
Modbus	67	97	77
Fridge	57	97	81

Table 4.4: Accuracy Table of Hybrid Model

4.6 Result for hybrid Model

In order to evaluate the performance of dataset the achieved accuracy of Hybrid Model on all devices is depicted in the Table 4.4 and Table 4.5 is for Precision, Recall and F1 Score while Average Training Time, Prediction Time, Validation Score and Mean Square Error are represented in Table 4.6.

Dataset	DT + XGBoost	RF + XGBoost	RF + DT
Garage Door	99.99	99.98	99.98
Weather	100	100	100
Motion Light	99.99	99.99	99.99
Thermostat	99.99	99.99	99.99
GPS Tracker	99.99	99.99	99.99
Modbus	99.99	99.99	99.99
Fridge	100	100	100

Table 4.5: Accuracy Table of Hybrid Model

Dataset	Hybrid Model	Precision	Recall	F1 Score
Garage Door	DT + XGBoost	0.9994	0.999	0.9992
	RF + XGBoost	0.9994	0.999	0.9992
	RF + DT	0.9994	0.999	0.9992
Weather	DT + XGBoost	1	1	1
	RF + XGBoost	1	1	1
	RF + DT	1	1	1
Motion Light	DT + XGBoost	0.9996	0.998	0.9997
	RF + XGBoost	0.9996	0.9998	0.9997
	RF + DT	0.9996	0.9998	0.9997
Thermostat	DT + XGBoost	0.857	0.8571	0.8571
	RF + XGBoost	0.857	0.8571	0.8571
	RF + DT	0.857	0.8571	0.8571
GPS Tracker	DT + XGBoost	0.9999	0.9997	0.9998
	RF + XGBoost	0.9999	0.9997	0.9998
	RF + DT	0.9999	0.9997	0.9998
Modbus	DT + XGBoost	0.9996	0.9848	0.9918
	RF + XGBoost	0.9996	0.9848	0.9918
	RF + DT	0.9996	0.9848	0.9918
Fridge	DT + XGBoost	1	1	1
	RF + XGBoost	1	1	1
	RF + DT	1	1	1

Table 4.6: Table for Precision, Recall and F1 Score of Hybrid Model

Dataset	Hybrid Model	Training Time	Prediction Time	Validation Score	MSE
Garage Door	DT + XGBoost	1.5662	0.0161	0.9998	0.0006
	RF + XGBoost	1.5383	0.0157	0.9998	0.0006
	RF + DT	0.4844	0.1162	0.9998	0.0006
Weather	DT + XGBoost	2.8387	0.0334	0.9999	0
	RF + XGBoost	2.8547	0.0262	0.9999	0
	RF + DT	0.6535	0.0314	0.9999	0
Motion Light	DT + XGBoost	1.2238	0.0296	0.9997	0.0004
	RF + XGBoost	3.8281	0.0277	0.9997	0.0004
	RF + DT	0.3829	0.0198	0.9997	0.0004
Thermo -stat	DT + XGBoost	1.1324	0.0229	1	0.0011
	RF + XGBoost	1.1018	0.0167	1	0.0011
	RF + DT	0.3709	0.0132	1	0.0011
GPS Tracker	DT + XGBoost	2.0926	0.0177	0.9998	0
	RF + XGBoost	2.1066	0.0188	0.9998	0
	RF + DT	0.5258	0.0157	0.9998	0
Modbus	DT + XGBoost	0.9795	0.0091	0.9998	0.0006
	RF + XGBoost	0.9679	0.0116	0.9998	0.0006
	RF + DT	0.2662	0.0114	0.9998	0.0006
Fridge	DT + XGBoost	0.9469	0.0134	0.9999	0
	RF + XGBoost	0.9514	0.0183	0.9999	0
	RF + DT	0.3388	0.012	0.9999	0

Table 4.7: Table for Training Time, Prediction Time, Mean Validation Score and Mean Square Error

Chapter 5

Discussion

5.1 Discussion on Federated Learning-based Deep Auto-encoder Model

5.1.1 Comparative Results of Both Datasets

5.1.1.1 Accuracy and F1 score

For evaluating the Federated Learning using the Deep Auto-encoder model, the Accuracy and F1 score of all devices are computed. From all processed datasets of IoT devices, the accuracy of the IoT_Weather dataset is good among all which is 70.694% while the F1 score is 45.734%. IoT_Modbus dataset has the F1 score of 54.005 with an accuracy of model 64.989. From the train-test dataset, the accuracy of the Garage Door dataset is good among all that is 96.916% with the highest F1-score of 97.777%. The graphs below represent the relation between accuracy and F1 Score of all devices of the Ton_IoT dataset.

