Detection of De-authentication attack in IEEE 802.11 Networks: A Machine Learning Strategy

MSc Research Project
Programme Name

Felipe Tavares de Sá
Student ID: x19132352

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Felipe Tavares de Sá

Student ID: X19132352

Programme: Cyber Security **Year:** 2022

Module: MSc Internship

Lecturer: Michael Pantridge

Submission Due Date: 15/08/2022

Project Title: Configuration Manual

Word Count: 6405 **Page Count:** 23

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

Signature: Felipe Tavares de Sá

Date: 15th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

Detection of De-Authentication attack in IEEE 802.11 Networks: A Machine Learning Strategy

Felipe Tavares de Sá
X19132352

Abstract

As technology evolves, new cyberattacks are emerging, creating real danger for users. IoT devices and their vulnerability have made them an easy target for attacks. Even with the dynamic nature of IoT networks, there is difficulty in developing rule-based security systems. This scenario becomes an invitation to employ machine learning techniques. New ways of protection are being studied every day. This research report presents a denial of service (DoS) analysis in a home Wi-Fi network. The environment is a residence with several IoT devices connected to the internet using the 802.11 Wi-Fi protocol. The threat scenario is the Deauthentication attack. The method for Deauth (DoS) classification uses a dataset made up of malicious and legitimate network traffic that was captured using a Raspberry Pi 4 and is based on Random Forest (RF), XGBoost, Logistic Regression (LR) and Decision Tree (DT) algorithms. De-authentication, (DoS) attack is classified with an F1-score of 100% by the XGBoost, RF, and DT models that were developed for this research.

1 Introduction

With the rapid growth in the number of devices connected to the Internet and the increase in connections, the security concerns of today's networks have increased due to the imminent danger of penetration of these networks. Breaches can leave networks vulnerable to attacks at different rates and even new intrusion methods (Hassija et al.; 2019).

According to Kaspersky and Furnell (2014), the technological world is constantly evolving. Users send and receive large amounts of sensitive data on their devices. Unfortunately, most people do not realise how often and easily their data can be intercepted by nearby strangers. According to Br, C.E.R.T. (2020), regardless of the type of technology used, your computer may be experiencing threats when you connect it to the network.

As reported by Assunção (2002), the vulnerability in computerised systems is nothing new, and even with the development of this technology, many people are still terrified of computers because of their inexperience with the computerised world. End users are the biggest victims of attacks, as criminals can steal information and even perform remote attacks from their computers, interrupting the normal operation of the device. Many users are not concerned about possible attacks because they think they have nothing of value that would attract an attacker or think that an attack cannot lead to great damage or loss in their systems (Kombo, 2008).

The standard Wireless Fidelity (Wi-Fi) IEEE 802.11network, according to Bianchi (2000), is an example of the dissemination of wireless technologies. Da Silva (2014) argues that wireless networks have rapidly become a form of communication most commonly used in home environments. By not requiring cables to connect, Wi-Fi networks offer mobility, robustness and ease of installation, making them very popular (Br, C.E.R.T.; 2020). A large number of people use at least one device connected to the internet that helps them perform tasks or search for information.

Several approaches are found in the literature aiming at detecting attacks on computer networks, as an example of these is the use of traditional techniques based on rules and Machine Learning techniques (Buczak and Guven,2015).

Communities (2008) maintains that the next leap in internet growth is based on the Internet of Things (IoT) paradigm since it encompasses hardware, software and services infrastructure that connect physical objects, called things, to the global system of interconnected computer networks. However, since its conception, IoT has presented unique requirements that require distinct strategies for security. With the increasing heterogeneity of devices, Babar et al. (2010) point out that it has become a challenge to add security mechanisms due to computational constraints.

The popularisation of Wi-Fi and smart home devices has empowered security vulnerability. Due to the high number of cyber-attacks end-users suffer on their home Wi-Fi network, the present study sought to contribute to machine learning with a focus on detecting Denial of Service (DoS) attacks, particularly the deauthentication attack.

This research provides information security principles and analyses several machine learning algorithms for 802.11 WLAN traffic analysis. It discusses the risks and threats that occur in technological devices that use this protocol for communication, as well as some types of attacks. The reminder of the research is organized as follows. Section 2 presents an overview on the related work. In Section 3 we briefly describe the research methodology. Section 4 shows the design specification. Section 5 presents the implementation of the proposed solution. Section 6 provides a comprehensive analysis of the results and mains findings. Finally, Section 7 concludes and describe proposal for future work.

2 Related Work

In this topic, it is presented the literature review that addresses the contents that sustain the subject chosen for the development of the research project. At first, brief concepts of Wi-Fi networks, Internet of Things, Information Security and Security Policies are described, to be followed by an explanation of the definitions of Denial of Service (DoS) Attacks. Concluding the section, the concept of Machine Learning through the use of the Logistic Regression, Decision Tree, XGBoost, and Random Forest algorithms is covered.

2.1 Wi-Fi Network

A Wireless Local Area Network (WLAN) is one of the most widely used network technologies for Internet access. Present in places such as companies, homes, universities, hotels and airports and with the forecast that in the near future, it will be available globally everywhere, just like the mobile phone network.

In the 1990s, several technologies were developed for wireless networks, but undoubtedly the one that received the greatest acceptance was IEEE 802.11, also known as Wi-Fi (Kurose and Ross, 2005).

A basic local home network usually consists of a router that receives the internet signal, usually from a broadband modem, and computers or mobile devices that are able to receive its signal (Jobstraibizer, 2010). Because it is flexible, it can be used as an extension or even an alternative to local networks. Because they combine data connectivity and mobility via radio frequency, wireless networks are widely used because of their simplicity of use and installation (De Moraes, 2010).

