University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Cybersecurity

Securing 5G Networks with Federated Learning and GAN

Supervisor
Prof. Alessandro Brighente
University of Padova

Co-supervisor
Msc Tuomo Makkonen
Fraktal

Master Candidate
Rayyan Hassan

Student ID
2013051

Academic Year
2022-2023

"A journey will have pain and failure. It is not only the steps forward that we must accept. It is the stumbles. The trials. The knowledge that we will fail. That we will hurt those around us. But if we stop, if we accept the person we are when we fail, the journey ends. That failure becomes our destination."

— Life

Abstract

The threat landscape of the 5G network is quite vast due to the complexity of its architecture and its use of virtualized network functions. This landscape can be divided into two categories: Attacks against the Access point and Attacks against the Core. This thesis has been dedicated to analyzing the threats that plague the 5G network with a special focus on the access point. The architecture for the access point was simulated with a federated learning environment to not only secure the privacy of the user data but to also present a realistic scenario from which to perceive the 5G network. The main objective of the thesis was to secure the access point of the 5G network in this federated learning environment. This was accomplished by placing an Intrusion Detection System at the endpoint which would classify the data as either benign or malicious. The effectiveness of this model was checked by simulating a malicious user and conducting certain adversarial attacks to determine if the model could defend against them. The study was conducted by performing two specific attacks i.e Label-Flipping attack and Generative Adversarial Networks. The attacks were successful and revealed that a new system should be designed and developed that could be resilient against these types of attacks.

Contents

Listing of figures

Listing of tables

Listing of acronyms

5g ..... Fifth Generation

eMBB ..... Enhanced Mobile Broadband

uRLLC ..... ultra-reliable low latency communication

DOS ..... denial-of-service

DDOS ..... distributed denial-of-service

IoT ..... Internet of Things

ENISA ..... European Union Agency of Cybersecurity

GAN ..... Generated Adversarial Networks

FL ..... Federated Learning

ML ..... Machine Learning

DL ..... Deep Learning

IDS ..... intrusion detection systems

SDN ..... software-defined networking

NFV ..... network functions virtualization

MIMO ..... massive multiple-input and multiple-output

AI ..... Artificial Intelligence

ADS ..... Anomaly Detection System

RNN ..... Recurrent Neural Network

CNN ..... Convolutional Neural Network

RF ..... Random Forest

GRU ..... Gated recurrent unit

LSTM . . . . . long short-term memory

DNN . . . . . Deep neural network

DBN . . . . . Deep Belief Network

SDPN . . . . . stacked-deep polynomial network

MLP . . . . . Multilayer perceptron

ANN . . . . . Artificial neural network

NB . . . . . naive Bayes

HW-DBN . . . . . Hybrid Weighted Deep Belief Network

LF . . . . . label-flipping

SVM . . . . . Support vector machines

UE . . . . . User Equipment

FGSM . . . . . Fast Gradient Signed Method

JSMA . . . . . Jacobian Based Saliency Map Attack

SGD . . . . . Stochastic Gradient Descent

KL . . . . . Kullback–Leibler

MSE . . . . . Mean Squared Error

NGMN . . . . . Next Generation Mobile Networks

TCP . . . . . Transmission Control Protocol

FTP . . . . . File Transfer Protocol

XSS . . . . . cross-site scripting

SSH . . . . . Secure Shell Protocol

HTTPS . . . . . Hypertext Transfer Protocol Secure

HTTP . . . . . Hypertext Transfer Protocol

UDP . . . . . User Datagram Protocol

ICMP . . . . . Internet Control Message Protocol

IP . . . . . Internet Protocol

1 Introduction

1.1 5G Intro

The fifth generation (5G) and beyond networks are expected to enhance the user experience with regard to all kinds of communication, be it human to machine or machine to machine. It accomplishes this by providing lower latency and higher connectivity and capacity and due to their projected reliability it will enable industries to provide new and improved services that will result in a better quality of life experience. According to current consensus, 5G will allow users to benefit from Enhanced Mobile Broadband (eMBB) which will provide data rates in excess of 20 Gb per second which is much faster than current technologies [1]. Similarly, the ultra-reliable low latency communication (uRLLC), will lower the latency to approximately 1 ms, hence, improving the quality of communication between connected devices. This is one of the features that will ensure the rise in the usage of IoT devices.

The emergence of 5G leads credence to the fact that more security is required in order to keep the threats at bay. The 5G core is quite susceptible to external threats such as DOS, DDOS, port scans, and even specific adversarial machine learning techniques. The objective behind this thesis was to form a federated learning environment that would allow the simulation of either different 5G smartphones or IoT devices and to secure it. This was followed by the usage of a supervised machine-learning algorithm on a 5G Network Intrusion Detection Dataset to detect anomalies. However, like all machine learning models, there were certain vulnerabilities

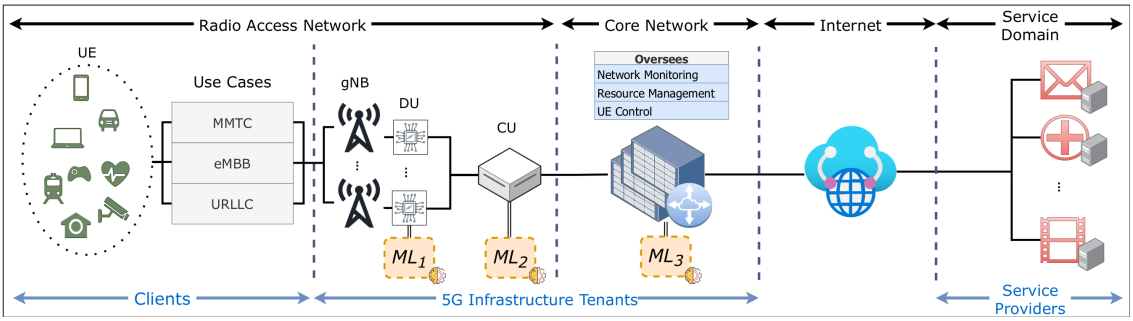


Figure 1.1: 5G Ecosystem Source [1].

that the attacker could exploit through the use of adversarial machine learning or Generated Adversarial Networks. The primary objective thus was to construct a robust and resilient environment for the functioning of the 5G network by protecting it from these external threats.

1.2 Background and Research Problem

Cybersecurity is a continuously evolving field and so are the threats which plague them. The objective behind this thesis is to improve a certain aspect of 5G security and this involved fully understanding the threat landscape of the 5G network. While there are multiple standardizations such as 3GPP and ENISA which have discussed this issue in detail, they are relatively old when considered from a research perspective. Many novel threats and their solutions have been found in recent years pertaining to 5G security and a robust yet resilient defense has yet to be found. Considering the taxonomy of 5G threats, it could be noticed that most of them target the access point and most devices have access to the network through their personal user equipment such as mobile phones. Therefore, the objective of this thesis was to reinforce the intrusion detection system for the 5G network against certain adversarial attacks. This framework was considered essential when considering the many ways in which the access network can be penetrated by an adversary connected to the network.

1.3 Novel solution

While none of the techniques discussed in this work by themselves are new to the scientific world they have been implemented in a unique way. Most adversarial attacks such as data poisoning attacks like label flipping and Generated Adversarial Networks (GAN) have been thoroughly researched and their potential to subvert the machine learning classifiers is known to

the research community. The problem however is that the threat landscape utilized in most of these attacks is quite unrealistic as the majority of these attacks assume a white box setting where the adversary already has access and knowledge of the training set and the classifier. In other scenarios they can gain this information by querying the system however in modern intrusion detection systems, there are usually only a limited number of queries that the attacker can do. Considering that the GAN is a neural network, it requires ample data to be properly trained which is simply not possible through repeated queries.

In this work, a federated learning environment has been proposed for the 5G environment which is quite realistic when taking into account the privacy concerns that users have with regard to their data. Now while the federated learning environment has been developed to test the privacy concerns that plague the 5G network, it has not been properly defended against adversarial machine learning. The research in this thesis shows the following:

- The vulnerability of the federated learning environment to the GAN model and a defense mechanism has also been designed to combat this vulnerability.
- Similarly, the model has also been shown to be vulnerable to data poisoning attacks due to which a framework for this threat has also been proposed.

To the best of our knowledge, no researcher has utilized this approach of utilizing GANs and label-flipping attacks in the federated learning environment.

1.4 Organization of the thesis

In this chapter, a brief introduction was provided regarding the 5G network and its security. Moreover, a concise summary regarding the scope and objective of the thesis was also given. The next chapters are organized as follows:

- In Chapter 2, the literature review regarding the relevant works will be conducted. The literature on this topic is quite vast, hence the focus shall be on the most significant works. In this chapter, the relevant threats induced by machine learning on the 5G system and the related papers regarding the different approaches utilized for intrusion detection frameworks will be discussed. Moreover, the various methodologies utilized to attack machine learning models such as label-flipping attacks and GANs will also be discussed.

- In Chapter 3, the system and threat models are examined. Considering that the objective is to secure the access network, it is imperative to understand the various threats that the access network faces. A detailed analysis of the working of these attacks will be presented followed by a discussion of the adversarial threat model.
- In Chapter 4, the methodology founded by the author will be discussed in detail. This will be a thorough explanation of the adversary's objective and the manner in which they accomplish it. It will provide details regarding the 5G environment, the algorithms utilized by the adversary, and lastly the framework of the defense that was implemented to prevent future attacks.
- In Chapter 5, the implementation of the aforementioned strategy is conducted on the 5G dataset. It explains the dataset in detail followed by the measures taken for pre-processing the data. Lastly, the performance metrics were defined and the results of the implementation were calculated and analyzed.
- In Chapter 6, the limitations of the work are discussed followed by providing future research directions and concluding the thesis.

2

Literature Review

In this chapter, the works related to the current thesis shall be analyzed. The literature with respect to intrusion detection systems in particular is quite vast with respect to network traffic and Internet of Things (IoT) devices. Similarly, the ideas behind adversarial attacks such as GANs and label-flipping attacks are also not unique proponents of this thesis and as such will be discussed and analyzed in detail.

2.1 Machine Learning threatens 5G

The increasing diversity in networking equipment, end-user devices, applications, and services in communication networks has made network operations more complex which in turn has necessitated the utilization of automation [3]. ML has emerged as a critical tool in recent years for automating the operations in the wireless network and this holds true for 5G as well. One of the challenges that arise with the usage of machine learning algorithms is security concerns. This is due to the reliance of ML models on data where the collection and dissemination of data can expose the network to certain vulnerabilities. Therefore, the very first step is to properly understand the threat landscape of the 5G network in relation to machine learning so as to understand and avert the risks associated with this strategy.

Considering the fact that ML approaches will be used throughout the network infrastructure, in order to reduce human involvement, it is paramount to understand the risks associated

with the technological enablers of 5G such as software-defined networking (SDN), network functions virtualization (NFV), massive multiple-input and multiple-output (MIMO) antennas, and diverse types of devices and services like the Internet of Things (IoT). This is because any integration with new technology always exposes previously unseen security vulnerabilities thus making it fundamental to conduct proper research in order to properly understand the threat landscape.

There have been ample surveys conducted on the threats represented by the utilization of machine learning in the 5G network as seen in the Table. There are many aspects to take into consideration when dealing with this framework such as the application of the technology in question, the two-way traffic between the server and the user equipment, and the inherent security of the device itself. The following surveys have taken these variables into account and also provided certain solutions and research directions which have been implemented in this thesis.

The high-level application areas which are susceptible to attacks from malicious parties are the following:

- 1) Infrastructure Management: Each of the application areas has its own special use case for ML. In Infrastructure Management, ML can be utilized to improve the deployment and management of the various components such as the radio antennas and the base stations. Similarly, it can also be used to decipher network traffic patterns and network performance so that the operators can recognize the high-capacity areas and the zones where upgrades might be necessary. Lastly, it could also assist in improving network availability and preventing network failures.

- 2) Network Operations: It is possible to utilize ML for automating and optimizing different network operations like performance monitoring and network configuration. Moreover, it can dynamically regulate the network parameters in order to enhance the network performance. With the help of AI, it can also recommend and perform remedial actions to fix network issues as well.

- 3) Service Orchestration and Management: ML can be used to interpret the data on performance, service usage, and user behavior to optimize service delivery and offerings based on the preferences of the user.

- 4) Assurance: ML can be used as an Anomaly Detection System (ADS) to mitigate and prevent security threats by analyzing the network traffic and detecting malicious access attempts to the system. Consequently, identifying the major security vulnerabilities to the access point and user equipment.

- 5) Security: ML can improve network security by enabling proactive threat detection and response. ML algorithms can analyze network traffic, system logs, and other data sources to identify patterns and anomalies that may indicate a security breach [3]. ML can also be used to identify and respond to advanced persistent threats, such as malware that may be present in the network over an extended period.

2.1.1 Threats Induced by ML

The STRIDE model is a good indicator to study the threats induced by Machine Learning algorithms in the 5G network. These threats are classified as spoofing, tampering, repudiation, disclosure of information, DoS, and elevation of privileges. The operating principle behind a classification ML model is the utilization of raw input and training it according to a certain algorithm to produce intelligent actionable output.