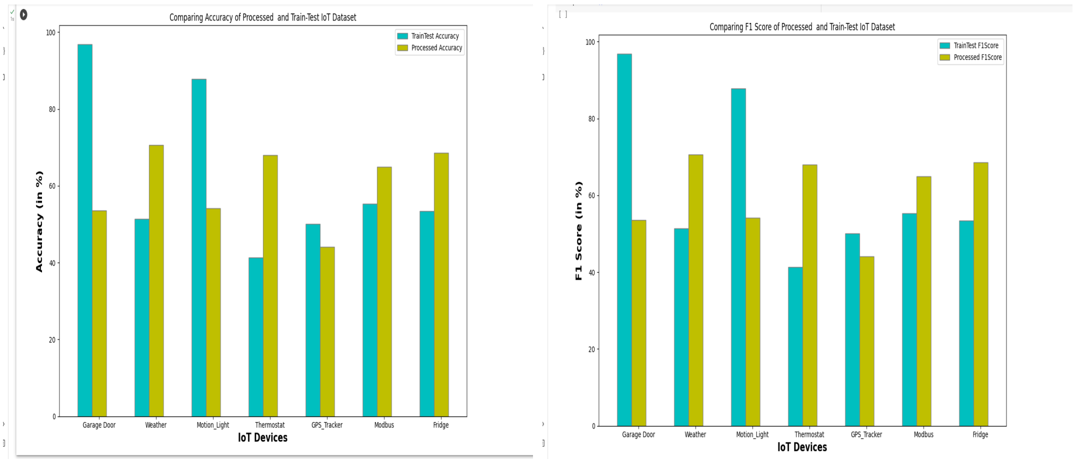


Figure 5.1: Comparison of Accuracy and F1 score on Train_Test and Processed Dataset

5.1.1.2 Precision and Recall

For evaluating the Federated Learning using Deep Auto-encoder model, precision and recall of all devices are computed. The graph below represents the relation between precision and recall of all devices of Ton_IoT dataset.

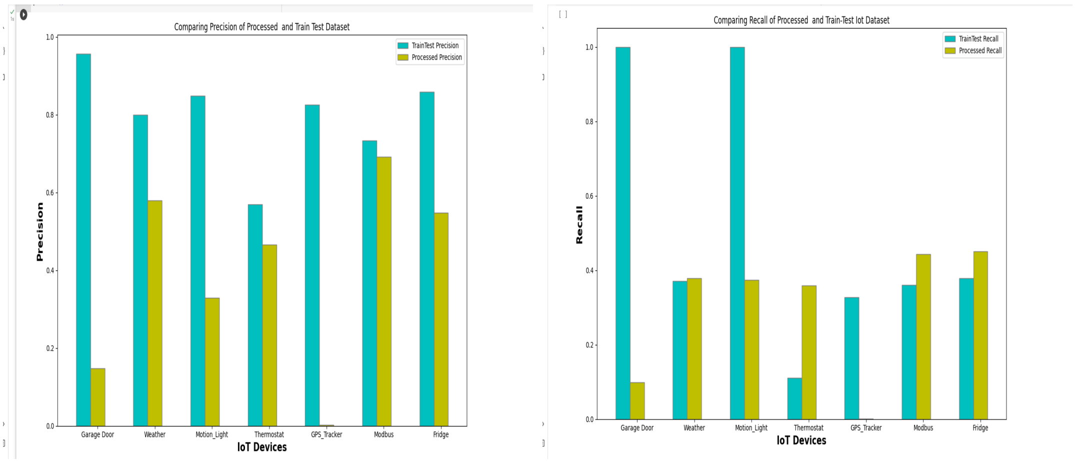
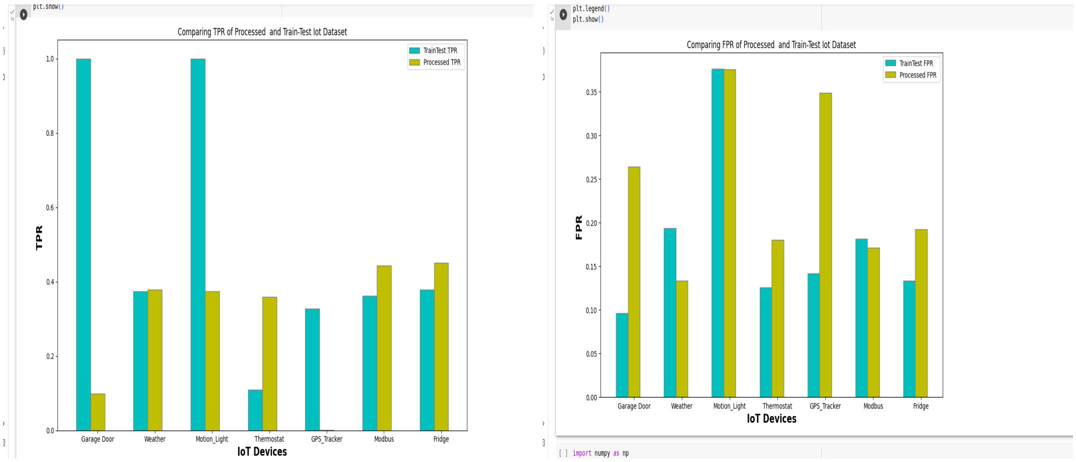


Figure 5.2: Comparison of Precision and Recall on Train_Test and Processed Dataset

5.1.1.3 TPR and FPR

For more detailed evaluation the TPR and FPR values computed from designed Federated Learning based Deep Auto-encoder model are computed and fig below show the Performance of all Ton_IoT devices.



```
plt.show()
```

```
plt.legend()
plt.show()
```

```
import numpy as np
```

Figure 5.3: Comparison of TPR and FPR on Train_Test and Processed Dataset

5.1.1.4 Loss of Model

The fig show the global loss graph of Federated Learning based Deep Auto-encoder model. From the graph it is inferred that the loss of trained model is constant i.e. 0.68 for both Train-Test IoT dataset and Processed IoT dataset.