2.2 Internet of Things

Today, the Internet of Things (IoT) is considered an extension of the internet as it makes common objects or "things" able to communicate and process commands as long as they are connected (Santos et al.; 2016).

These gadgets exchange data in addition to collecting it. Numerous physical things may now be transformed into IoT devices because to the expansion of wireless networks and the availability of affordable CPUs. Their basic building structures are: Identity, which includes the unique identification of objects to connect them to the network; Sensors, which collect contextual information about inserted objects and store them; Communication, which includes technologies to connect intelligent objects; Computing, which includes processing units; Services, which make up the Internet of Things and may vary the services provided; and Semantics, referring to the intelligent capabilities of the objects in the Internet of Things (De Magalhães, 2016).

IoT has already become a significant security threat that has caught the attention of leading IT companies and governments worldwide. According to Figueira (2016) privacy in the IoT space is an issue that needs to be approached with caution. Matching functionality and privacy requirements at different stages of IoT product development and operations is important because some of them are designed to collect data from the environment in a large network of IoT-connected devices. This data will be stored locally or transmitted to the cloud. Consequently, personal or sensitive data are at stake, and, therefore, they must be protected.

2.3 Information Security

Over time, the meaning of computer security has changed; however, Guttman and Roback (1995) define computer security as a group of rules, techniques and mechanisms that aim to preserve the integrity, availability and confidentiality of the system as well as its resources: hardware, software, firmware, data and communication.

For Landwerhr (1981), when occurs the non-verification of confidentiality, availability, and integrity then occurs, the violation of security.

De Moraes (2010) defines the principles governing the fundamentals of information security as:

- **Integrity**: assurance that the information has remained intact, that is, that it has not been modified during transmission or storage.
- **Confidentiality**: the process by which the message remains protected so that unauthorized users are unable to access it. Only the sender and recipient know the content of the message.
- **Availability**: assurance that security mechanisms are in place to prevent the system from being inaccessible and that it is always available to users.

Risks are conceptualised based on their negative impacts because of the exploitation of a vulnerability (Stoneburner et al., 2001). It is possible to identify and reduce them, but they cannot be eliminated in their entirety.

According to Nakamura and Geus (2007), vulnerabilities in a network or system materialise as a result of a design or implementation failure of a misconfigured service, protocol or software. However, there will always be bugs in these applications, even if they are fixed.

Some types of attacks are more frequent, among them, Landwerhr (2001) cites the following:

- **Denial of service (DoS):** bombard the server with numerous requests so that the system cannot respond to any requests.
- **Port scanning:** its purpose is to identify network services that are available, software versions, operating system, whether there is a firewall in the way, and other information.
- **Phishing sites:** aim at stealing users' bank details, for this, they imitate the original sites.

For Wadlow (2000), there is no possibility of acquiring a software or device that makes the network protected in its entirety since, when it comes to security, it is compared to a journey in which the destination is security itself. However, this journey has no end, and as a mitigating measure an acceptable level of risk must be managed.

Digital crimes only grow in number and in the way they are organised because innovative techniques are used by criminals to cover their tracks and prevent their real identities from being revealed (Nakamura and Geus, 2007).

In order to preserve the security properties of the system, a set of defined rules that become a Security Policy is elaborated (Landwerhr, 2001). Previously mentioned forms a foundation to establish what is and what is not allowed (Bishop, 2003).

Spafford et al. (2003) distinguish two different implementation models for security policies:

- **Denial pattern:** aims to identify only what is allowed and deny everything else.
- **Permission pattern:** aims to identify only what is forbidden and allow everything else.

Security policies are divided into three levels: physical security, management security and logical security. Physical security proposes protecting the system's physical resources by prohibiting access by unauthorised persons. While the managerial security policy is responsible for defining the processes that create and maintain an organisation's security policies. The logical security policy elaborates the definition of users who have access rights to the system, as well as what these rights are (Landwerhr, 1981).

2.4 Denial of Service Attack (DOS)

In the early 1990s the first problems arising from denial-of-service attacks on the Internet were recorded. From the acronym Denial-of-Service or DoS, it includes attempts to prevent legitimate users from using a particular service on a computer or network, even if the server, system or network provides available resources or services (Kumar and Selvakumar, 2011).

Kuncheva (2004) exemplifies denial of service as a wireless network vulnerability since it allows an attacker to easily bombard an access point through crafted protocol messages that aim to consume system resources.

Denial of service attacks are conceptualized as a common destruction attack technology because they interfere with the network operation using various methods with the aim of making the network unable to provide services (Augusto Filho, 2021).

Even if the victims are large business conglomerates, distributed attacks are a real threat capable of flooding the service. Some well-known websites such as Yahoo, eBay, Amazon.com, and CNN.com have become targets of successfully executed denial of service attacks (Garber, 2000). The victim simply stops serving legitimate customers while trying to process the traffic generated by the attack (Mirkovic et al.; 2004).

2.4.1 Deauthentication attack

The De-authentication attack is defined by De Moraes (2010) as an attack aimed at communicating between the router and the device, causing a disruption in the communication between them and, consequently, making the device inaccessible.

Among the various purposes of the deauthentication attack is to leave the user disconnected or create a fake network with the same name, also known as rogue access point. By accessing the rogue network, the user becomes vulnerable to having his data stolen by the attacker. (Moreno, 2016).