It is mathematically denoted as a function that maps an input vector representing the features of the data point to a discrete output value representing the predicted class label. Let X be the input vector, Y be the output variable representing the class label, and let f denote the function that maps X to Y. Then, we can represent a machine learning classifier as:

$$Y = f(X)$$

In other words, given an input vector X, the function f outputs a predicted class label Y. The function f is typically learned from a training dataset using a machine learning algorithm such as logistic regression, decision trees, or support vector machines.

There is a myriad of attacks against machine learning models that the adversary can accomplish. The adversary can have different abilities depending on the threat landscape and the privileges that they have access to as such they can perform a wide array of actions depending on those circumstances. They can submit malicious data to attack the network or the ML system or attempt to intercept, modify, and eavesdrop on transmitted data. Moreover, the adversary can also attempt to gain access to the ML model through model extraction or model inversion techniques.

- Influence: This describes whether the attack impacts the training process through model poisoning as seen in [5] or if it tampers with the learning outcomes to evade analysis.

- Specificity: This attribute refers to whether the objective behind the attack is to cause misclassifications or if it is indiscriminate and affects the overall performance and reliability of the model [6].
- Security Violation: The third attribute is the adversary's security goal. This can be a violation of integrity, availability, or privacy.
- Frequency: This describes the occurrence of the attack and whether it happens iteratively.
- Knowledge: This refers to the amount of information the adversary has on the target system. In white-box attacks, the adversary has access to the internal workings of the machine learning system, while in black-box attacks, the adversary only knows the inputs and outputs.
- Falsification: This describes whether the objective is to produce false positives or false negatives in the ML model.

2.2 Anomaly Detection Frameworks

There have been many works done regarding 5G security and each author has attempted to secure the system against some specific threat. This is due to the wide range of threats that the 5G system is susceptible to as seen in the 3GPP specifications [7] and the guidelines provided by the ENISA [2]. However, new threats are emerging, and as such the 5G network is still being extensively researched due to the growing use of this network technology.

While the domain for intrusion detection and anomaly detection for IoT has been extensively studied by researchers, the same cannot be said for the 5G network. The 3GPP has set the standard for 5g security and has defined the threats that the network should be protected against [7]. These threats include both passive and active attacks with a special focus on the MITRE attack framework.

The rule-based algorithm for anomaly detection functioned by creating rules that defined the benign expected behavior and flagged any deviations from those rules as potential anomalies. The world however is evolving from the usage of these algorithms that manually updated the signatures from previous attacks. This is because with the 5G network, the connectivity rates will increase exponentially and the latency will also be reduced as such while rule-based algorithms might work for certain threats, it will not be possible to base the entire system on it.

2.2.1 Deep Learning Solutions

Deep learning is a widely used data mining technique that utilizes neural networks to model abstract concepts. There are a wide array of applications for this algorithm in the following fields: speech recognition, image classification, natural language processing, and even semantic analysis [8]. These algorithms work by identifying the correlations between large datasets and are widely used for classification purposes. This class of machine learning algorithms utilizes multi-layered neural networks to solve complex equations through black-box methods [9]. These layers are quite useful in extracting high-level features from raw input as each underlying layer is responsible for a different feature. The idea behind using Deep Learning for IDS revolves around detecting abnormal traffic or abnormal activities by the users that could point towards manipulation of the network [10].

It is possible to employ deep learning for intrusion detection using supervised, unsupervised, or even semi-supervised approaches [10]. Machine learning approaches come with their drawbacks such as overfitting and under-fitting. Over-fitting occurs when the model is unable to generalize and only fits the training data. This causes the model to have inaccurate results on data that it has not yet seen. Under-fitting is the counterpart of overfitting and usually occurs because the model did not have enough data to learn the patterns in the training data and is also unable to generalize to the new data.

2.2.2 Convolution Neural Network

Anomaly detection is usually modeled as a classification model using supervised learning and researchers have previously used this approach in combination with a Convolution Neural Network architecture in order to optimize the accuracy of the model [22]. The dataset used was the CICIDS2018[23] which is an Intrusion Detection Dataset. The data was transformed into 100x100x3 images so that the CNN could be utilized thus turning it into a Computer vision approach to Intrusion Detection. The model had a precision score of 98.2% and a recall score of 98.1% [22].

The random forest algorithm was also utilized on the aforementioned dataset and provided an accuracy of 99.99% on benign traffic however the accuracy dropped to 82 % when classifying anomalous traffic [24]. Although the three-layered neural network detected anomalous traffic with an accuracy of 99.3%, the model did not generalize well and was thus overfitted.

Author	Dataset	Algorithm	Findings
2017 [11]	RedIRIS	RNN and CNN	In this work, the authors have improvised the existing deep learning algorithms by modifying the hidden layers.
2017[12]	KDD99	GRU and RF	Minimizing the loss function has helped the researchers to achieve better results.
2018[13]	UNSW-NB15	LSTM and RNN	In this work, feature normalization and conversion of categorical features to numeric values have helped them to generate improvised results
2018 [14]	NSL-KDD	DNN	In this work, SGD was used to minimize the loss function of DNN
2019 [15]	CICIDS2017	MLP, 1d-CNN, LSTM, and CNN+LSTM	In this work, researchers have balanced the dataset by performing data processing in which they have duplicated the records
2019 [16]	NSL-KDD	DBN	The neural network was optimized by assigning a cost function to each layer of the model
2019[17]	NSL-KDD	SDPN	The SMO algorithm is used for feature selection
2020[18]	NSL-KDD	RF	Weka tool was used for evaluation purposes
2021[19]	KDD99, NSL-KDD	ANN	The stack-based feature selection technique has been proposed to optimize the computation
2021 [20]	Bot-IoT	RF, NB, and MLP	In this work, a hierarchical approach was used for intrusion detection.
2021 [21]	CICIDS2017	HW-DBN	In this work, the low-frequency attack was detected

Table 2.1: Analysis of various existing frameworks

Sr.No	Algorithm	Accuracy (%)
1	Random Forest [18]	98.73
2	Recurrent neural network+ convolutional neural network [11]	96.12
3	Gated recurrent neural networks [12]	98.91
4	Bidirectional long short-term memory recurrent neural network [13]	95.72
5	Distributed deep model [14]	99.2
6	Convolutional neural network+ long short-term memory [15]	97.16
7	Spider monkey optimization+ stacked-deep polynomial network [17]	99.02
8	Artificial neural network [16]	98.56
9	Hybrid Weighted Deep Belief Network [19]	99.38

Table 2.2: Results from various Deep Learning frameworks.

2.2.3 Auto encoders

The auto-encoders are an unsupervised approach to deep learning and work by reducing the dimensionality of the feature space. The model has the same number of layers as its feature vectors and also has hidden layers to help with the dimensionality problem. The idea behind this algorithm is to train both an encoder and a decoder where the encoder is responsible for learning the characteristics and representation of the data by transforming the input into a lower-dimensional representation. This is followed by passing the results through the decoder which learns to reconstruct the original input from the compressed representation. This is a more unique approach to anomaly detection as it does not utilize supervised learning and as such has no need for labels. The idea behind this approach for anomaly detection is to train the model to recognize the benign data so that if it does not recognize the representation then it classifies it as an anomaly. While it is possible for anomalies to not be malicious, most researchers are of the consensus that it is better to classify the anomaly as malicious rather than the alternative as it leads to fewer security issues [25].

There are several authors who have worked on using auto-encoders for anomaly detection. The best results were achieved by [26] who employed an ensemble learning approach based on the stacking method with the self-attention mechanism. This helped track the ling term depen-

dencies in the data samples. Thus this stacking ensemble learning algorithm was composed of the base learner and the auto-encoder where the auto-encoder reduced the dimensionality of the dataset and the stacking method integrated the detection results of sample embedding and the base learner. The algorithm had a high precision score of 99.6% and a recall score of 99.7%.

2.3 Label Flipping Attacks

A label-flipping attack as the name suggests functions by changing the target class in the dataset to get the classification algorithm to misclassify the input [27]. There have been various techniques that were introduced in the literature that relates to this attack as each technique either uses a distinct algorithm or a different threat landscape or in certain cases both. This data poisoning attack compromises the integrity of the model and is commonly used by attackers to evade intrusion detection systems [28]. The following paragraph details the recent findings of the scientific community with regard to these types of attacks.

It is possible to conduct an LF attack following the optimization formulation that focuses on optimizing the loss function of the model as proposed by Paudice [29]. However, this strategy was limited to white box attacks which while useful for comprehending the worst-case scenario did not truly represent a real-world scenario. Another approach was found by Xiao [30] where the model was capable of performing an attack on the SVM model however this was again limited in the sense that it could only work if the defender was using the SVM as a classifier. Moreover, it did not present a suitable scenario where the attacker could easily gain access to the dataset. Similarly, there were many different poisoning attacks that targeted the anomaly detection system [31][32][33]. There have been other approaches as well like neural networks [34][35][36][37], unsupervised machine learning [38], dimensionality reduction [39], linear classifiers [40] and regression[41] but all of these adversarial attacks follow the traditional setting of the attacker having access to the network and also having the capacity to manipulate the data before the training phase.

The best-related work was the one that the thesis took inspiration from which was conducted by Tolpegin [42]. The idea was to perform data poisoning attacks against a federated learning system. This was a very feasible and realistic approach due to the fact that the attacker can manipulate the training data on their own device quite easily. The attack is conducted on the client's UE which causes the local model to send poisoned model parameters to the global

model. The global model is then trained with these poisoned parameters which cause it to lose its accuracy. While the same objective is accomplished using the same techniques as the aforementioned modes, the difference lies in the approach that was undertaken to conduct the attack.

2.4 Adversarial Machine Learning

Similar to the data poisoning attack, it is possible to generate malicious samples to evade the detection of the classification machine learning algorithm. However, unlike the aforementioned attack, the manner in which a traditional GAN attack operates is that it generates synthetic data which replicates the data distribution of the input that is fed to it [43]. This synthetic data is malicious however statistically it appears to be benign to the machine learning model due to which it misclassifies this input [44]. This is a very strong adversarial attack however it requires that the adversary have access to the training set so that they may train their GAN model.

There are many white-box adversarial attacks in research as they postulate that the attackers have access to the complete dataset. These attackers generate the adversarial data through FGSM and JSMA methods [45] however these methodologies are not very realistic due to the difficulty of gaining access to the complete dataset. Yang has proposed the usage of the GANs to elude the deep learning-based network IDS model by utilizing the zeroth-order optimization to attack the IDS [46]. The problem with this approach was that the discriminator was not reliable as the attacker just used one of the common IDS datasets to form the adversarial examples. One of the approaches that researchers have used for the black box scenario is that they have assumed that they can query the intrusion detection system and through that make their own dataset which they can then use to train their GAN model [47]. While this approach provides great results as the adversary can essentially create an entire dataset to train their GAN model, the problem is that in realistic scenarios, it is not possible to query the model repeatedly without drawing attention from the defender.

3

System and Threat Models

The Next Generation Mobile Networks (NGMN) [48] provides suggestions for the improvement and enhancement of the 5G network by taking into consideration the threats present in the current network architectures. The lack of security measures in the previous generation networks like 4G and 3G was a great concern when developing 5G. Therefore, since the very conception of this new-generation technology, research concerning security practices has been a prime force due to the wide array of cyber threats that can be used against the network. The European Union Agency of Cybersecurity (ENISA) has defined the taxonomy of cyber-threats [2] that the 5G system can be vulnerable to and this can be seen in 3.1.

3.1 Security Challenges in the Access Network

Secure network access ensures that the user can access the network and its services while being protected from external threats. This implies that the user is secure from both malicious network activities and also from unauthorized access attempts. In the modern era, there is a multitude of threats that can be effectively utilized against the network. These threats are not restricted to any one component of the network but rather encapsulate the entirety of the network which includes the Network Core, User Equipment, Access points, and also the Cloud which hosts the services [3]. 5G networks face a greater threat as compared to the technologies of the previous eras because it utilizes a wide array of access technologies in order to achieve better coverage, throughput, and lower latencies. This can be seen in 3.2.