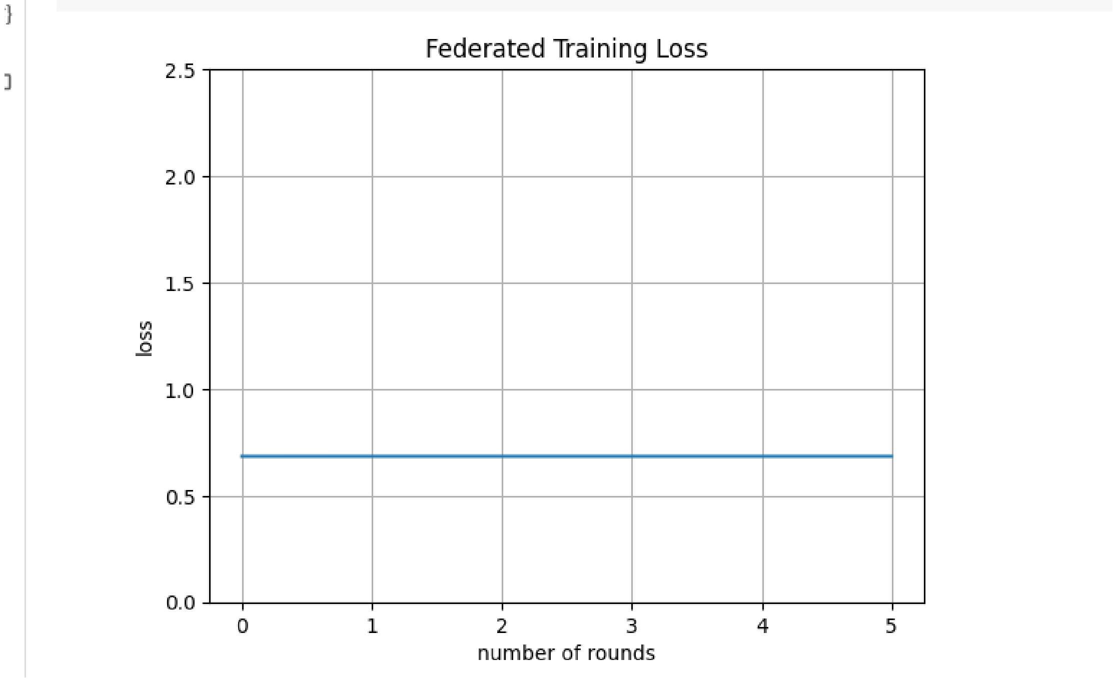


Figure 5.4: Training Loss of Models on Both Datasets

5.2 Discussion on Result of Single ML Algorithm

The performance of single ML algorithm i.e. Decision Tree, Random Forest and XGBoost is not satisfactory on the Ton_IOT Dataset. So, to improve the performance and to improve the intrusion detection Hybrid model is designed.

5.3 Discussion on Result of Hybrid Model

The combine effect of two ML algorithms improves the learning rate of designed model. In order to evaluate the performance of dataset using the hybrid models the accuracy and F1 score of all devices are compared. The highest accuracy is 100% for IoT Weather and IoT Fridge Dataset. The minimum accuracy among all devices is 99.98%.