For this type of attack, there is no security measure by the IEEE 802.11 protocol, which requires a security action with the responsibility to identify and ignore these malicious frames without eliminating the functionality of the network. In a home network, as the object of study of this research, the impacts caused by this type of attack are that the victim's device gets disconnected from the network and, therefore, becomes incommunicable.

IoT devices typically face the same types of cybersecurity and privacy risks as traditional IT devices. However, the prevalence and severity of those risks are different. The consequences of a deauthentication-type attack on IoT devices are as varied as possible. Many of these devices are empowered with actuators to make changes to physical systems that ultimately affect the physical world (Magrani, 2018).

For many IoT devices, availability and integrity are more important than confidentiality due to the latent repercussions in the physical world. For example, if a camera monitoring system is no longer connected to the network, human safety may be compromised because of operational disruption (Boeckl et al.; 2019). Another example would be the automation of a residence to facilitate and simplify the actions to be performed daily, such as temperature sensors, controllable lighting, sensors and alarms. The Smart Home needs to be connected 24 hours a day (Lima and Panham, 2019).

According to Moraes (2010), in this type of network, the security point has left much to be desired since extending the security perimeter, as a consequence, increases vulnerability, causing breaches with the use of simple techniques.

Wright (2005) states that, unlike most radio jammers, deauthentication acts uniquely. The IEEE 802.11 (Wi-fi) protocol provides a deauthentication frame. The malicious attacker can perform sending of the deauthentication frame at any time to a wireless access point, using a fake address for the victim. The malicious attacker can send deauthentication frames at any time to a wireless access point using a fake MAC address. In this attack mode, the 802.11 protocol does not require any encryption, even though the established session uses Wired Equivalent Privacy (WEP) for data privacy. It only requires the attacker to know the MAC address of the device, information which can easily be obtained by sniffing the surrounding Wi-Fi environment (Mateti, 2005).

In the cyber community, the deauthentication attack aims to force users to connect to a twin access point, which can then be used to capture network packets transferred between the user and the access point. By conducting the deauthentication attack, the attacker disconnects the target from its current network and leads it to automatically connect to an evil twin point (McCullagh, 2013).

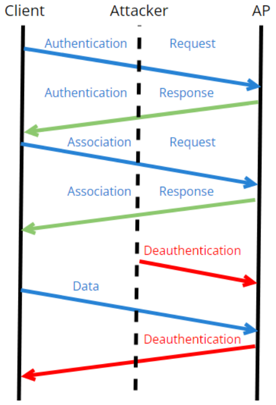


Figure 1: How the De-authentication attack works on Wi-Fi 802.11

2.5 Machine Learning Development

Since the 19th century anomaly detection has been studied, and as a result of these studies a variety of techniques and solutions have emerged. Denning (1987) was the precursor in introducing basic concepts of anomalies within the context of network security. Since then, numerous approaches have been developed that among themselves borrow schemes, techniques, and methodologies from the machine learning area.

Géron (2017) defines machine learning as the science of programming computers so that they can learn from data. A computer program can learn through experiences related to tasks and with some measure of performance. Through performance measurement, it can improve the experience (Mitchel, 1997). DDoS detection system can be built to learn through the network data samples and can elaborate classification of traffic as Benign or Malicious.

Gates and Taylor (2006) state that the anomaly detection approach needs to deal with a cluster of well-known problems: detectors that tend to generate a high number of false positives; it is difficult to find attack-free data to train the solution; and attackers can avoid detection by gradually teaching the system to accept malicious activities as benign.

The main machine learning techniques are known as SVM (Support Vector Machines), k-NN (k-Nearest Neighbors), ANNs (Artificial Neural Networks), K-Means, NaiveBayes, and classifier sets, among others, applicable in the area of anomaly detection in computer networks (Henke, et al, 2011).

Haykin et al. (2008) divides machine learning into two fundamental paradigms: unsupervised or supervised learning, where the system needs to know the environment and transmits a set of input-output pairs as a sequence to be followed until the goal is achieved. In contrast, the unsupervised learning approach considers that there are no labelled examples of the behaviour that should be learned; this way, learning can be performed from an input-output mapping with continuous interaction or learning by obtaining knowledge of data resulting from generated processes.

The experiment used in this research project applied as learning techniques the Logistic Regression, Decision Tree, XGBoost and Random Forest algorithms with sklearn, which we will briefly describe, according to the literature review.

2.5.1 Logistic Regression

According to Gonzalez (2018), logistic regression can be defined as a statistical technique that aims to produce, starting from a set of observations, a model that enables the prediction of values taken by a categorical variable, regularly binary, in accordance with one or more autonomous continuous and/or binary variables.

It is a supervised algorithm used in the task of classifying data for machine learning and is usually referred to as a distinct variable classifier. The Logistic Regression algorithm assumes the parametric form of the direct probability classification, considering the weights of the parameters of the training data; through this procedure, it will establish a function that determines the behaviour of these data. For Mitchell (2010) the overfitting of training data becomes a problem that affects the logistic regression. Thus, a measure to reduce the overfitting would be the regulation, whose task is to penalize large values of the weights by the log of maximum reliability.

2.5.2 Decision Tree

The Decision Tree is described as one of the main machine learning techniques (Jain, 2011). In this technique, a decision tree is built based on the data set for training. The rules built can be extracted from various paths proposed by the tree (Henke et al., 2011).

One well-known supervised type of machine learning method for classifications is the Decision Tree algorithm. Based on a tree structure with criteria or rules, this method provides the output as the optimum result. Mitchell (1997) classifies it as a greedy algorithm, which performs a 'top-down' search in space for all viable trees, having the entropy (a measure of the purity of the set of instances) used for the calculation of the gain ratio (GR), which frames the attributes with many probable values.