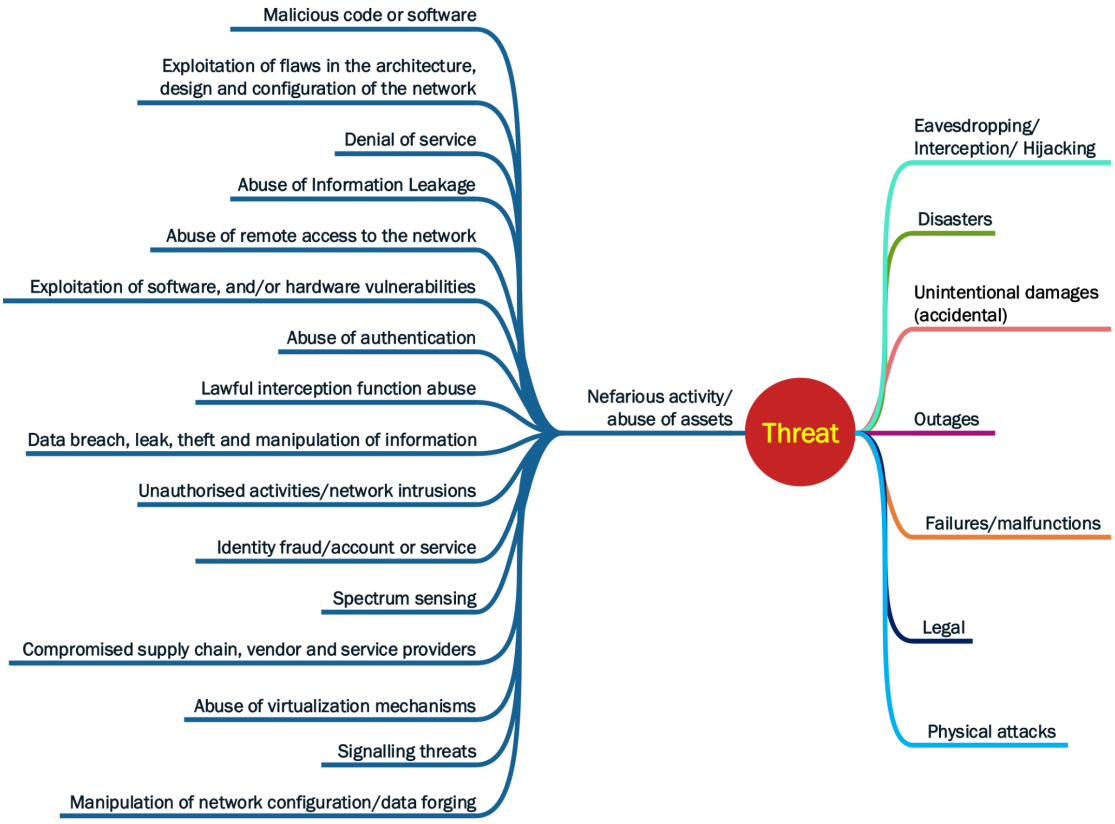


Figure 3.1: Taxonomy of Threats in the 5G Network Source [2]

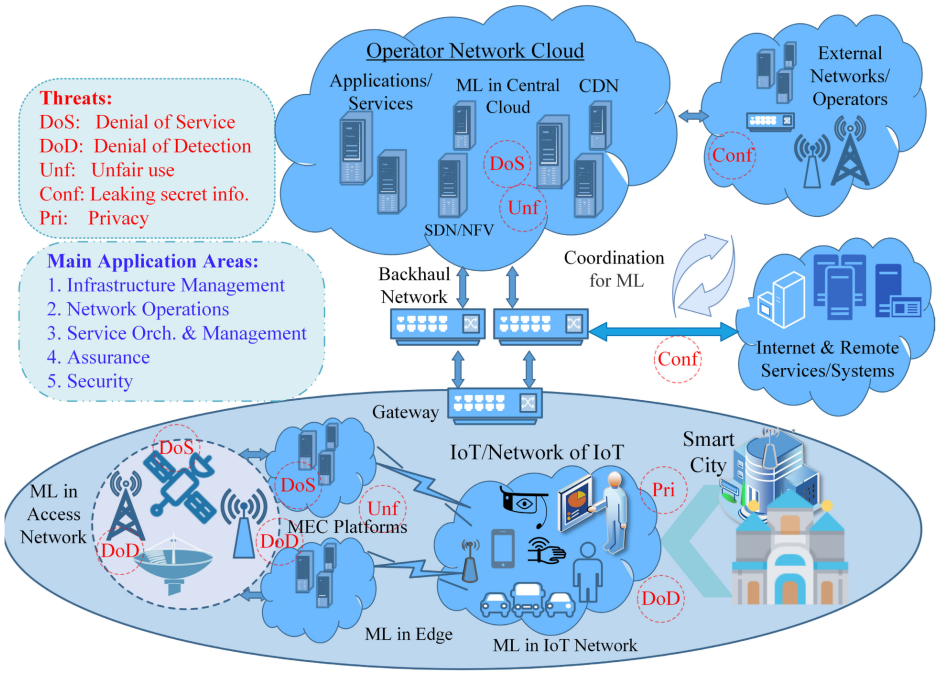


Figure 3.2: 5G Threat Landscape Source: [3]

Hence, it is imperative that the access point be secured as it is the first line of defense for user security. One of the key challenges and concerns for the defense is the possibility of jamming attacks and Denial of Service attacks because the adversary can manipulate the rate of data transmission and reception due to the increased number of devices that the 5G network can support [1]. Moreover, this threat can be further exacerbated by flooding the servers with excessive traffic due to the high likelihood of the attacker gaining access to multiple nodes that they can manipulate simultaneously. This can result in slower response times and accessibility issues for the users.

Therefore, 5G must improve the resilience of the system against jamming attacks on radio channels and signals. These types of signaling traffic and attacks should be detected and prevented before the network is compromised. Another aspect that can improve the security of the access network can be accomplished by securing the small cell nodes since their wide geographic distribution and ease of accessibility make them a prime target for the attacker.

3.1.1 DoS Attacks

Denial of Service (DoS) and Distributed Denial of Service(DDoS) attacks are one of the key challenges that could impact the performance of the 5G network due to the large number of connected devices. The objective of the DoS attack is to exhaust the resources of the network's operator which would in turn directly affect the subscribers to the network [3]. With DDoS however, the aim of the attacker is to exhaust both the resources of the users and devices themselves which would not only directly impact both the subscribers and their devices but would also affect the operational capability of the network operator. Furthermore, it can also lead to the utilization of hijacked devices to launch attacks against the network's architecture.

In order to carry out the DoS attacks on the 5G network infrastructure, the malicious users would likely need to focus on the resources related to the connectivity and the bandwidth since these are critical for meeting the required levels of service. To disrupt 5G services, the DoS attacks can exploit a wide array of vulnerabilities across the signaling, management, and user planes of the 5G architecture and their supporting systems. Specifically, the attacker can focus on the following [25]:

- 1. The signaling plane can be targeted as it would affect the ability of the users to be authenticated and connected to the network. Moreover, it also assigns bandwidth to the subscribers and thus would limit their mobility.
- 2. The user plane can also be targeted as it facilitates two-way communication between the servers and the connected devices.
- 3. The management plane configures the network elements that support the user and signaling planes.
- 4. The radio resources enable access to the 5G network for the subscribed devices.
- 5. The support systems handle the administrative-related tasks for the users like billing
- 6. The logical and physical resources which support the network clouds like servers, storage, and virtual machines.

Meanwhile, DoS attacks against the user equipment do so with the objective that they wish to exhaust the logical and physical resources of the user such as the memory, battery, processing units, radios, sensors, operating systems, applications, configuration data, and user data. The attacker can deter the user's ability to enter and utilize the network resources by compromising these resources which can also lead to cascading effects on the wider network infrastructure. In contrast, however, DoS and DDoS attacks against the critical architecture components of a certain network like the following: energy, health, transportation, and telecommunication can have long-standing devastating effects on the entire community. It is possible to orchestrate an attack on these systems from either a large number of geographically dispersed machines or certain compromised IoT devices. This showcases the need for a robust and resilient security strategy that can enhance the network's capability against these threats.

Therefore, the vision for 5G systems is to present highly secure and robust services which have the tendency to persevere against a wide array of cyber threats while also ensuring privacy and security for the subscribers and their devices. This necessitates a revamped approach to security which is comprehensive and encapsulates both the network infrastructure and the end devices [22]. Moreover, it should also be continuously evolving so that it can adapt to new threats and vulnerabilities. It is only through this evolution that 5G will become a truly transformative technology that can safely be used for a wide range of applications. The idea behind this is to fully implement security by design principles at the onset of the development of the 5G network to ensure that the proper security measures are maintained throughout the system.

3.1.2 Port Scanning

Port scanning is a type of reconnaissance attack that attackers use to identify and target open ports on a network. These scans are typically performed before carrying out an attack on the vulnerable host. These scans are executed by sending traffic to a series of different ports on the target system and then analyzing the response sent back by the system. This response allows the attackers to determine whether the ports are open and to check which services they are currently running. It is also possible to determine the operating system of the target from the response of the system [49]. Port scanning can be a significant threat to the 5G network because of the high traffic that the system will need to manage. There are different types of port scanning techniques that the adversary uses and while they all have the same objective, they utilize different techniques to accomplish it. The most common difference between the

various types of port scans is that the message requests contain different flags and protocols [48]. A network architecture generally has a multitude of devices connected to it and as such the different ports of the network are each responsible for providing their own unique service to the device. A single UE device can have 65546 ports which are further divided into three categories i.e the well-known ports (0 - 1023), the registered ports (1024 - 49151), and private ports (59152 -65535) [49].

Adversaries utilize port scanning attacks to gain information about the network in order to identify its vulnerabilities so that they can perform the ideal attack on the system. A port scan is conducted by sending a message to the port and waiting for the response which is generally the port's status. The port scans can be classified as shown in the figure. The most common type of port scan is the Transmission Control Protocol (TCP) based port scan due to it providing the adversary with more information as it is connection-oriented. This scan sends a request for a three-way TCP handshake which makes it difficult to detect. The stealth scan functions by transmitting SYN, TCP, FIN, or other stealth flags to the network ports and analyzing whether or not the ports are accepting connections. These scans can inform the attacker about the services that are running on the target system as certain services like HTTP traffic or HTTPS traffic always use the same port for communicating. Socket Secure (SOCKS) on the other hand is an internet protocol that allows users to securely communicate over the network with the servers. This internet protocol acts as a proxy between the server and the client to keep their communication secure. The SOCKS port scan however allows the attacker to conceal their location and identity when sending requests to the ports. Bounce scans utilize is a third-party system to send requests to the target system. This works by changing the source IP address by spoofing and then sending the message to the network. These scans use one of the features of the File Transfer Protocol (FTP) called "Bounce Attack" to mask the scanning requests. This is easily accomplished due to the fact that it is not a vulnerability of the FTP protocol but rather a misconfiguration of the FTP server which allows the server to be used as a proxy when conducting the scan. The UDP port scan is connection-less and as such while it can be observed, it does not provide adequate information to the attacker.

In conclusion, port scanning attacks pose a significant threat to 5G systems, which are designed to support massive amounts of traffic and connectivity. Port scanning attacks can be used by attackers to gain information about a target system, identify potential vulnerabilities, and plan further attacks. To defend against port scanning attacks, network operators and secu-

rity teams can use a range of mitigation strategies, including firewalls, intrusion detection and prevention systems, network segmentation, and threat intelligence. By implementing these measures, 5G networks can be made more secure and resilient against port scanning attacks and other forms of cyber threats.

3.1.3 Traffic Monitoring

Observing network traffic is crucial for the detection and prevention of security breaches and it has become even more important due to the diverse new technologies and services provided in 5G. The Intrusion Detection System and Intrusion Prevention Systems are mainly utilized for the monitoring of network traffic, the identification of security threats, and the application of countermeasures for the protection of the network.

3.2 Intrusion Detection System

There are numerous deployment options for the Intrusion detection system in the 5G Network depending on the particular use case. These particular use cases are defined by two attributes: specific requirements of the network and the types of attack that the system should detect.

Network Edge: The first option is to install the IDS at the Network edge so that it can detect the traffic entering and exiting the network. This is the approach most suited for the detection and prevention of attacks that originate from outside the network i.e a malicious actor. Some common attacks that target this domain of the network are DoS attacks, Distributed Denial of Service (DDoS) attacks, malware downloads, and phishing attempts.

Endpoints: Another option is to deploy the IDS on the endpoints of the particular devices that utilize the network such as laptops, smartphones, and IoT devices so that it is possible to discover and avert the attacks that target the devices of the user. Some of these attacks include malware infections, data exfiltration, and privilege escalation attempts.

Network Core: The IDS can also be employed at the core of the network so that it may detect traffic between the different modules of the core and prevent attacks that target the architecture of the core. These attacks include protocol attacks, routing attacks, or traffic hijacking.

Cloud or Data Center: It is also possible to deploy the IDS in the cloud or data center where the network services are hosted. This approach is useful in monitoring and preventing attacks that target the network infrastructure and services. These attacks include SQL injection attacks, cross-site scripting (XSS) attacks, or unauthorized access attempts.

The positioning of the IDS depends primarily on the objective that it needs to accomplish. In the case of this thesis, we are primarily concerned with defending the access points and the end devices themselves from malicious actors. Hence, the IDS should be designed in a manner such that it provides comprehensive coverage and effective threat detection and prevention of both the Endpoints and the Network Edge.

3.3 Threat Model

The Federated learning environment developed in this thesis consists of a central server that acts as a global model and a set of nodes that are decentralized and represent the client devices just as in a regular FL system. For the scope of this thesis, the adversary is considered to be an entity that intends to infiltrate the system either via a direct poison attack such as a label flipping attack, or a model poisoning attack through the implementation of GAN.

The threat model is based on the 5G federated learning environment. As such it will utilize the general characteristics of 5G networking. The attacker has access to one or more clients i.e the UE and as such operates as a client within the 5G network where there already exist numerous clients. The attacker has the ability to use their device to interact with the 5G network interface and send malicious data to the global model. In actuality, the 5G interface will not actually receive malicious data but rather it will receive the statistical results of the local model being trained on the malicious data. This in turn will poison the model and lower its overall accuracy.