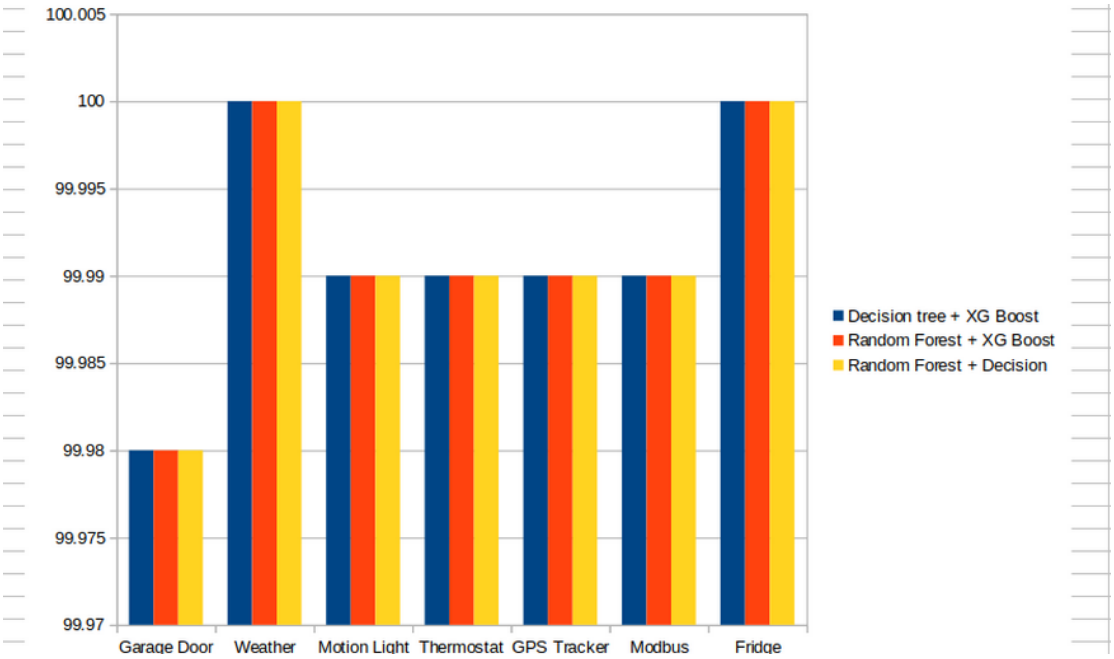


Figure 5.5: Accuracy Graph of Hybrid Model

5.3.1 Precision, Recall and F1 score of Hybrid Model

To evaluate the performance hybrid Models on datasets the precision, recall and F1 score is also computed and compared. The dataset of Iot weather and IoT Fridge has higher values of mentioned measures. The table below shows the performance measures of precision, recall and F1 score. The Fig 4.6 below illustrates the performance measure of Precision, Recall and F1 Score of Hybrid Model

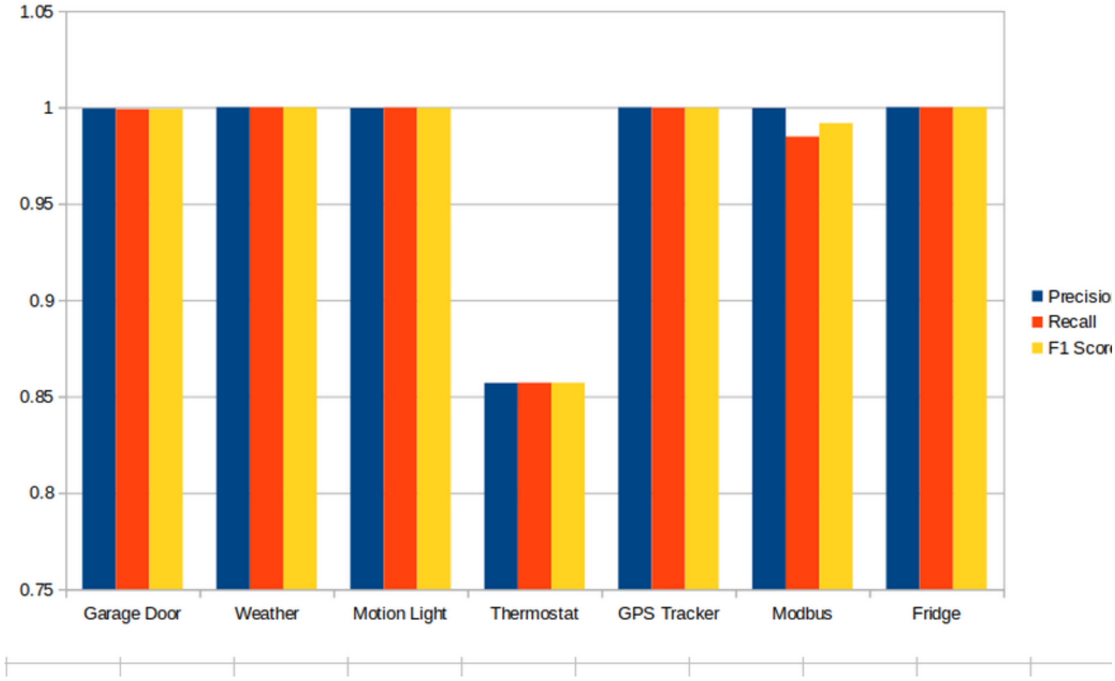


Figure 5.6: Performance Measure of Precision, Recall and F1 Score of Hybrid Model

5.4 Different Performance Measures of Hybrid Model

Hybrid Model is evaluated on different performance measures which are training time, prediction time, mean validation score and mean square error. Decision Tree + XG Boost and Random Forest + XG Boost models takes more training time on IoT Weather and IoT GPS Tracker dataset. The Validation Score of IoT

Thermostat is highest among all IoT Devices. Mean square error of IoT Weather, GPS Tracker and Fridge is zero. The Mean Square Error of ToN_IoT devices dataset is illustrated in Fig 4.7.