The Decision Tree learning approach is one of the most used algorithms because of its practical applications, has a method of approximation of discrete functions, is resistant to noise and can learn disjunctive expressions (Mitchell, 1997).Also according to the author, if the sample is reduced, overfitting can occur, which is a biased classification, and as an alternative to avoid overfitting in the decision tree, pruning is suggested.

The fact that it does not require the definition of parameter values makes it advantageous to use Decision Trees in network security applications (Likarish et al., 2009).

2.5.3 XGBoost

This is an optimized "gradient boosting" library that was designed to act in a highly efficient, portable and flexible way. Under the "Gradient Boosting" framework are implemented machine learning algorithms. Classified as a method that helps solve many problems accurately and quickly, even if it contains a rather high number of inputs (Deng et al.; 2021).

The Optimized Gradient Boosting (XGBoost) algorithm is an evolution of "Gradient Boosting" and applies parallelism to processing, in addition to handling missing values as a way to avoid overfitting (Brownlee, 2016).

2.5.4 Random Forest (RF)

Misra and Li (2020) define the Random Forest algorithm as a method that trains numerous decision trees in parallel with bootstrapping (initialisation), followed by clustering. It is a widely used classifier as it tends to outperform most of the usual classification methods in terms of accuracy.

Through this method, numerous individual decision trees are trained in parallel by means of multiple subsets of the training data set, ensuring that each individual decision tree becomes unique, reducing the overall variance of the classifier (Silva, 2021).

3 Research Methodology

As for the technical procedures, the research methodology adopted in this research project was bibliographic research. Gil (2008) argues that bibliographical research is conducted based on material already developed, especially books and articles, and some research prepared from bibliographic sources.

The experimental method was used as research method. Prodanov and Freitas (2013) state that experimental research has greater complexity since it aims to explain, record and analyse researched phenomena. Its main objective is to identify determining factors that explain the reason, the motivation and the motive of all events raised in the research.

The descriptive and exploratory methodology was used as research objectives. According to Gil (2008), descriptive research aims to describe the characteristics of a given place, population or phenomenon. Exploratory research aims to present the maximum amount of information on the subject matter at the preliminary stage (Prodanov and Freitas, 2013).

To build the theoretical framework, research was conducted by extracting concepts developed in books, journals and articles already published on the subject.

As an exploration of the practices elucidated in the referential, a three-step approach was proposed, starting with the capture of 802.11 Wi-Fi traffic considered normal and malicious, data cleaning, followed by the development of machine learning using the aforementioned algorithms.

3.1 Data capture

Wireshark was used to capture the dataset that was utilised in this research, composed of management frames coming from the 802.11 WLAN protocol configured on channel 9. Wireshark is a tool that comes pre-installed on Kali OS, which provides a capture interface for the user, as well as network packets filtering and post-processing.

The Alfa AWUS036NHA antenna device was set up for data collection in monitor mode. With this configuration, the device can capture every packet sent and received in the test environment and all nearby 802.11 network traffic as well.

During the period of two hours uninterruptedly, the normal traffic of 802.11 WLAN was captured and labelled as “normal” traffic. Then, using one attack scenario — a De-Authentication Attack — we create a dataset of packets labelled "deauth". In this scenario, a portable Raspberry Pi with Kali OS was used as the attacker machine, in conjunction with the *aircrack-ng* suite to perform the De-authentication attack. The two databases were then merged into one, and a single dataset with "normal" and "deauth" 802.11 network traffic information was built for further machine learning development.

The dataset obtained in this process is summarised below in Table 1.

Table 1: Data capture overview

Scenarios	Sub-Class	Number of Packets	%
Normal		373941	90.12
Denial of Service	De-Authentication	40999	9.88
TOTAL		**414940**	**100.00**

3.2 Machine Learning Development

In this research, we approach the learning task as a classification task with the goal of identifying whether the packet is a legitimate one (labelled "normal") or a malicious one (labelled "deauth"). All of the assessments were completed using the Python-based scikit-learn kit, an open-source toolkit that includes methods for classification, regression, clustering, dimensionality reduction, and pre-processing.

3.2.1 Criteria for ML performance with an unbalanced dataset

An unbalanced dataset is one in which the majority of the samples belong to a certain class, as seen by an enormous difference ratio between the positive and negative instances. Additionally, the bulk of the packets in security event categorization jobs are from the usual class, while a smaller percentage reflect harmful activity; this is also true for this research (Table I). As a result, the dataset utilised in this study and the anticipated behaviour following deployment (operational phase) are based on the bulk of the packets being normal and a small number being maliciously marked.

The only assessment criteria for this intrinsic imbalance trait should not be Classification Accuracy. Precision, Recall, and F1-Score criteria have been employed in order to achieve this:

Accuracy: by computing the sample ratio that is correctly categorised and the overall number of samples according to the test dataset, accuracy may be measured.

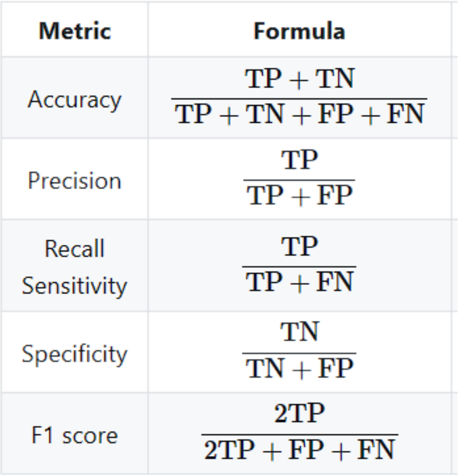
Precision: by dividing true positives by true positives and false positives, precision may be calculated.