In this scenario, a subset of the total participants is considered to be malicious. Let K be the compromised clients and N be the total number of clients. The malicious clients are incentivized to poison the global model M. The assumption is made that the malicious users have full control over their user equipment and can thus freely manipulate the data on their devices. It is further assumed that the global model is not initially compromised by outside sources.

The global model is a black box for the adversary and as such, they do not have any information regarding the inner working of the model. Considering that the adversary has complete control over his own UE, this would indicate that the attacker has complete access to all the data stored in it. The first assumption in order to conduct the LF attack is that the adversary has knowledge of the feature space F, and can also manipulate the training data on their device. Due to these assumptions, an LF attack becomes possible.

3.3.1 Label Flipping attack

The malicious users intend to corrupt a certain percentage $m\%$ of the total local data D_i available to them. The adversary can modify the dataset D_i by changing the source class to a target class from C which represents the total classes. This is mathematically denoted as $c_i \rightarrow c_j$. Therefore, in a binary classification problem, they can switch the class from benign to malicious or vice versa. The objective of the attacker is to reduce the overall accuracy of the global model M.It is possible to conduct this black box attack as the attacker does not require any knowledge of the architecture of the global model, the data distribution of the global model, the optimizer, etc.

3.3.2 GAN attack

The adversary controls K poisonous node i.e the User Equipment and can use these nodes to conduct the attack. The adversary has complete control over his UE, however, they are not aware of the total nodes,N, in the system. The model will take the client updates from all the nodes including the malicious user who controls the poisonous nodes. The malicious users will use the locally available data D_i to create synthetic data by using GANs. They can accomplish this as they are aware of the Data distribution on their own device. They can then use these synthetic examples to attack the global model.

4 Methodology

The initial idea was to consider the project as three distinct entities and then to join them together as one. The primary step was to construct a black box poison attack which in this case was a label-flipping attack and check whether it resulted in a drop in accuracy. This was followed by designing a system that would alert the system and prevent this attack. The second entity was the development of a Generative Adversarial Network model that would insert noise in the training data to confuse the model and cause it to misclassify the samples. The idea to prevent this attack was to develop a similar GAN model and train the model to recognize the noise so that it would not misinterpret this kind of attack. The last step was to construct the environment where the 5G network would be used. For practical purposes, the best environment was the federated learning one as it would not only improve the model due to the new data that it would receive from the clients but also maintain the privacy of the users.

4.1 Federated Learning Environment

The Federated Learning environment enables the training of a machine learning algorithm without having access to any private user data rather individual participants share the statistical information such as model parameters which are used instead. A good example of this type of environment is the Deep Neural Network which consists of multiple layers of nodes. Each node corresponds to a certain set of parameters and receives input from either the previous layer or the original training data. This is then followed by the node applying certain mathe-

matical functions to the input and sending it to the next layer. The final layer is responsible for generating the predictive result.

4.1.1 Architecture

In a traditional DNN model, there exists a training set T where $T = (x_0, x_1, x_2 ..... x_n)$ and $n \in N$. The model also has a specific loss function L and each data sample $x_i \in T$ has a set of features $f_i \in F$ and a class label $c_i \in C$. The values F and C represent the entirety of the Feature Space and Class values respectively. Furthermore, in a classification scenario, the final layer of the DNN shall contain nodes equal to the total class values, C, where each node will correspond to a subsequent class. The loss of the DNN given the total parameters θ is denoted by:

$$L = \frac{1}{n} \sum_{i}^{n} L(\theta, x_i)$$

When the DNN model is given f_i with model parameters θ as input then the model will output a set of probabilities p_i. The predicted probability $p_{c,i} \in p_i$ represents the likelihood of the given sample x_i having class $c \in C$. The probability $p_{c,i}$ is calculated in the final layer of the neural network and each node outputs the predicted probability for the class associated with it. The results for a specific sample passed through a model M with parameters θ would generally be:

$$M_\theta(x_i) = argmax_{c \in C} p_{c,i}$$

The architecture for the environment can be seen in 4.1.

4.1.2 Model Parameters

While there are different loss functions that can be implemented with the DNN, we are mostly concerned with the classification and as such will use the cross-entropy loss function which is mathematically computed through the following equation:

$$L(\theta, x_i) = -\sum_{c \in C} y_{c,i} \log(p_{c,i})$$

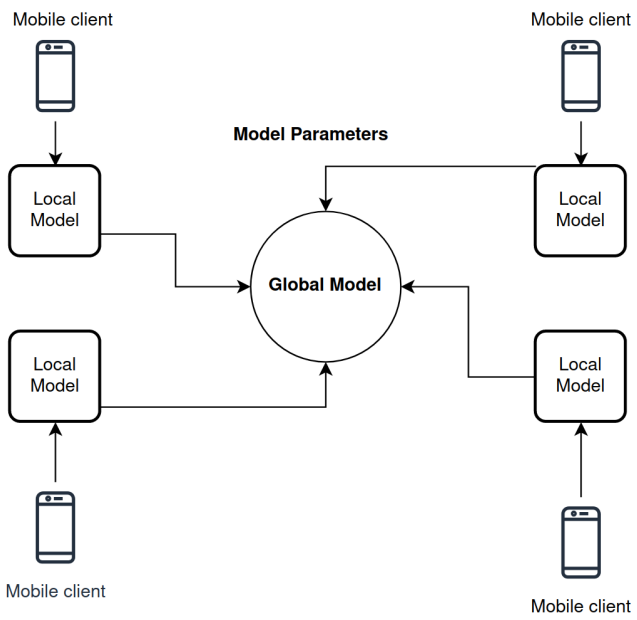


Figure 4.1: FL environment.

The objective of the DNN model is to reduce the loss function L to improve the model's accuracy. This is usually accomplished through the use of different optimizers such as ADAM or Stochastic Gradient Descent (SGD). These iterative processes work through the mini-batch strategy where at each step the optimizer selects a batch of samples $B \subset T$ and computes the gradient of the loss function.

$$g_B = \frac{1}{|B|} \sum_{x \in B} \nabla_\theta L(\theta, x)$$

As the objective of the model is to reduce the loss function which can be accomplished by reducing the gradient, therefore the parameters θ are updated in the direction opposite to the calculated value. The mini-batch strategy works by evenly dividing the training set into predefined batches so that no samples occur in multiple batches. The optimizer is then applied a set number of times on each of these batches where one iteration is referred to as one epoch.

4.1.3 Global Model

In Federated Learning environments, there is a dynamic allocation of the training dataset T as the global aggregator does not have access to the complete dataset rather the clients K each hold their own private dataset $[T_1, T_2, T_3 ..... T_n]$. The clients execute the optimizer on their own UE and then upload the final updated model parameters to the global model which acts as the aggregator. Therefore, the model is first trained locally before being sent to the global model for further updates. At the global training round r, a subset of the participants $k \subset K$ is chosen based on availability. Hence, each available participant k locally trains the model on their data T_i and sends the updated model parameters to the global model. The global model then aggregates the parameters θ:

$$\theta_g = \frac{1}{k} \sum_i \theta_{r,i}$$

This process is repeated for a set number of predefined rounds R after which the model M finishes the training process and has its final updated parameters θ_R

4.1.4 Adversarial Capabilities

The adversary views the system as a black box however they have access to a certain percentage of the total clients. They can thus influence the system with their user equipment and also have knowledge and access to their local dataset.

4.2 GAN model

The security of the Intrusion Detection system is questionable when faced with adversarial attacks as the classification algorithm is usually based on supervised machine learning. With label poisoning, the attacker attempts to change the classification class of the input data in order to decrease the accuracy of the model, however, there are other methods for the attacker to conduct adversarial attacks as well. One of the more popular methods is the usage of GAN models where the attacker generates adversarial examples through the use of noise. The attacker obscures the data with noise which causes the model to misclassify the query. Therefore, while the poison-based model directly attacked the system, the GAN model is a more passive attack as the attacker simply understands the patterns that the model has learned and obscures them with the help of noise. Thus, the objectives behind this model are two-fold. The primary ob-

jective behind this model is to protect it from the GAN attack as most supervised classifying algorithms are susceptible to it. The second objective is to enhance the dataset through the usage of these GAN examples in order to improve both the data volume and the classification effect.

4.2.1 Adversarial Model Design

The attacker constructed a Conditional Generative Adversarial Network model to accomplish the task. The architecture for this framework is the proposed CGAN model which consists of the following components i.e. Generator network, Discriminator network, and the black box classifier. The idea is to take the malicious traffic and use the GAN to generate adversarial traffic.

The network design is based on two feed-forward neural networks which are referred to as Generator and Discriminator respectively. These two networks are then trained in an adversarial way in order to generate data. The generator is constructed by taking the input which is the random noise and labels and acts as a nonlinear mapping function as defined below:

$$G : (Z\ x\ Y) \rightarrow X$$

The noise Z is generated randomly through $N(\mu, \sigma)$. We assume that the attacker has access to the class pairs i.e

$$[(x_0, y_0), (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)]\ ,\ \ n \in [1, N]$$

Thus, the generator works by taking samples from the prior distribution $p_z(z)$ where $z \in Z$ generates adversarial samples. These samples should be approximately close to the original class pairs. The objective of the generator is thus to learn the data distribution $p(x, y)$ and then to produce samples x which are similar enough to the class pairs in order to confuse the machine learning models.

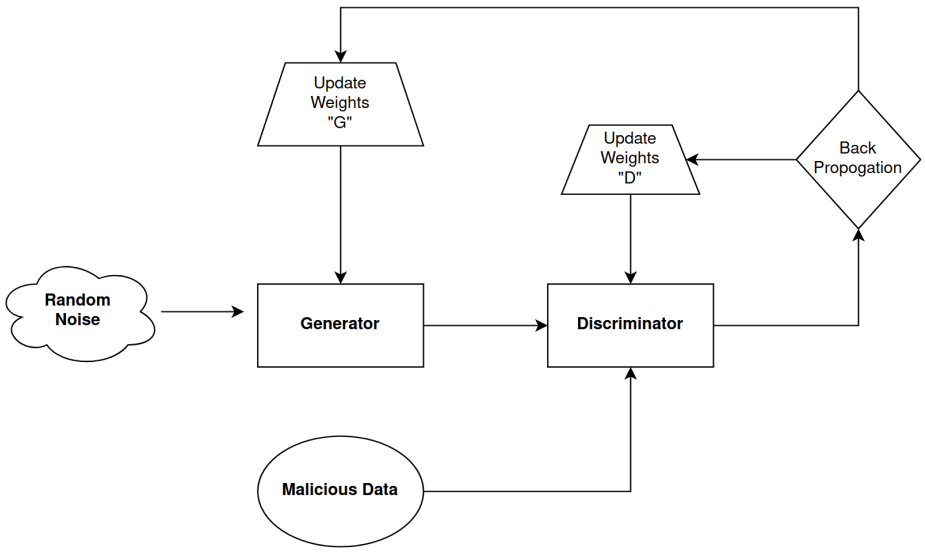


Figure 4.2: GAN architecture.

The discriminator is also a feed-forward neural network but takes the generated samples x and the class y as input. It then maps the input to either 0 or 1 i.e whether the input samples are real or generated from the distribution $p_z(z)$:

$$D : (X \times \ Y) \rightarrow [0, 1]$$

The architecture for the model can be seen in 4.2

4.2.2 Training the Model

The two networks are trained simultaneously by competing in a two-player min-max game where the objective of the generator is to confuse the discriminator and the goal of the discriminator is to properly classify the output. Thus this value function is defined as:

$$min_G \ \ max_D \ V(D, G) = E_D + E_G \qquad (1)$$

The expectation of the generator is calculated in the following way:

$$E_G = \varepsilon_{z\sim p_z(z), y\sim p(y)}[\log\,(1 - D(G(z,y) \mid y))]$$

Similarly, the expectation calculated for the discriminator is:

$$E_D = \varepsilon_{x,y\sim p_data(x,y)}[\log\ D(x \mid y]$$

Now through simple substitution, the value becomes

$$min_G\ \ max_D\ V(D,G)$$

$$= \varepsilon_{z\sim p_z(z), y\sim p(y)}[\log\,(1 - D(G(z,y) \mid y))] + \varepsilon_{x,y\sim p_data(x,y)}[\log\ D(x \mid y)] \qquad (2)$$

In the above equation, the generator is conditioned over the data distribution and the class labels as it is the objective of the generator to fully understand and learn the data distribution. The discriminator however is conditioned over the data distribution of x, y as it utilizes the original data in order to distinguish it from the generated data. The SGD is the optimizer used in this model to lower the loss of the model however ADAM could also be used. The training data is split into smaller subsets called mini-batches. Thus rather than computing the gradients of the loss function with respect to the model parameters on the entire dataset, the gradients are computed on each mini-batch separately. The parameters are then updated after taking the average gradient across all the mini-batches. We repeat this process for multiple epochs. For this case, the following loss function is used as it is based on the combined loss of both the generator and the discriminator as can be seen from equation (2)

$$J(\theta)_D = -\frac{1}{2m}(\sum_{i=1}^{m} \log D(x_i \mid y_i)$$

$$+ \sum_{i=1}^{m} \log(1 - D(G(z_i, y_i) \mid y_i))) \qquad (3)$$

$$J(\theta)_G = -\frac{1}{m}\sum_{i=1}^{m} \log D(G(z_i, y_i) \mid y_i) \qquad (4)$$

The KL divergence is a measure of how different two probability distributions are. The KL divergence is calculated between the generated dataset and the original dataset as this is the

metric we utilize to check the success of this model. In this case, the KL divergence is used to evaluate how well the generated data matches the real data. It is calculated in the following way:

$$KL(Distribution(P_{data})||Distribution(P_z))$$

$$= p_{data}(x)\log(\frac{p_{data}(x)}{p_{zi}}(x) \qquad (5)$$

In the optimal case, the accuracy of the discriminator becomes approximately 50% while the KL-divergence converges to 0. This means that the samples produced by the generator have the exact same class as the real data distribution. Therefore, the discriminator can no longer differentiate between the real and generated samples. The reason behind using KL divergence is to check the network stability and also to check whether the quality of the generated samples will mimic the original samples. In this case, the value of P_{data} is known to the attacker and they are inferring the distribution P_z of the generator network in this model during the training phase. The detailed algorithm is presented below:

4.2.3 Data Generation

Now after the model has been trained, it is possible to generate adversarial synthetic data. It is significant to factor in the amount of data that the model should generate as the objective of the attacker is not only to get the machine learning classifier to mislabel the data but also not to decrease the accuracy significantly as it would then become noticeable to the defender. In order to keep the dataset balanced, we generate the samples with the same class distribution as the original data distribution. The effectiveness of this model is then checked through the comparison with the test set and with performance metrics such as precision, recall, and accuracy.