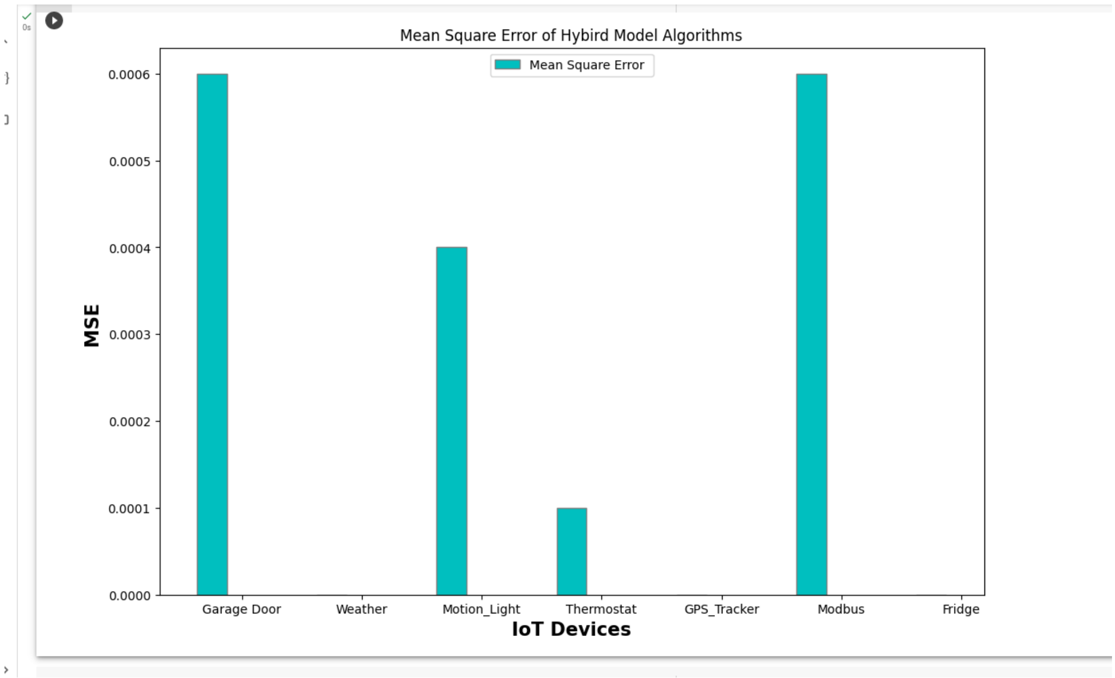


Figure 5.7: Mean Square Error of Hybrid Model on IoT Devices

5.5 Limitations

In the research under progress, some limitation exists which need some improvement in future and most of which are due to the time and resource constraint of the project.

5.5.1 Sample size

Traditionally, enormous devices for training and testing were used by federated learning in previous literature. However, only seven devices are included in the TON_IoT dataset which is used in this report. Hence, the sample size may be too small to achieve a whole performance of federated learning.

5.5.2 Dataset Quality

Attacked and normal data distribution was uneven. Many devices had a large number of normal data and attacked traffic is less in comparison. Such uneven distribution may lead to less accuracy of devices while model training except for Garage Door and Motion Light.

5.5.3 Auto-encoder model in previous literature

Literature results displayed that Auto-encoder has the best performance, hence, the data selection process and threshold calculation methods from previous literature are used. Methodology from previous literature limits us due to the requirement of different percentages of training and testing data to accumulate the best threshold figures for the auto-encoder model.

5.5.4 Design Challenges of Federated Learning

The federated learning model is very sensitive to Non-IID data properties of local clients; hence it is not very stable. We adopted the easy way of FedAvg based on Momentum for the aggregation of algorithms. Other literature exists that focuses on the solution to resolve the issue of Non-IID in Federating Learning.

In comparison to the concept of centralized machine learning, federated learning had advantage to deal with issues of data privacy and lack of training data but architectural challenge arises because of large scale distributed nature of federated learning while managing data of central server and client devices. There are four main challenges Expensive communication, Systems heterogeneity, Statistical heterogeneity, Privacy concerns while implementation of federated learning [64]. Some of the design challenges by Sin Kit Lo in his research are listed below [59]:
1. Global models have low accuracy and they lack in generality in case when client devices generate non-IID data. In conventional machine learning algorithm centralizing and randomizing techniques are used for data heterogeneity issue but federated learning has privacy preserving nature that renders mentioned techniques inappropriate.
2. High quality models require high communication cost because they need to adhere concept drift, they also require multiple rounds to exchange updates of local updates.
3. The model quality is affected by the limitation of resources required in model training and in data communication.
4. In federated learning process coordination is difficult because of many client devices participate that also affect the security and reliability of system.

5.5.5 Tuning of Hyper-parameter

It has been exercised on batch set, however, F1 score and accuracy have no difference while hyper-parameter tuning.

5.5.6 Platform Constraints

Google Colab is used to run codes for all models. Momentum-based federated averaging requires many training rounds (minimum 300 rounds) to formulate a good model. But it is time taking and results in Colab crashing.