Recall: is the percentage of positive "charged-off" events that the classifier correctly determines to be true. It is sometimes referred to as the real positive rate.

F1-Score: a balanced average of recall and accuracy is provided by the F1 score.

Below is table 2 that illustrates how the metrics work.

Table 2: Metrics



Metric	Formula
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
Recall Sensitivity	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
Specificity	$\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$
F1 score	$\frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$

3.2.2 Feature Selection

A packet serves as the inspection unity, and each one that is recorded makes up this dataset. A flow granularity secondly takes into account data aggregation from packet data, such as "sent bytes between source and destination." Although the flow data cannot be converted into packet data, the packet data can be changed into a flow representation.

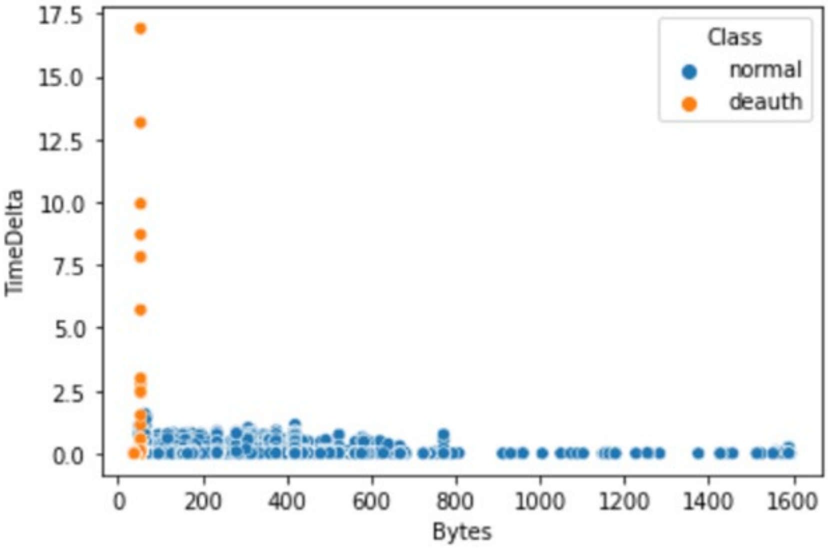


Figure 2: Clear distinction between selected features in the visualization of the characteristics containing Deauth packets

3.2.3 Preprocessing and Exploratory Data Analysis

The dataset acquired at the time of data capture was organized by 6 features: protocol (WLAN), frame length (Bytes), time delta (seconds), destination name, source name and, finally, class with 2 possible categories: “normal” and “deauth”. The solution to this classification problem is a binary classification, where the only classes are “normal” (0) and “deauth” (1), which stand in for Deauthentication attack. With the dataset shown in Table I, we use a stratified split of 60% for training and 40% for testing purpose.

The Wireshark characteristics that offer the best separation between malign (Deauth) and regular traffic were identified as the features taken into consideration for machine learning development: bytes (frame length) and from the previously recorded frame's time delta (seconds).

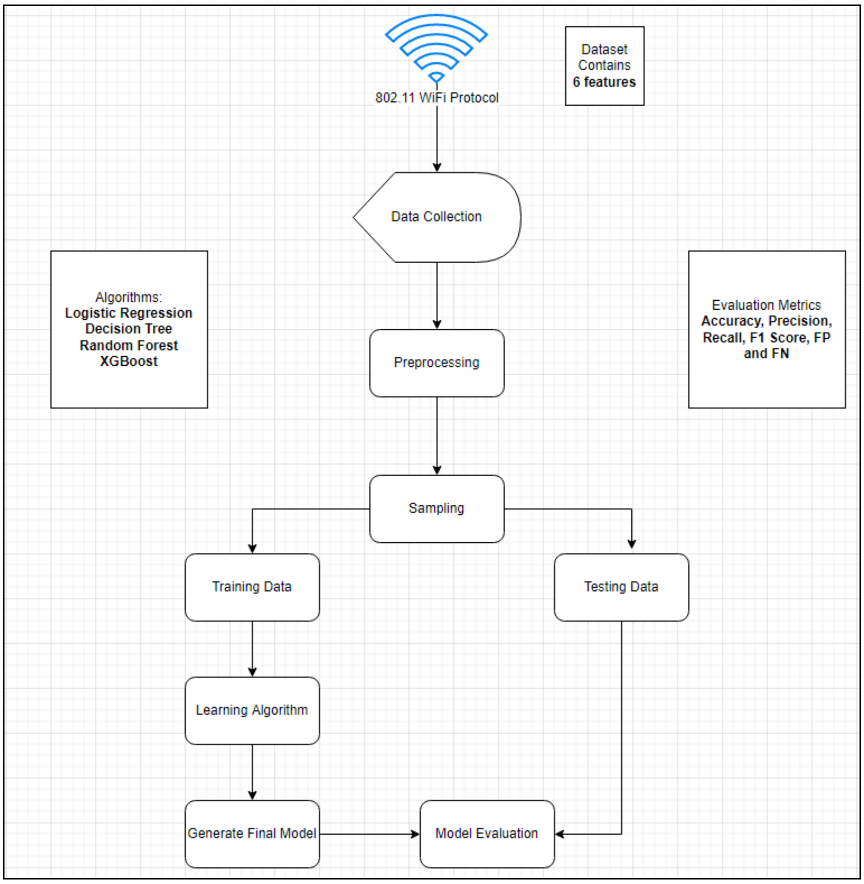
3.2.4 Machine Learning Algorithms

To identify the relevant variables to the problem under study, four different algorithms were implemented:

- **Logistic Regression:** which is a statistical parameterization method that takes into account a binominal classification problem, in the case of this study, normal or deauth packets, determining the classification value using the probability of each feature.
- **Decision Tree**: any form of data, whether numeric, category, or boolean, may be handled by the data type of a decision tree.
- **XGBoost**: powerful and user-friendly algorithm that, in comparison to other algorithms, provides great performance and accuracy.
- Lastly, to ensure superior classification performance and greater accuracy: **Random Forest.**

4 Design Specification

This part goes into detail regarding the suggested model's design and architecture that was covered in the methodology section. The architecture of this approach, as seen in Figure 3, is made up of four major components: data collection, training data, learning algorithm and model evaluation.