After both the generator and discriminator have been defined, the model is trained. The two models are trained separately and their loss and accuracy are calculated. This is followed by training the combined model and calculating its loss and accuracy. The model and training history are then saved. The accuracy of the model is then checked by taking a set of random labels and generating synthetic data.

Testing the Model

It is important to check the quality of the synthetic data which can only be accomplished by using it to train different machine learning classifiers. The idea is to train the machine learn-

ing algorithm on the original data that was used to train the GAN and test it with T_s which is the synthetic data. If the test accuracy is high then it would indicate that the model is robust and not affected by the synthetic data whereas the alternative would be that the accuracy is low and that the model is incapable of classifying the generated data. The most common machine learning classifiers are the following: Random forest, SVM, multi-level perception, and decision trees.

4.2.4 Defense Mechanism

The defense mechanism is analyzed through its performance on several machine learning models. The mechanism works in the following way:

Before feeding the model into the federated learning environment, the defense for the adversarial setting is checked on the global model itself. The idea is that feature permutation i.e noise makes it so that adversarial examples can not be detected by the ML model. Therefore, the idea is to train the model on CGAN examples by appending the generated examples with the original dataset so that the model is more robust.

Initially, down-sampling of the training data is done in order to keep the balance of the malicious and benign data samples by using one of several methods ADASYN, SMOTEENN, BorderlineSMOTE, or SVMSMOTE. The function then generates some synthetic data the generator function concatenates the synthetic data with the original training data to create a new training set, and trains and tests several machine learning models on this new data.

The first step was to check whether the GAN model is effective against the current IDS system. The attack was tested on multiple classification algorithms such as Support Vector Machines, Random Forests, Decision trees, and Multi-layer Perceptrons in order to ensure that the attack was successful against each model. The results of the model before the attack showed an accuracy of 98% on the test set whereas it dropped down to 39% after the GAN-generated samples were considered as the test set. Thus, it demonstrated a loss of approximately 50% showcasing the success of this strategy against this particular system.

4.3 Development of the Label Flipping Attack

It is imperative to understand the threat landscape that this attack is being conducted under so as to better understand the objectives the attacker needs to achieve and the obstacles that they need to face. There are two possible scenarios that exist for the attacker: the model is either a black box or a white box.

4.3.1 Traditional Scenarios

In the case of a model being a white box, the attacker has all possible knowledge of the model including the model design, classifying algorithm, and even the underlying features of the data. While this case is unlikelier to occur in the real world as compared to its counterpart, however both the attack and defense for this scenario have already been researched and published [50].

In the black box scenario, the attacker has no knowledge of the model parameters but they can access the training set and query the model. As such when the attacker inputs traffic into the model, they gain access to the output and understand whether the model will label their attack as malicious or benign. Hence, if the adversary were to query any input x then they would receive the predicted class probabilities $P(y|x)$ for all the classes y. Both the attack and defense for this scenario have also been researched and published [29].

4.3.2 Implementation

This is an important area to explore as in the modern era most of the refined datasets are in the public domain and it is those very datasets that are used for commercial purposes. A good example of this in intrusion detection is the UNSW-NB15 and KDDCUP where both have been used for scientific research. Similarly, the MNIST dataset is also quite famous for image classification and recognition. Hence, it can be difficult to keep a dataset obscure in the modern world as they are usually open source.

The objective of the attacker is to use their access to the training set in order to implement an adversarial attack on the binary classification model. This is accomplished by looping through each sample xi in X and making changes to the label according to the poison parameter epsilon. As such only a certain portion of the data is perturbed in order to keep it unnoticeable to the defender. The feature values for the chosen example are perturbed by adding epsilon times

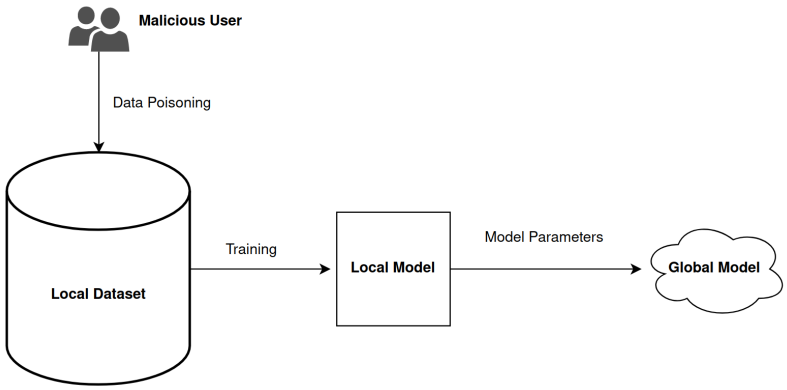


Figure 4.3: Label Flipping Model.

the sign of the difference between the true label and the predicted label. The function then predicts the class for the perturbed feature values using the black box IDS model. Lastly, as a precaution, if the accuracy of the model on the perturbed features is greater than its accuracy on the original features, the perturbed feature values are set back to their original values. The attack model is shown in 4.3

Defending against a black box label-flipping attack can be challenging because the attacker has access to the full training dataset and can modify the labels in a way that is difficult to detect. One strategy to defend against black box label flipping attacks is to use an ensemble of models, each trained on a different subset of the data or using a different learning algorithm. Combining multiple models to make predictions can lead to better robustness against adversarial attacks. For example, using an ensemble of classifiers can help reduce the impact of flipping a single label. This approach assumes that the attacker cannot modify the labels in a way that affects all of the individual models in the ensemble, making it more difficult for the attacker to succeed in the attack.

4.4 Implementation of the Autoencoder

An autoencoder is an unsupervised machine-learning algorithm and is widely utilized for IDS. The initial idea was to protect the system in the best possible manner as such the unsupervised approach was considered superior to other methodologies.

A deep autoencoder was chosen as the model for the federated learning environment so that it may recognize the patterns of the benign data and learn the feature representation through its numerous neurons and layers. The benign data was split into three parts so that one part could be used to train the mode, the second part could be used to calculate the threshold for the reconstruction error and the last part could be used to test the model. The model was trained on one part of the benign data so that it could extract significant information and learn the input representation. The Mean Squared Error loss (MSE) was considered due to it being an unsupervised classification problem. The threshold is calculated with the second part of the benign data by inputting it into the trained deep autoencoder. Lastly, the third part of the benign data is concatenated with the malicious data in order to check whether the intrusion detection system would correctly classify the malicious and benign data respectively.

5 Datasets and Results

5.1 Existing Datasets

A wide array of Datasets for intrusion detection and anomaly detection have been released in the past two decades where each dataset has something which makes it unique and better for a particular problem. The first dataset for ML-based intrusion detection was DARPA [43] which consisted of the following attacks: buffer overflow, synflood, simulated Denial of Service (DoS) attacks guess the password, and NAMP attacks. However, in recent years the research communities have advanced by providing datasets that not only have better quality but have also improved the weaknesses of their predecessors. The following are the most commonly used datasets: KDD Cup 99 [51], NSL KDD [52], DEFCON [53], CAIDA [54], LBNL [55], CTU-13 [56], UNSW-NB 15 [57], and Bot-IoT [58] datasets. Considering the advancements made in recent years with regard to telecommunication, most of these datasets have also become outdated for modern networks. Currently, the most widely used dataset are the UNSW-NB15, CICIDS2017, and Bot-IoT as they are not only recent but are also useful for a wide array of tasks that involve machine learning.

The CICIDS2017 dataset is an intrusion detection dataset and contains a wide array of attacks which include DDoS, DoS, infiltration and heart bleed attacks [23]. The dataset is based on a realistic network and the benign traffic contains protocols that mimic a real network such as the Hypertext Transfer Protocol Secure (HTTPS), Hypertext Transfer Protocol (HTTP),

File Transfer Protocol (FTP), email protocols, and Secure Shell Protocol (SSH). On the other hand, the UNSW-NB15 dataset that was created by the IX-IA Perfect-Storm tool in the Cyber Range Lab of UNSW Canberra had real traffic for benign activity whereas the malicious traffic was generated through synthetic data [57]. However one of the good qualities of this dataset was that it contained nine types of attacks, which were Backdoors, Fuzzers, DoS, Generic, Analysis, Exploits, Reconnaissance, Worms, and Shellcode.

Lastly, the Bot-IoT dataset was similar to the UNSW-NB15 as it was also created in a virtual environment where both the attackers and the defenders were virtual instances. It contained approximately 72 million records and a wide array of attacks such as DoS attacks, probing attacks, and information theft that was simulated in an IoT network through the Node-red tool [58]. The IoT devices included a smart fridge, motion-activated lights, a weather station, a remotely activated garage door, and a smart thermostat. Although these datasets have been used for ample research projects, researchers have listed problems that have arisen from using these data. Some of the issues are the usage of synthetic data, redundant data, ignorance of real-world conditions, and the lack of diversity [59].

5.2 5G Network Intrusion Detection Dataset

Before 2022, there did not exist a dataset based on real 5G traffic however researchers [4] created a 5G Network Intrusion Detection dataset through the utilization of 5G Test Network Finland. The benign and malicious traffic was real as it utilized a real 5G architecture. The test-bed [4]is shown in 5.1. This dataset is thus novel as a real network flow for 5G traffic did not exist prior to the release of this dataset. The two attacks that this dataset contains are the different variants of the DoS and port scan attacks.

The three most common categories of DoS/DDoS attacks are volume-based, protocol-based, and application layer attacks [60]. In volume-based DoS attacks, the adversary sends high traffic to the server in order to deplete the resources. Malicious users thus exploit this weakness by using multiple end devices to send traffic simultaneously in order to cause system failure or slow it down. User Datagram Protocol(UDP) floods, Internet Control Message Protocol (ICMP) floods, and SYN floods are the most common examples of volume-based DoS attacks. While the manner in which these attacks are conducted is different however they each have the same objective. The UDP flood attack works by the attacker sending the UDP packets at a faster rate

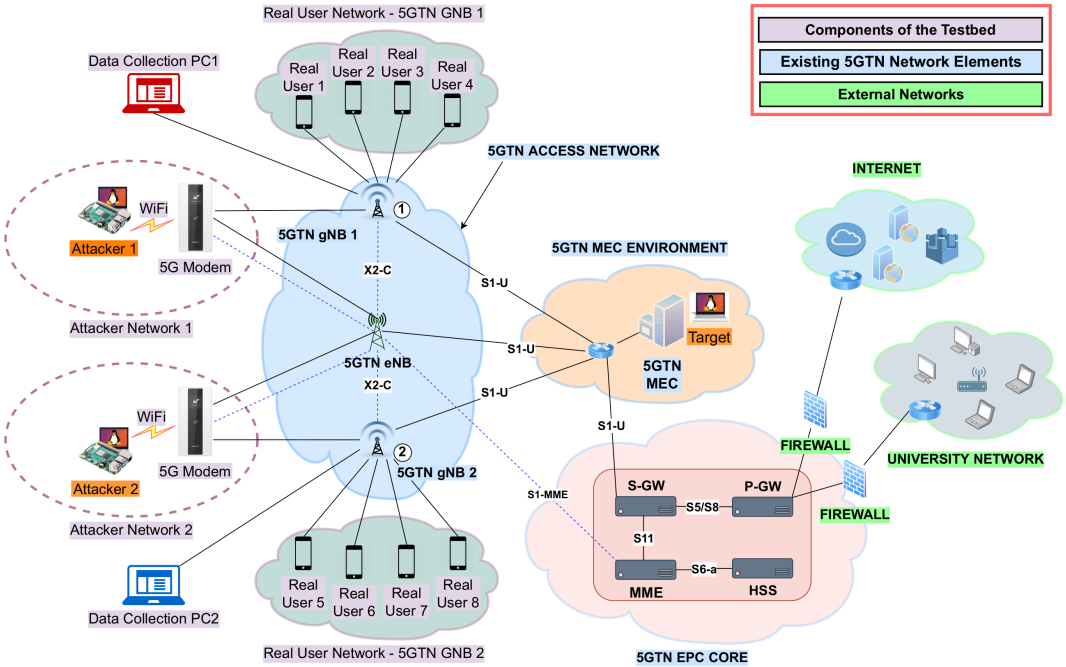


Figure 5.1: 5G Testbed Architecture. Source: [4]

and this is possible due to the fact that this is a connection-less protocol. Similarly, the ICMP flood attacks use the ICMP echo requests to request the same Internet Protocol (IP) repeatedly and this high frequency of requests results in overwhelming the network and making services unavailable to the users. This dataset contains all these examples of DoS attacks.