Chapter 6

Conclusion

6.1 Conclusion

The purpose of the study is to enhance the efficiency of Intrusion Detection Systems. To achieve this TON_IoT dataset for IoT and IIoT devices is used to evaluate the performance of cybersecurity-based IoT devices using Artificial Intelligence. ToN_IoT dataset is comprised of data collected from different heterogeneous data sources about Fridge, Garage Door, Weather, GPS Tracker, Modbus, Motion Light and Thermostats. Processed and Train-Test IoT and IIoT devices dataset was the main target to analyses the intrusion detection using federated learning-based deep auto-encoder and Hybrid Models. The IoT/IIoT dataset is evaluated in terms of measuring accuracy, F1 score, precision, Recall, TRP and FRP.

On the Processed dataset the accuracy score of Deep auto-encoder based on federated learning is very low for all of the seven IoT devices. One of the main reasons for the cause can be that data is not uniformly distributed. The total count of normal data is very large in comparison to the anomalous one in the dataset. The second reason is that the traditional federated learning models uses a very large no of clients for their training purpose. So, from this fact, it is inferred that if the number of clients or IoT devices is large in number then the federated learning-based deep auto-encoder model works efficiently. While on the Train-Test Ton-IoT dataset, the federated model gives 96.9% and 87.8% accuracy on the IoT Garage Door and IoT Motion Light dataset. From these results, it is analyzed that most features of these two datasets are in binary form except “type” which is for identifying the normal or attacked data (DDoS, ransomware, password, injection etc). The boolean features of Garage Door are door_state, sphone_signal and label while the binary features of Motion Light are motion_status, light_status and label. Other IoT devices datasets have different features and their accuracy can be enhanced by converting features into the same binary format but it is not possible because of their working nature.

The hybrid model for intrusion detection performs excellently on ToN-IoT datasets. Result of model out performs for all devices. The main reason for this is that the XGBoost algorithm combines all weak classifiers into strong ones for providing better efficiency. The learning rate of XGBoost is more than other algorithms so it performs well in case when combined with Random Forest and Decision Tree. The performance of Hybrid Random Forest and Decision Tree is outstanding and the risk of overfitting is reduced with their combined effect.

6.1.1 Future Work

In the future, federated learning techniques should be explored more for defining a state of art model with large and balanced IoT/IIoT dataset. The future work may include use of Machine Learning Algorithms along with Federated Learning or FL Frameworks for efficient intrusion detection on the ToN_IoT Dataset.

References

[1] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions, 2012.

[2] Abdullah Alsaedi, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar. Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. *IEEE Access*, 8, 09 2020. doi: 10.1109/ACCESS.2020.3022862.

[3] SVN Santhosh Kumar, M Selvi, A Kannan, et al. A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things. *Computational Intelligence and Neuroscience*, 2023, 2023.

[4] Srinidhi N N, S.M. Kumar, and Venugopal K R. Network optimizations in the internet of things: A review. *Engineering Science and Technology, an International Journal*, 22, 09 2018. doi: 10.1016/j.jestch.2018.09.003.

[5] Imran Makhdoom, Mehran Abolhasan, Justin Lipman, Ren Ping Liu, and Wei Ni. Anatomy of threats to the internet of things. *IEEE Communications Surveys Tutorials*, 21(2):1636–1675, 2019. doi: 10.1109/COMST.2018.2874978.

[6] Jiadong Ren, Jiawei Guo, Wang Qian, Huang Yuan, Xiaobing Hao, and Hu Jingjing. Building an effective intrusion detection system by using hybrid data optimization based on machine learning algorithms. *Security and communication networks*, 2019, 2019.

[7] Mainduddin Ahmad Jonas, Md Shohrab Hossain, Risul Islam, Husnu S Narman, and Mohammed Atiquzzaman. An intelligent system for preventing ssl stripping-based session hijacking attacks. In *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*, pages 1–6. IEEE, 2019.

[8] Bruno Bogaz Zarpelão, Rodrigo Sanches Miani, Cláudio Toshio Kawakani, and Sean Carlisto de Alvarenga. A survey of intrusion detection in internet of things. *Journal of Network and Computer Applications*, 84:25–37, 2017.

[9] Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, 2018.

[10] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[11] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. Loadaboost: Loss-based adaboost federated machine learning with reduced computational complexity on iid and non-iid intensive care data. *Plos one*, 15(4):e0230706, 2020.

[12] Shaashwat Agrawal, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Mamoun Alazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. Federated learning for intrusion detection system: Concepts, challenges and future directions. *Computer Communications*, 2022.

[13] Sawsan Abdul Rahman, Hanine Tout, Chamseddine Talhi, and Azzam Mourad. Internet of things intrusion detection: Centralized, on-device, or federated learning? *IEEE Network*, 34(6):310–317, 2020.

[14] Zhuo Chen, Na Lv, Pengfei Liu, Yu Fang, Kun Chen, and Wu Pan. Intru-

sion detection for wireless edge networks based on federated learning. *IEEE Access*, 8:217463–217472, 2020.