802.11 WiFi Protocol
Dataset Contains 6 features
Data Collection
Algorithms: Logistic Regression Decision Tree Random Forest XGBoost
Evaluation Metrics Accuracy, Precision, Recall, F1 Score, FP and FN
Preprocessing
Sampling
Training Data
Testing Data
Learning Algorithm
Generate Final Model
Model Evaluation

Figure 3: Dataflow diagram of the suggested model

5 Implementation

This section outlines the procedure used to put the suggested solution into practise. Using Microsoft Excel, the data was merged, cleaned, and examined. Google Colab was used for data preparation, pre-processing, and model implementation. It is an environment for Jupyter Notebooks that is totally hosted by a cloud server. Due to the fact that it supports a large number of deep learning frameworks7, Python is utilised as a programming language. The Matplotlib package was used in Python to analyse and create graphs from the output of the models.

6 Evaluation

To conduct a Deauth-DoS attack, the attacker Raspberry Pi is equipped with an antenna Alfa AWUS036NHA and the aircrack-ng suite running on Kali Linux. The attacker's main goal is to deluge the target devices(s) with an excessive amount of de-authentication frames, causing the device(s) to disconnect.

The classification performance of the Logistic Regression (LR), Decision Tree (DT), XGBoost and Random Forest (RF) algorithms are listed in Table 3. These measurements were from trained models tested on the test dataset, which made up 40% of the total dataset.

The Decision Tree focused on simplicity while upholding great metrics, employing a maximum depth of 3 that was determined through experimentation and error. We may conclude that DT, XGBoost and RF performed similarly in terms of accuracy, precision, recall and F1-Score. However, as stated in section 3.2.1, other learning measures are better suited for this unbalanced binary classification job (9.88% of Deauth and 90.12% of normal packets - Table 1). In order to determine this, we assessed the F1-score, Recall, and Precision along with False Positive and False Negative produced from the confusion matrix from all of the algorithms. From these measurements, it can be seen that DT, XGboost, and RF outperform the LR in this learning task.

Table 3: "Positive label": Deauth classification algorithms metrics

Metric	Logistic Regression	Decision Tree	XGBoost	Random Forest
Accuracy	0.91	1.00	1.00	1.00
Precision	1.00	1.00	1.00	1.00
Recall	0.10	1.00	1.00	1.00
F1-Score	0.19	1.00	1.00	1.00
False Positive	0.000	0.000	0.000	0.000
False Negative	0.088	0.000	0.000	0.000