Port scans on the other hand are not actual attacks but rather they precede the attacks in order to identify the vulnerabilities in the architecture. These port scans send the requests to the host of the system and monitor the response given by the system for that particular input. This response is usually sufficient to determine the status of the port however in certain circumstances the attacker may require a higher understanding of the network architecture. This scan allows the malicious user to determine the exploitable host in the network and target the attack toward that specific network point. This dataset contains the following classes for port scans: TCP Connect Scan, SYN scan, and UDP scan.

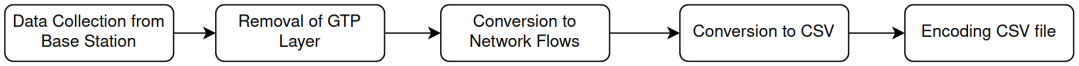


Figure 5.2: Data Preprocessing.

5.3 Data Preprocessing

The data was collected by capturing malicious and benign traffic passing through the network at the two base stations. The data was captured in the pcap format and this was followed by a series of processing steps in order to convert it into a CSV file. The packets contained a GTP-U layer as the data was captured from the radio of the base station. GTP-U is a new protocol specific to 5G as it is an evolution of the GTP which was used in the 4G networks. Therefore, in order to recognize the significant features available in the data, it is paramount that the GTP-U layer be removed. This was followed by converting the data from a packet-based format to a network flow-based format. While there are certain techniques that can be used through the packet level analysis, it is not the best approach for 5G networks. This is because of the bottleneck problem that is caused by the high traffic and low latency due to the massive number of packets and the high amount of time it takes to monitor each specific packet. Flow-based data is thus more popular in the recent era as it can be utilized in conjunction with machine learning and deep learning algorithms. While there are inherent problems with the probabilistic nature of these algorithms, they are still a better option as they will with relativistic certainty be accurate for a long period of time. The CSV file had 112 features and each network flow had a label classifying it as malicious or benign traffic, and the dataset contained a total of 1,215,800 flows [4]. The data preprocessing steps are shown in 5.2

5.3.1 Encoding

The categorical data in the 112 features were converted into numerical data through one hot encoding. One hot encoding works in the following manner. Suppose that there exists a categorical variable that has $'d'$ distinct values denoted by $(v_1, v_2, v_3 \ldots . v_d)$, then, in order to convert this categorical data into numeric data, a unique integer value is assigned to each index starting from 0. Therefore, index 0 is assigned to v_1, followed by index 1 being assigned to v_2 up till

the last index which is $d - 1$ for v_d. Furthermore, for each categorical value, a binary vector of length d is created where the value of the element is either 0 or 1. The value of the binary vector is 1 in the position that corresponds to the index of the categorical value while it is 0 otherwise. The following is the mathematical notation for one hot encoding:

$$x_i[j] = 1 \textit{ if } j = i - 1 \textit{ where the index of } v_i = i - 1$$

$$\textit{else } x_i[j] = 0$$

It is important to convert the data into numeric data so that all the data types are uniform and it is easy to apply the machine learning algorithms. The null values present in the dataset were also removed however rather than removing the entire network flow, the median of the values was taken instead.

5.3.2 Feature Selection

The objective of feature selection is to identify and select a subset of relevant features from the feature space in order to utilize the features which provide the most value to the predictive mode. This in turn helps to improve the accuracy of the model by reducing the prevalent noise in the data and helps with the interpretability of the model by ensuring that it focuses on the most significant features. Moreover, it is one of the essential techniques for preventing the overfitting of the model which occurs either when the model does not generalize well due to the complex nature of the model or due to the model being overly biased by the training data. The model, therefore, performs inadequately on the new data. Another advantage of feature selection is also the reduction in the computational power as the model is comparatively simpler.

There are several different approaches that can be utilized to perform feature selection such as filter methods, wrapper methods, and embedded methods. Filter methods utilize a ranking-based approach where the features are given a hierarchy based on their importance with respect to the target class. This hierarchy is calculated based on certain statistical measures such as correlation, mutual information, or chi-squared test. Wrapper methods on the other hand involve selecting the features based on their performance against different machine learning algorithms such as MLP, SVM, decision trees, random forest, etc. The features are iteratively selected at random and the performance of each subset is evaluated. The subset with the best

performance is then selected. There were three distinct approaches that were utilized to find the relevant features for this dataset.

Pearson Correlation

The first approach that was used for feature selection was Pearson Correlation. The idea behind this approach is that if one single feature has the exact same relationship with another feature i.e correlation is very high then that feature is considered redundant and removed. These features can thus be removed without affecting the quality of the data. The mathematical notation for Pearson Correlation is given by :

$$p_{x,y} = \frac{COV(X, Y)}{\sigma_x \sigma_y}$$

$$= \frac{E(X - \mu_x)(Y - \mu_y)}{\sigma_x \sigma_y}$$

This correlation is calculated for every single pair of features and as both positive and negative correlation indicate a strong association thus the absolute values of the coefficient are taken. The arbitrary threshold was kept at 0.85 to determine whether any pair had a strong relationship and to drop one of the two in case they did. The methodology behind dropping the feature was to eliminate the feature which had a lower correlation score with the label which was the target variable. The statistical score is calculated after dropping these redundant features. The result of applying this is shown in 5.3

ANOVA F-scores

The ANOVA F-score is one of the many statistical measures which can be utilized to rank the importance of the feature in a hierarchical manner [61]. This score is calculated based on the ratio between the variances of the features [61]. The following equations were used to calculate both the variances within the groups and between them:

$$Variance \;\; between \;\; groups = V_G = \frac{\sum_{i=1}^{n} n_i (Y_i - Y)}{K - 1}$$

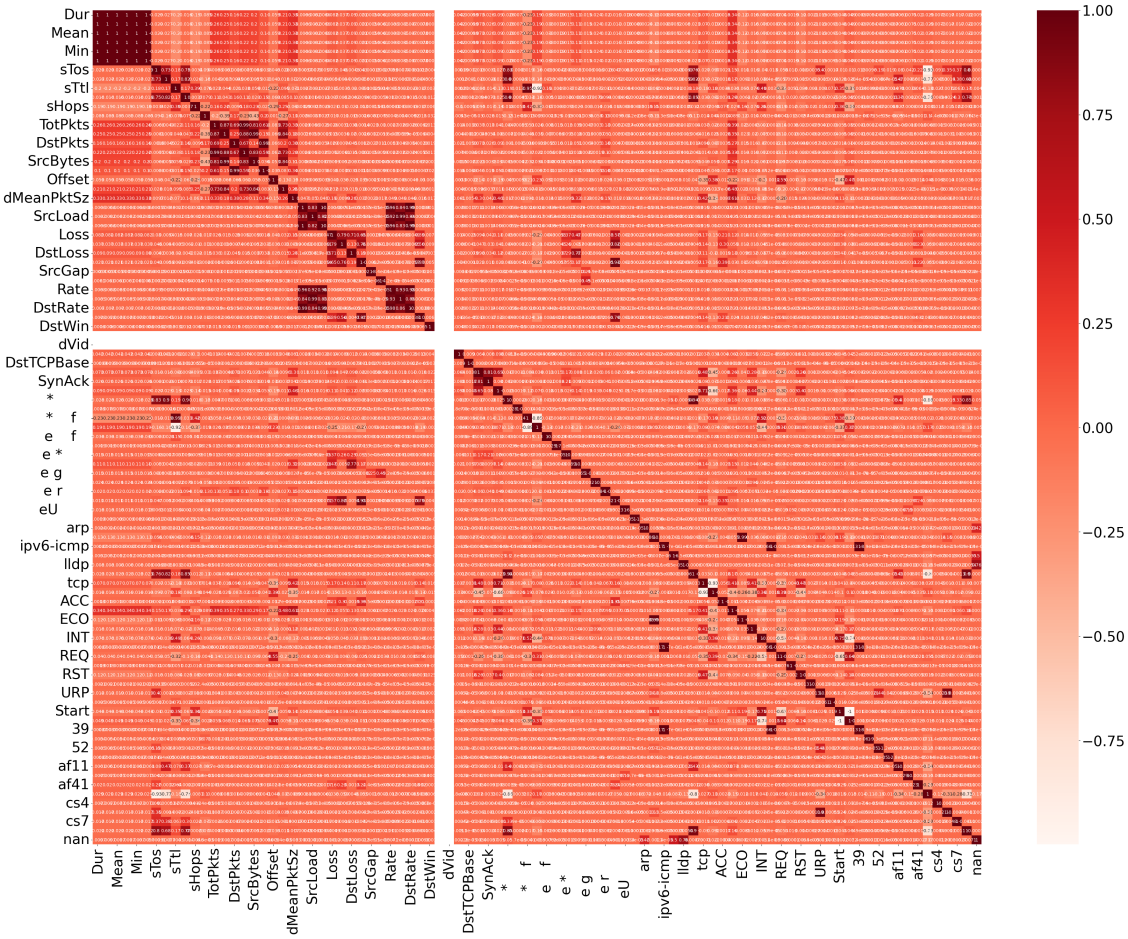


Figure 5.3: Pearson Correlation.

$$Variance \ \ within \ \ groups = V_{wg} = \frac{\sum_{i=1}^{K} \sum_{j=1}^{n_i} (Y_{ij} - Y)}{N - K}$$

The F-score is thus simply the ratio of the above two values

$$ANOVA \ F - Score \ = \frac{V_G}{V_{wg}}$$

In the above equations, Y is the total data available in the dataset, N is the total sample size, and Y_i refers to the mean of the group i. Moreover, n_i is the number of observations in that particular group, K is the total number of groups, and Y_{ij} is the observation j in group i. This statistical analysis was conducted using the sci-kit library available in python and the top 22 features were considered for both the multi-class classification and the binary classification. 5.4 shows the features selected for the binary classification.

5.3.3 Data Normalization

After eliminating the redundant features, the next step is to scale the training data. A uniform scale is needed so that more weight is not assigned to higher values. While there are many options to choose from such as Min-Max scaler, Z-score normalization, and Standard scaler, this dataset was normalized through the implementation of the standard scaler. This technique works by subtracting the mean of each observation from the mean of the dataset and then dividing the result by the standard deviation of the dataset. The mathematical notation is given by:

$$z_{i,n} = (x_{i,n} - u_i)/\sigma_i$$

	Features	Absolute PCC	Scores	Total
13	Offset	0.456284	8.164353e-01	1.272720
16	Load	0.005509	1.000000e+00	1.005509
1	Seq	0.528342	4.183069e-03	0.532525
5	sTtl	0.427134	2.976782e-06	0.427137
37	e	0.396809	7.649025e-09	0.396809
36	* f	0.394846	6.023962e-08	0.394846
54	tcp	0.379058	4.696540e-08	0.379058
33	AckDat	0.294298	9.944602e-10	0.294298
55	udp	0.288874	9.123324e-09	0.288874
0	Unnamed: 0	0.253466	5.552771e-03	0.259018
63	RST	0.199772	1.588193e-08	0.199772
59	INT	0.195829	1.192646e-08	0.195829
31	TcpRtt	0.191785	1.051901e-09	0.191785
49	icmp	0.183665	1.390326e-08	0.183665
14	sMeanPktSz	0.175235	3.731736e-06	0.175239
58	FIN	0.167183	1.127322e-08	0.167183
7	sHops	0.165791	6.623109e-08	0.165791
25	SrcWin	0.089857	7.509463e-02	0.164952
2	Dur	0.163026	2.358066e-08	0.163026
6	dTtl	0.130383	2.184993e-08	0.130383
12	SrcBytes	0.121893	1.474484e-03	0.123367
11	TotBytes	0.116454	1.465140e-03	0.117919
10	SrcPkts	0.112048	4.828538e-07	0.112048
15	dMeanPktSz	0.108740	3.735041e-06	0.108743
68	Status	0.105471	1.905941e-09	0.105471
67	Start	0.104000	2.776096e-09	0.104000
9	TotPkts	0.098169	4.993730e-07	0.098170
76	cs0	0.092742	1.986505e-11	0.092742
3	sTos	0.086001	5.666893e-07	0.086002
34	*	0.079720	2.648986e-09	0.079720

Figure 5.4: Feature Scores.