[15] Enrique Mármol Campos, Pablo Fernández Saura, Aurora González-Vidal, José L Hernández-Ramos, Jorge Bernal Bernabé, Gianmarco Baldini, and Antonio Skarmeta. Evaluating federated learning for intrusion detection in internet of things: Review and challenges. *Computer Networks*, 203:108661, 2022.

[16] Davy Preuveneers, Vera Rimmer, Ilias Tsingenopoulos, Jan Spooren, Wouter Joosen, and Elisabeth Ilie-Zudor. Chained anomaly detection models for federated learning: An intrusion detection case study. *Applied Sciences*, 8 (12):2663, 2018.

[17] Yi Liu, Sahil Garg, Jiangtian Nie, Yang Zhang, Zehui Xiong, Jiawen Kang, and M Shamim Hossain. Deep anomaly detection for time-series data in industrial iot: A communication-efficient on-device federated learning approach. *IEEE Internet of Things Journal*, 8(8):6348–6358, 2020.

[18] Tae-Young Kim and Sung-Bae Cho. Web traffic anomaly detection using c-lstm neural networks. *Expert Systems with Applications*, 106:66–76, 2018.

[19] Naveed Chouhan, Asifullah Khan, et al. Network anomaly detection using channel boosted and residual learning based deep convolutional neural network. *Applied Soft Computing*, 83:105612, 2019.

[20] Yifan Guo, Weixian Liao, Qianlong Wang, Lixing Yu, Tianxi Ji, and Pan Li. Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach. In *Asian Conference on Machine Learning*, pages 97–112. PMLR, 2018.

[21] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[22] Kuang-Yao Lin and Wei-Ren Huang. Using federated learning on malware classification. In *2020 22nd international conference on advanced communication technology (ICACT)*, pages 585–589. IEEE, 2020.

[23] Viraaji Mothukuri, Prachi Khare, Reza M Parizi, Seyedamin Pouriyeh, Ali Dehghantanha, and Gautam Srivastava. Federated-learning-based anomaly detection for iot security attacks. *IEEE Internet of Things Journal*, 9(4): 2545–2554, 2021.

[24] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N Asokan, and Ahmad-Reza Sadeghi. Dïot: A federated self-learning anomaly detection system for iot. In *2019 IEEE 39th International conference on distributed computing systems (ICDCS)*, pages 756–767. IEEE, 2019.

[25] Yang Qin and Masaaki Kondo. Federated learning-based network intrusion detection with a feature selection approach. In *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pages 1–6. IEEE, 2021.

[26] Raed Abdel Sater and A Ben Hamza. A federated learning approach to anomaly detection in smart buildings. *ACM Transactions on Internet of Things*, 2(4):1–23, 2021.

[27] Marc-Oliver Pahl and François-Xavier Aubet. All eyes on you: Distributed multi-dimensional iot microservice anomaly detection. In *2018 14th International Conference on Network and Service Management (CNSM)*, pages 72–80. IEEE, 2018.

[28] William Schneble and Geethapriya Thamilarasu. Attack detection using federated learning in medical cyber-physical systems. In *Proc. 28th Int. Conf. Comput. Commun. Netw.(ICCCN)*, volume 29, pages 1–8, 2019.

[29] Ying Zhao, Junjun Chen, Di Wu, Jian Teng, and Shui Yu. Multi-task network anomaly detection using federated learning. In *Proceedings of the 10th international symposium on information and communication technology*, pages 273–279, 2019.

[30] Weishan Zhang, Qinghua Lu, Qiuyu Yu, Zhaotong Li, Yue Liu, Sin Kit Lo, Shiping Chen, Xiwei Xu, and Liming Zhu. Blockchain-based federated learning for device failure detection in industrial iot. *IEEE Internet of Things Journal*, 8(7):5926–5937, 2020.

[31] Yulin Fan, Yang Li, Mengqi Zhan, Huajun Cui, and Yan Zhang. Iotdefender: A federated transfer learning intrusion detection framework for 5g iot. In *2020 IEEE 14th international conference on big data science and engineering (BigDataSE)*, pages 88–95. IEEE, 2020.

[32] Yuwei Sun, Hideya Ochiai, and Hiroshi Esaki. Intrusion detection with segmented federated learning for large-scale multiple lans. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020.

[33] Kun Li, Huachun Zhou, Zhe Tu, Weilin Wang, and Hongke Zhang. Distributed network intrusion detection system in satellite-terrestrial integrated networks using federated learning. *IEEE Access*, 8:214852–214865, 2020.

[34] Noor Ali Al-Athba Al-Marri, Bekir S Ciftler, and Mohamed M Abdallah. Federated mimic learning for privacy preserving intrusion detection. In *2020 IEEE international black sea conference on communications and networking (BlackSeaCom)*, pages 1–6. IEEE, 2020.