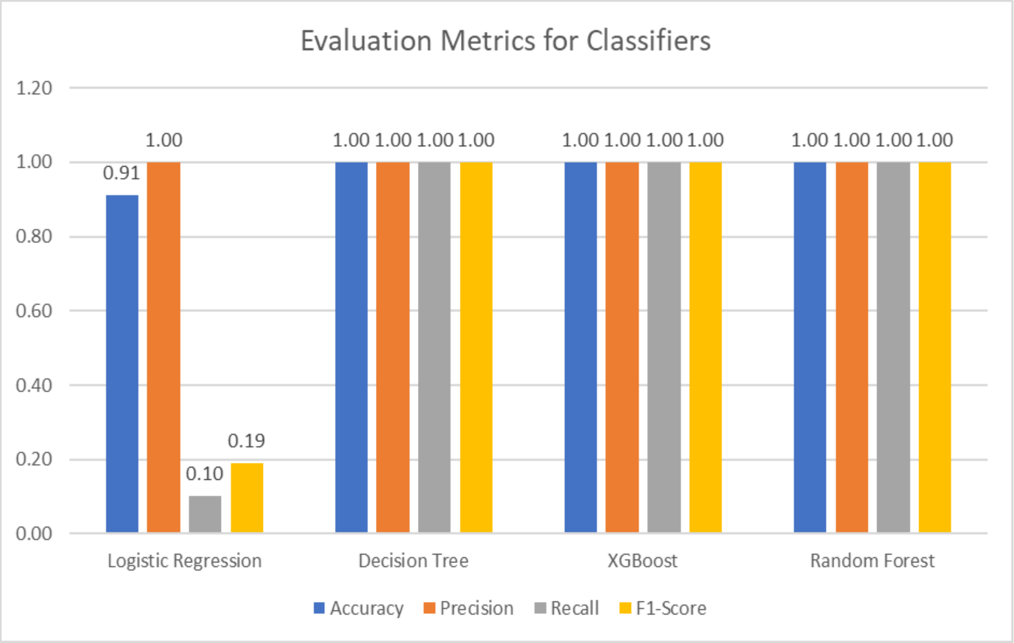


Figure 4: Evaluation Metrics for Classifiers

6.1 Experiment 1: Decision Tree

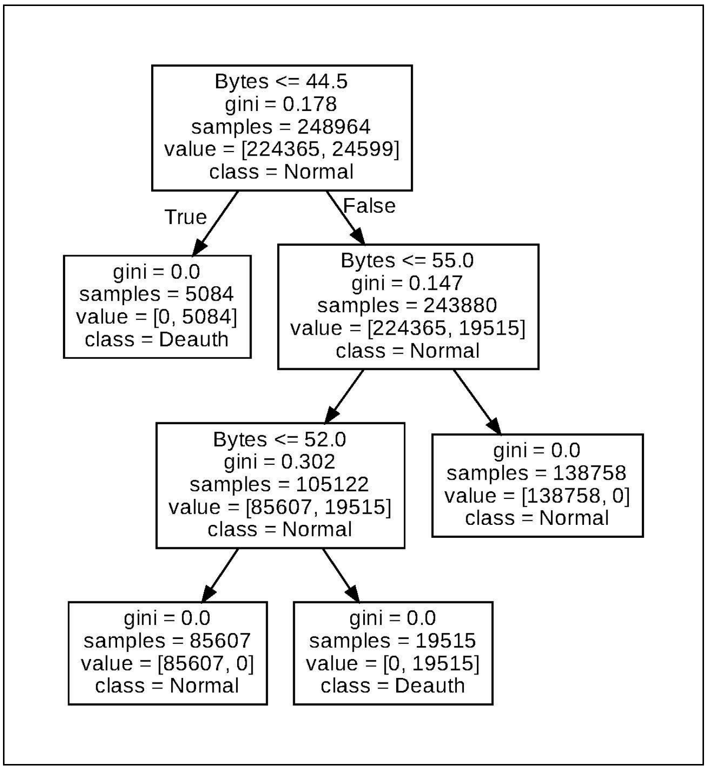
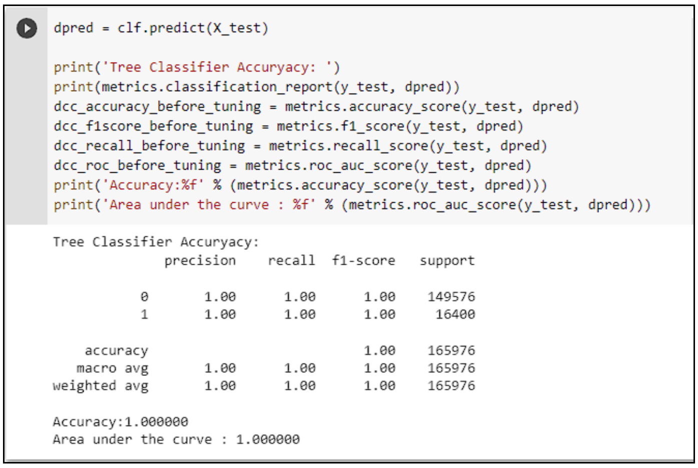


Figure 5: Decision Tree (DT) Visualization - Max. Depth of 3



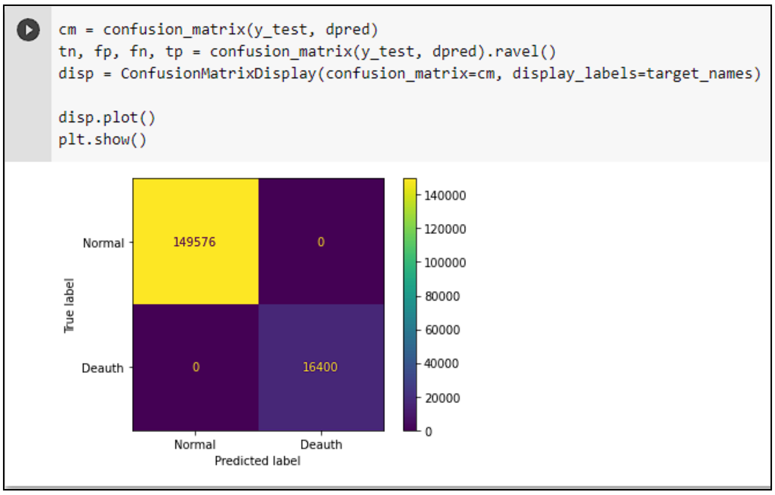
```
dpred = clf.predict(X_test)

print('Tree Classifier Accuryacy: ')
print(metrics.classification_report(y_test, dpred))
dcc_accuracy_before_tuning = metrics.accuracy_score(y_test, dpred)
dcc_f1score_before_tuning = metrics.f1_score(y_test, dpred)
dcc_recall_before_tuning = metrics.recall_score(y_test, dpred)
dcc_roc_before_tuning = metrics.roc_auc_score(y_test, dpred)
print('Accuracy:%f' % (metrics.accuracy_score(y_test, dpred)))
print('Area under the curve : %f' % (metrics.roc_auc_score(y_test, dpred)))
```

```
Tree Classifier Accuryacy:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    149576
           1       1.00      1.00      1.00     16400

    accuracy                           1.00    165976
   macro avg       1.00      1.00      1.00    165976
weighted avg       1.00      1.00      1.00    165976

Accuracy:1.000000
Area under the curve : 1.000000
```

Figure 6: Decision Tree Classifier

Using a conventional Machine learning technique, the precision, recall, F1-score, and accuracy of Decision Tree is 100%, the method requires an average of 7.91 ms to run.