5.4 Performance Metrics

Considering that this problem can be a multi-label and multi-class classification task or a binary classification task, it was not possible to measure the performance of the system with the usual accuracy score; since the latter, in fact, is calculated on the number of correctly guessed targets divided by the number of total samples in a test dataset. Even if this could still work, it suffers from a big problem: it's hard to state how to count the semi-correct inferred samples; if we count only the totally correct samples, we would probably get a close-to-zero accuracy even with a pretty precise model, because only a wrong inferred label on a sample invalidates the full inference.

There are certain performance metrics that need to be utilized in order to understand the performance of the machine learning algorithm, The most commonly used metrics for the purpose of evaluation are precision, recall, accuracy, and F-score. These are shown in conjunction with the confusion matrix.

5.4.1 Precision

Precision is defined as a measure of the percentage of positive predictions correctly classified by the model. It is the proportion of all the correct predictions to the total positive predictions made by the model. Therefore, a high precision score shows that the model is making accurate predictions whereas a low precision would state that the model is making more false predictions as compared to true predictions. Mathematically, precision is the ratio of the sum of all the true positives for each class with respect to the total number of true positives and false positives of all classes. In this domain, we express precision as the proportion of correct inferred labels on the total number of inferred labels; it is, therefore, an expression of *how precise* our model is, disregarding how many labels are inferred.

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

The value of precision is quite significant in conditions where false positives can prove detrimental such as in anomaly detection or medical diagnosis. This indicator is quite powerful when combined with recall and F-1 scores.

5.4.2 Recall

The recall is defined as the capacity of the classifier to correctly identify the positive instances in a dataset. This measures the percentage of positive instances that the model identified correctly. In an ideal scenario, the recall of the model would be high as it would indicate that the classifier is capable of classifying most of the positive instances whereas a lower recall would indicate that the model does not recognize the positive instances with a good frequency. This metric is mathematically defined as the number of true positives divided by the number of true positives and false negatives:

$$recall = \frac{TP}{TP + FN}$$

In this domain, the recall was defined as the proportion of correct inferred labels among the total number of correct labels; it is, therefore, an expression of *how many* correct labels are inferred by the model. The recall, contrarily, to the precision focuses on the cases where false negatives are costly.

5.4.3 F1-score

This metric is defined as the harmonic mean of precision and recall. This performance metric is the combination of precision and recall to provide the overall measure of the accuracy of the model. It is mathematically computed in the following way:

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

The precision and recall are both given an equal priority in this measurement so that there is a balance where both the recognition of the false positives and the false negatives. A high score would indicate that both the precision and the recall are high whereas a low score would indicate that either both of the aforementioned values are low or at least one of them is. This would imply that the model is either making incorrect predictions or missing many of the positive instances in the dataset.

These three metrics enable a complete understanding of the performance of the system. For example, if the model has low precision and high recall, then it would imply that the system is inferring an incorrect label for most of the network flows on average i.e. the model is making

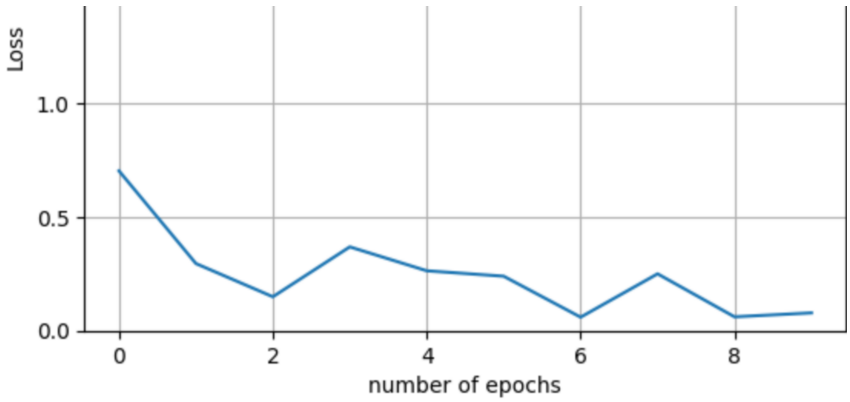


Figure 5.5: GAN loss

incorrect predictions. However, if there is a high precision and low recall then the system is probably inferring fewer labels, but most of them are correct.
In order to compare the overall performance of two or more versions of the system (*e.g.* different hyperparameters or a different network structure), the best practice is to use the F1-score: a lower score on system A means that system B is performing somewhat better, and vice versa.

In the implementation, all three metrics were computed both during the training phase and during the testing phase: during the former, the measurements were received after every epoch, in order to have an idea of the training process whereas in the latter, it was computed only to understand the final performance of the system.

5.5 Results of GAN

The following section is dedicated to displaying and explaining the results obtained from the implementation of section 4.2 on the 5G NIDS dataset. The following hyperparameters were turned in order to provide the best results for the GAN model:

- **Batch size**: The following batch sizes were tried `32`, `64` , `128` and `256` both for the training and the testing phase;
- **Learning Rate**: The following learning rates were used: `0.001` `0.01`, `0.05`, `0.1` and `0.5`;
- **Number of epochs**: While the different number of epochs was attempted however due to the time constraints regarding training only the following epochs were used: `5`, `10` and `20` epochs and tweak the patience parameter instead;

Parameter	Values
Batch Size	128
Learning Rate	0.01
Activation Threshold	0.3
Weight Decay	0.1
Activation	tanh
Optimizer	Stochastic Gradient Descent
Epochs	10
Loss	Cross Entropy
Layers	10

Table 5.1: Hyperparameters for the GAN model

- **Patience**: A variable patience between `5` and `15` was used;
- **Activation threshold**: An activation threshold of `0.2` and `0.4`, sometimes with some adjustments such as `0.25` or `0.35`;
- **Weight decay**: we tried a weight decay equal to `0.01`, `0.1` and `0.5`.

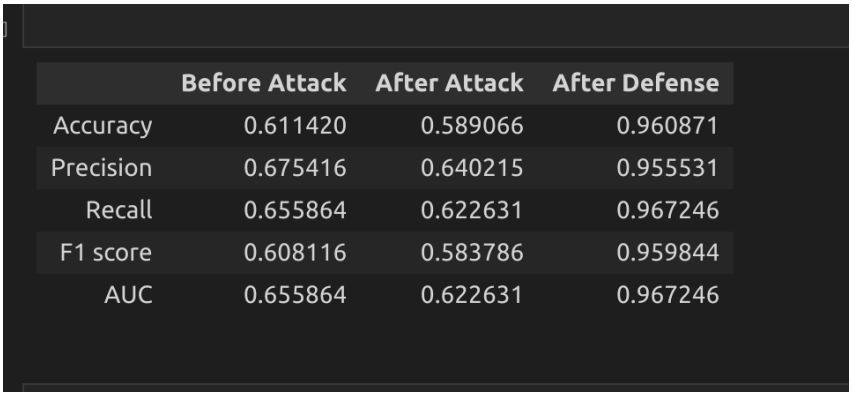
The hyperparameters for this model are shown in 5.1:

The following graph i.e 5.5 shows the training loss for the GAN model. The discriminator and generator loss along with the KL divergence are shown in the table. After training the GAN, the samples were tested on the global model to check whether they would reduce the performance of the model. The results of the attack on the machine learning algorithm are given in the table:

This was followed by the development of a defense for this model which was done by stacking the generated samples in the dataset and then retraining and testing the model. The models used were the following: random forest, decision tree, multi-layer perception, and SVM. The new results after this mechanism was in place showed the following results:

5.6 Results of Label Flipping Attack

The label-flipping attack was an adversarial machine-learning algorithm explained in section 4.3. The results from the attack and defense are depicted through the confusion matrix, accuracy, recall, and the F-1 score. The algorithm used to check the original accuracy was the Gaussian NB. The attacker had access to the dataset and changed the ”epsilon” percentage of



	Before Attack	After Attack	After Defense
Accuracy	0.611420	0.589066	0.960871
Precision	0.675416	0.640215	0.955531
Recall	0.655864	0.622631	0.967246
F1 score	0.608116	0.583786	0.959844
AUC	0.655864	0.622631	0.967246

Figure 5.6: Label Flipping results

Parameters	Device 1	Device 2	Device 3	Device 4
Accuracy	99.987%	99.968%	37.625%	99.792%
Precision:	0.999	0.998	0.997	0.999
Recall:	0.999	0.999	0.351	0.998
F1score:	99.992	99.988	51.933	99.931
TPR:	0.99888	0.99994	0.35017	0.99768
FPR:	0.00701	0.4121	0.00465	0.02128

Table 5.2: Results of Autoencoder before attack

labels which in this case was 0.1. Lastly, a defense mechanism was put in place to prevent the attack. The results of this attack and its defense are shown in 5.6.

5.7 Results for Autoencoder

While the deep autoencoder in conjunction with the federated learning environment provided good results unfortunately due to time constraints it was not possible to defend it against the different attacks defined in this thesis as such this task has been left for future research work.

The autoencoder used the MSE loss function and the SGD optimizer. It was run for 10 epochs with a learning rate = 0.01, weight decay=0.01, and momentum=0.6. The performance metrics for the deep autoencoder are shown in 5.3 and 5.2.

Parameters	Device 1	Device 2	Device 3	Device 4
Accuracy	46.462%	50.540%	53.438%	53.376%
Precision:	0.962	0.936	0.630	0.630
Recall:	0.283	0.352	0.725	0.724
F1score:	43.785	51.121	67.459	67.396
TPR:	0.28343	0.35169	0.72544	0.72443
FPR:	0.03117	0.06736	0.84537	0.84506

Table 5.3: Results of Autoencoder after attack

Parameter	Values
Batch Size	128
Learning Rate	0.01
Log Interval	5
Dropout	0.1
Activation	Rectified Linear-unit
Optimizer	Stochastic Gradient Descent
Epochs	5

Table 5.4: Hyperparameters for the FL model

5.8 Results with Federated Learning Environment

The federated learning environment was set up with 4 devices through pysyft and pytorch. The pysyft library was utilized to decouple the training data for the local model training. Both the global and local models were neural networks i.e Multi-Layer Perceptrons. The hyperparameters used for this model are shown in 5.4.

The learning rate was set to 0.01 and the SGD optimizer was used. Moreover, the cross entropy loss function was used as it performs well for both binary and multi-class classification. The loss for the global model and its accuracy can be seen in 5.7.

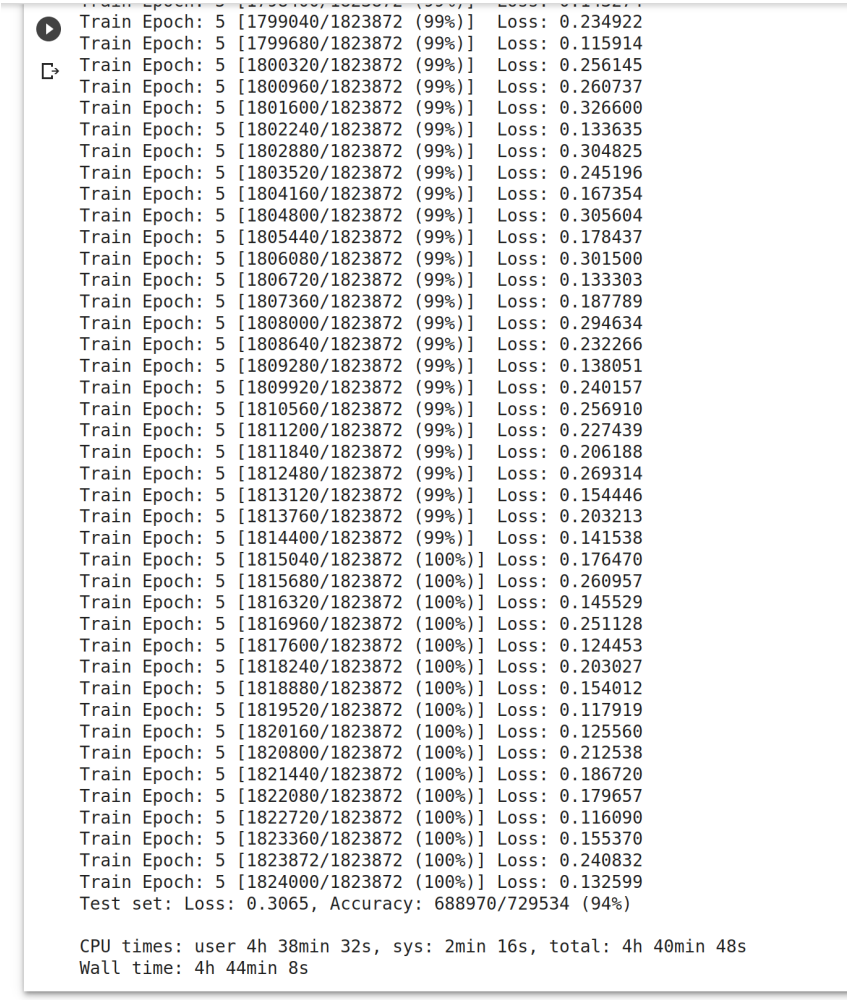


Figure 5.7: FL loss.