[35] Segun I Popoola, Ruth Ande, Bamidele Adebisi, Guan Gui, Mohammad Hammoudeh, and Olamide Jogunola. Federated deep learning for zero-day botnet attack detection in iot-edge devices. *IEEE Internet of Things Journal*, 9(5): 3930–3944, 2021.

[36] Gustavo de Carvalho Bertoli, Lourenço Alves Pereira Junior, Osamu Saotome, and Aldri Luiz dos Santos. Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach. *Computers & Security*, 127:103106, 2023.

[37] Cristiano Antonio De Souza, Carlos Becker Westphall, Renato Bobsin Machado, João Bosco Mangueira Sobral, and Gustavo dos Santos Vieira. Hy-

brid approach to intrusion detection in fog-based iot environments. *Computer Networks*, 180:107417, 2020.

[38] Azam Rashid, Muhammad Jawaid Siddique, and Shahid Munir Ahmed. Machine and deep learning based comparative analysis using hybrid approaches for intrusion detection system. In *2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*, pages 1–9. IEEE, 2020.

[39] Hongwei Ding, Leiyang Chen, Liang Dong, Zhongwang Fu, and Xiaohui Cui. Imbalanced data classification: A knn and generative adversarial networks-based hybrid approach for intrusion detection. *Future Generation Computer Systems*, 131:240–254, 2022.

[40] Mohammad Reza Parsaei, Samaneh Miri Rostami, and Reza Javidan. A hybrid data mining approach for intrusion detection on imbalanced nsl-kdd dataset. *International Journal of Advanced Computer Science and Applications*, 7(6):20–25, 2016.

[41] Wei-Chao Lin, Shih-Wen Ke, and Chih-Fong Tsai. Cann: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems*, 78:13–21, 2015.

[42] Sunil Gautam, Azriel Henry, Mohd Zuhair, Mamoon Rashid, Abdul Rehman Javed, and Praveen Kumar Reddy Maddikunta. A composite approach of intrusion detection systems: hybrid rnn and correlation-based feature optimization. *Electronics*, 11(21):3529, 2022.

[43] Murat Emeç and Mehmet Hilal Özcanhan. A hybrid deep learning approach for intrusion detection in iot networks. *Advances in Electrical and Computer Engineering*, 22(1):3–12, 2022.

[44] Sandhya Ethala and Annapurani Kumarappan. A hybrid spider monkey and hierarchical particle swarm optimization approach for intrusion detection on internet of things. *Sensors*, 22(21):8566, 2022.

[45] Mavra Mehmood, Talha Javed, Jamel Nebhen, Sidra Abbas, Rabia Abid, Giridhar Reddy Bojja, and Muhammad Rizwan. A hybrid approach for network intrusion detection. *CMC-Comput. Mater. Contin*, 70:91–107, 2022.

[46] Kishor P Jadhav, Tripti Arjariya, and Mohit Gangwar. Hybrid-ids: An approach for intrusion detection system with hybrid feature extraction technique using supervised machine learning. *International Journal of Intelligent Systems and Applications in Engineering*, 11(5s):591–597, 2023.

[47] P Sanju. Enhancing intrusion detection in iot systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks. *Journal of Engineering Research*, page 100122, 2023.

[48] Abdullah Alsaedi, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar. Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. *Ieee Access*, 8:165130–165150, 2020.

[49] Saman Taghavi Zargar, James Joshi, and David Tipper. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys & tutorials*, 15(4):2046–2069, 2013.

[50] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Stefanos Gritzalis. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Communications Surveys & Tutorials*, 18 (1):184–208, 2015.

[51] Bander Ali Saleh Al-rimy, Mohd Aizaini Maarof, and Syed Zainudeen Mohd Shaid. Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions. *Computers & Security*, 74:144–166, 2018.

[52] Muna Al-Hawawreh, Frank Den Hartog, and Elena Sitnikova. Targeted ransomware: A new cyber threat to edge system of brownfield industrial internet of things. *IEEE Internet of Things Journal*, 6(4):7137–7151, 2019.

[53] Maede Zolanvari, Marcio A Teixeira, Lav Gupta, Khaled M Khan, and Raj Jain. Machine learning-based network vulnerability analysis of industrial internet of things. *IEEE Internet of Things Journal*, 6(4):6822–6834, 2019.

[54] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[55] Yujing Chen, Yue Ning, Martin Slawski, and Huzefa Rangwala. Asynchronous online federated learning for edge devices with non-iid data. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 15–24. IEEE, 2020.

[56] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[57] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

[58] Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data. *Google Research Blog*, 3, 2017.

[59] Sin Kit Lo, Qinghua Lu, Liming Zhu, Hye-young Paik, Xiwei Xu, and Chen Wang. Architectural patterns for the design of federated learning systems. *arXiv preprint arXiv:2101.02373*, 2021.

[60] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993.

[61] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., volume 2, pages II–II. IEEE, 2004.

[62] Zou Xia Wei Ruijue Zou Sophie Yan Wenbo, Tong Ling Nga Meric. Anomaly detection using advance machine/deep learning.

[63] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[64] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.