6

Conclusion and Future Work

The objective of the thesis was to study the 5G network and to understand its threat landscape in order to design and develop a framework that would protect it against malicious users and adversaries. The federated learning architecture was chosen as the environment for the 5G network to protect the privacy of user data. However, on further analysis of the federated learning environment from the 5G perspective, it was observed that certain exploitative adversarial attacks could be performed against the system. These attacks showed that they could bypass the standard network intrusion detection systems placed at the access point of the 5G network. This revealed that the FL environment while great for privacy concerns had issues against adversarial machine learning as these attacks decreased the overall accuracy of the model which increased the vulnerability of the 5G network. Moreover, it demonstrated the effectiveness that malicious users can have on the IDS through their possession of the user equipment, hence the scope of this thesis mostly revolved around studying the model poisoning attacks from two unique perspectives.

The adversary was able to accomplish their objective through the utilization of label-flipping attacks and GANs. The first idea for the model poisoning attack was the label flipping attack which was responsible for interfering with the training process of the ML model by changing the labels of the target class. This would induce the model to misclassify the data and allow the malicious network data to enter the system. While the first attack was focused on targeting the model during its training stage, the second attack was more direct. The idea was that the

malicious user could create synthetic data which appears to be benign but is however malicious and can use that to infiltrate the system. A defense strategy was also proposed for both of these attacks.

6.1 Future Works

There are certain limitations to this work which would ideally be done for future work. One of the biggest limitations of this model is that it is based on supervised learning and as such can only detect certain threats for which it has been trained. The autoencoder would be the ideal model to be used in conjunction with federated learning as it is an unsupervised model. However, the work has shown that the autoencoder model is currently unable to detect the anomalous data generated by the GANs, and as such, it is susceptible to failure. The ideal case would be for future researchers to modify the autoencoder so that it can recognize the generated samples.

The 5G NIDS dataset was used for both the training and testing of these strategies and as such it is possible to modify the current by using a different dataset. Moreover, this thesis was limited to checking against attacks on the access network however there is data transmission in the 5G core as well such as inside the SEPP. Therefore, another research direction would be aimed towards checking the presence of the intrusion detection systems placed at the SEPP and whether the same adversarial approaches can be conducted or not.

References

[1] "Alliance, n.g.m.n." *5G security recommendations Package, White paper*, 2016.

[2] ENISA, "Enisa threat landscape for 5g networks," *Threat assessment for the fifth generation of mobile telecommunications networks (5G)*, 2019.

[3] A. J. JANI SUOMALAINEN 1, ""machine learning threatens 5g security"," *IEEE Access).*, 2020.

[4] P. P. Sehan Samarakoon, Yushan Siriwardhana, ""5g-nidd: A comprehensive network intrusion detection dataset generated over 5g wireless network"," *IEEE Dataport*, 2022.

[5] I. C. B. Biggio, ""evasion attacks against machine learning at test time,"," *Machine Learning and Knowledge Discovery in Databases,*, pp. 387–402, 2013.

[6] R. A.-K. O. Ibitoye, ""the threat of adversarial attacks on machine learning in network security—a survey,"," *arXiv:*, 2019. [Online]. Available: Available:http://arxiv.org/abs/1911.02621

[7] 3GPP, ""3gpp 33.501: Security architecture and procedures for 5g system. [online]"," 2019. [Online]. Available: "https://portal.3gpp.org/desktopmodules/Specifi-cations/SpecificationDetails.aspx?specificationId=3169"

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[9] A. Dawoud and S. Shahristani", "Deep Learning for Network Anomalies Detection," *International Conference on Machine Learning and Data Engineering (iCMLDE)*, 2018.

[10] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48 231–48 246, 2018.

[11] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things," *IEEE Access*, vol. 5, pp. 18 042–18 050, 2017.

[12] M. K. Putchala, "Deep learning approach for intrusion detection system (ids) in the internet of things (iot) network using gated recurrent neural networks (gru)," *Wright State University,*, 2017. [Online]. Available: https://corescholar.libraries.wright.edu/etd_all/1848/

[13] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–6.

[14] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X17308488

[15] M. Roopak, G.-Y. Tian, and J. A. Chambers, "Deep learning models for cyber security in iot networks," *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0452–0457, 2019.

[16] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the internet of things," *Sensors*, vol. 19, no. 9, 2019.

[17] Y. Otoum, D. Liu, and A. Nayak, "Dl-ids: a deep learning–based intrusion detection framework for securing iot," *Transactions on Emerging Telecommunications Technologies*, vol. 33, 03 2022.

[18] M. Rahman, "Detection of distributed denial of service attacks based on machine learning algorithms," *International Journal of Smart Home*, vol. 14, pp. 15–24, 10 2020.

[19] S. D. Pande, A. Khamparia, and D. Gupta, "An intrusion detection system for healthcare system using machine and deep learning," *World Journal of Engineering*, 2021.

[20] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "A hierarchical hybrid intrusion detection approach in iot scenarios," *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–7, 2020.

[21] X. Z. Lifeng Lei, Liang Kou, "An anomaly detection algorithm based on ensemble learning for 5g environment," *Frontiers in Mobile Multimedia Communications*, vol. 22(19), 10 2022.

[22] J. Lam and R. Abbas, "Machine learning based anomaly detection for 5g networks," *CoRR*, vol. abs/2003.03474, 2020. [Online]. Available: https://arxiv.org/abs/2003.03474

[23] A. H. L. Iman Sharafaldin and A. A. Ghorbani, ""toward generating a new intrusion detection dataset and intrusion traffic characterization"," *4th International Conference on Information Systems Security and Privacy (ICISSP)*, 2018.

[24] I. Alrashdi, A. Alqazzaz, E. Al Oufi, R. Alharthi, M. Zohdy, and H. Ming, "Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning," *IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0305–0310, 01 2019.

[25] N. H. Amir Afaq, ""machine learning for 5g security: Architecture, recent advances, and challenges"," *Ad Hoc Networks*, 2021.

[26] Z. Kamil, Y. Robiah, S. Mostafa, N. Bahaman, O. Musa, and B. Al-rimy, "Deepiot.ids: Hybrid deep learning for enhancing iot network intrusion detection," *Computers, Materials and Continua*, vol. 69, pp. 3945–3966, 01 2021.

[27] B. T. Z. Hu and et al., "Learning data manipulation for aug- mentation and weighting„" *ArXiv.org*, 2019. [Online]. Available: Online]Available:https://arxiv.org/abs/1910.12795.

[28] M. Comiter, ""attacking artificial intelligence: Ai's security vulnerability and what policymakers can do about it,"," *Belfer Center for Science and International Affairs, Harvard Kennedy School. Cambridge, MA, USA,*, 2019. [Online]. Available: [Online]Available:https://www.belfercenter.org/publication/AttackingAI

[29] L. M.-G. A. Paudice and et al, "Label sanitization against label flipping poisoning attacks,"," *Proc. ECML PKDD,*, pp. 5–15, 2018.

[30] H. Xiao, H. Xiao, and C. Eckert, ""adversarial label flips attack on support vector machines"," *ArXiv.org,*, 2012. [Online]. Available: [Online]Available:http://arxiv.org/abs/1206.6389.

[31] N. B.-H. L. Rubinstein, B.I., “”antidote: understanding and defending against poisoning of anomaly detectors.”,” *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement.,*, p. 1–14, 2009.

[32] B. B.-G. G. Maiorca, D., “”towards adversarial malware detection: Lessons learned from pdf-based attacks”,” *ACM Computing Surveys (CSUR),*, vol. 52, no. 4, pp. 1–36, 2019.

[33] X. M.-F. L. H. S. X. L. Z. H. L. B. Chen, S., “”automated poison- ing attacks and defenses in malware detection systems”,” *An adversarial machine learning approach. computers and Security*, p. 326–344, 2018.

[34] M. M. Demontis, A., “”automated poisoning attacks and defenses in malware detection systems”,” *28th USENIX Security Symposium*, pp. 321–338, 2019.

[35] B.-B. Muñoz-González, L., “”towards poisoning of deep learning algorithms with back-gradient optimization.”,” *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security.*, pp. 27–38, 2017.

[36] H. W. Shafahi, A., “”poison frogs! targeted clean-label poisoning attacks on neural networks.”,” *Advances in Neural Information Processing Systems.*, pp. 6103–6118, 2018.

[37] M. R. Suciu, O., “”when does machine learning fail? generalized transferability for evasion and poisoning attacks.”,” *27th USENIX Security Symposium.*, pp. 1299–1316, 2018.

[38] R. B. Pillai, I., “”is data clustering in adversarial settings secure?”,” *2013 ACM Workshop on Artificial Intelligence and Security.*, pp. 87–98, 2013.

[39] B. B. Xiao, H., “”is feature selection secure against training data poisoning?”,” *International Conference on Machine Learning.*, pp. 1689–1698, 2015.

[40] M. M. Demontis, A., “”why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks”,” *28th USENIX Security Symposium*, pp. 321–338, 2019.

[41] O. A. Jagielski, M., “”manipulating machine learning: Poisoning attacks and countermeasures for regression learning.”,” *2018 IEEE Symposium on Security and Privacy*, pp. 19–35, 2018.

[42] S. T. Vale Tolpegin and L. Liu, ""data poisoning attacks against federated learning systems"," *Georgia Institute of Technology,*, 2020.

[43] J. Lippmann, D. Graf, ""evaluating intrusion detection systems: The 1998 darpa offline intrusion detection evaluation,"," *Information Survivability Conference and Exposition*, vol. 2, 1999.

[44] W. D. Carlini, N., ""adversarial examples are not easily detected: Bypassing ten detection methods."," *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security.*, pp. 3–14, 2017.

[45] M. Rigaki, ""adversarial deep learning against intrusion detection classifiers"," 2017.

[46] L. J. Yang, K., ""adversarial examples against the deep learning based network intrusion detection systems"," *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM).*, pp. 559–564, 2018.

[47] Y. S. Zilong Lin, ""idsgan: Generative adversarial networks for attack generation against intrusion detection"," *Indiana University Bloomington*, 2022.

[48] I. Ahmad, M. Liyanage, S. Shahabuddin, M. Ylianttila, and A. Gurtov, *Design Principles for 5G Security*. John Wiley and Sons, Ltd, 2018, ch. 4, pp. 75–98. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119293071.ch4

[49] B. D. Bhuyan M.H. and K. J.K., "Surveying port scans and their detection methodologies," *The Computer Journal*, no. 54, pp. 1565–1581, 2011.

[50] T. X. Y. Ma and et al, ""explaining vulnerabilities to adversarial machine learning through visual analytics,"," *IEEE Transactions on Visualization and Computer Graphics,*, vol. 26, 2020.

[51] "Kdd cup 1999 data," 1999. [Online]. Available: Available:http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[52] "Nsl-kdd dataset," 1999. [Online]. Available: Available:https://www.unb.ca/cic/datasets/nsl.html

[53] "Defcon," 2011. [Online]. Available: Available:http://cctf.shmoo.com/

[54] "Caida ddos 2007 attack dataset," 2007. [Online]. Available: Available:https://www.impactcybertrust.org/dataset_view?idDataset=117

[55] V. P. B. Nechaev, M. Allman and A. Gurtov, ""lawrence berkeley national laboratory (lbnl)/icsi enterprise tracing project"," 2004.

[56] "Ctu-13." 2011. [Online]. Available: Available:https://www.kaggle.com/datasets/dhoogla/ctu13

[57] N. Moustafa, ""designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic"," *University of New South Wales, Canberra, Australia,*, 2017.

[58] E. S. N. Koroniotis, N. Moustafa and B. Turnbull, ""towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset,"," 2019.

[59] A. H. L. I. Sharafaldin and A. A. Ghorbani, ""toward generating a new intrusion detection dataset and intrusion traffic characterization"," *ICISSp*, p. 108–116, 2018.

[60] P. S. K. Singh and K. Kumar, ""application layer http-get flood ddos attacks: Research landscape and challenges,"," *Computers and security*, vol. 65, pp. 344–372, 2017.

[61] J. Brownlee, ""how to choose a feature selection method for machine learning,"," *Machine Learning Mastery*, 2019.

Acknowledgments

I would like to praise Allah, the Most Gracious, and the Most Merciful for His blessings during my study and in completing this thesis. I would not be here today if it were not for His Guidance.

First and foremost, I would like to thank my supervisor Alessandro Brighente and my co-supervisor Tuomo Makkonnen for their continued support, guidance, patience, and most importantly, the continuous positive encouragement that helped me in finishing this thesis. Having them as my supervisors has been a great learning experience and a pleasure.

I am very grateful for the opportunity that Fraktal Oy has given me and I will take their message of Cyber Positivity close to my heart for the rest of my career. I want to express my gratitude to Jani Kallio and Marko Buuri in particular for the smooth onboarding process and their assistance with all of the bureaucratic process. A special thank you to all of my fellow colleagues who took the time from their day to encourage and motivate me.

It's not a proper journey without the companions that you meet along the way and I have been fortunate to make some great friends who have been with me through both the good and difficult times. So while I don't have the space to write all your names, know that your names are etched in my heart and I will always remember the great times that we have had.

Lastly, I would of course like to give my gratitude to my family, my father Ahmed Hassan, my mother Sadaf Hassan, and my sister Laiba Hassan whose love and prayers have been a constant throughout my life. They have been the perfect role models that a son could ask for and I will be eternally grateful for their constant support and for the values that they have instilled in me.