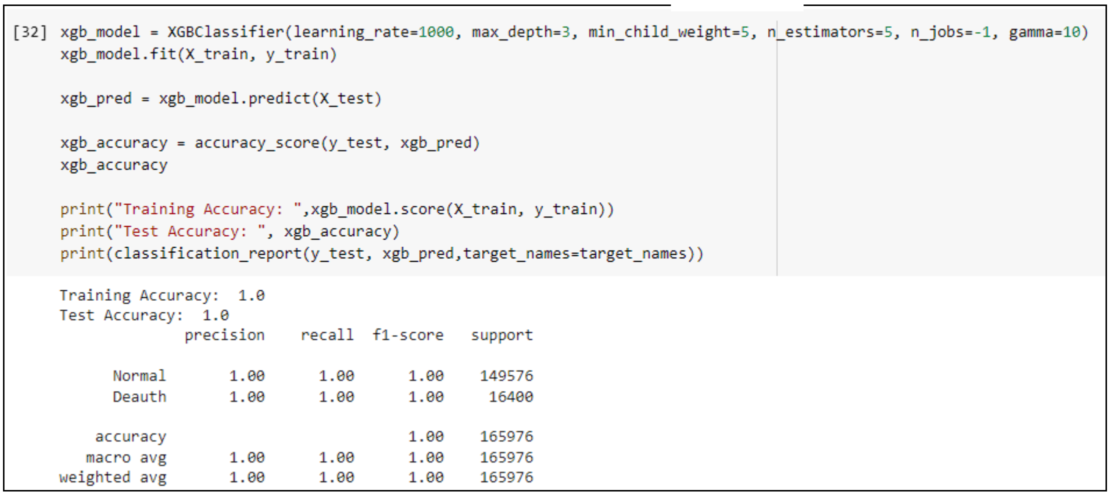
6.1.1 DT: Confusion Matrix



```
cm = confusion_matrix(y_test, dpred)
tn, fp, fn, tp = confusion_matrix(y_test, dpred).ravel()
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=target_names)

disp.plot()
plt.show()
```

Figure 7: Confusion Matrix DT

6.2 Experiment 2: XGBoost



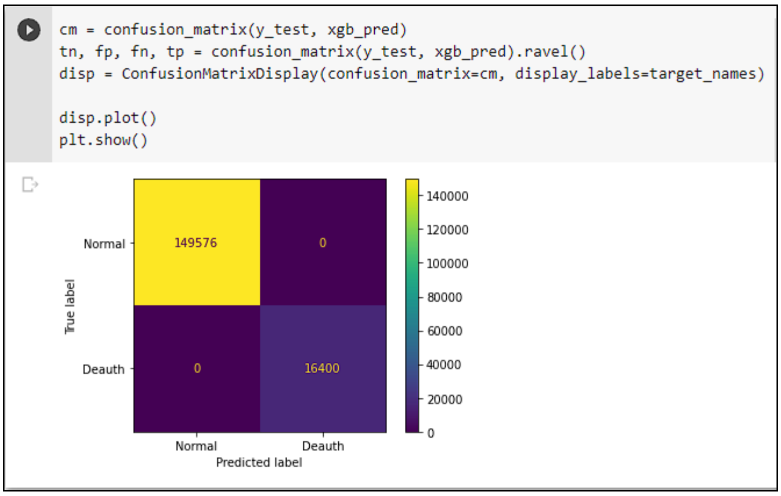
```
[32] xgb_model = XGBClassifier(learning_rate=1000, max_depth=3, min_child_weight=5, n_estimators=5, n_jobs=-1, gamma=10)
     xgb_model.fit(X_train, y_train)

     xgb_pred = xgb_model.predict(X_test)

     xgb_accuracy = accuracy_score(y_test, xgb_pred)
     xgb_accuracy

     print("Training Accuracy: ",xgb_model.score(X_train, y_train))
     print("Test Accuracy: ", xgb_accuracy)
     print(classification_report(y_test, xgb_pred,target_names=target_names))
```

```
Training Accuracy:  1.0
Test Accuracy:  1.0
              precision    recall  f1-score   support

      Normal       1.00      1.00      1.00    149576
      Deauth       1.00      1.00      1.00     16400

    accuracy                           1.00    165976
   macro avg       1.00      1.00      1.00    165976
weighted avg       1.00      1.00      1.00    165976
```

Figure 8: XGBoost Classifier

In this section, XGBoost gave the highest accuracy of 100%, the method requires an average of 32.81 ms to run.

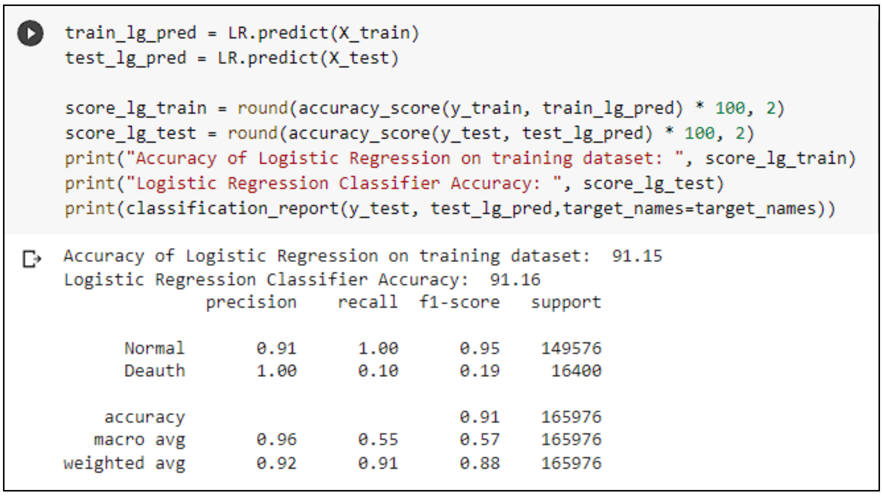
6.2.1 XGBoost: Confusion Matrix



```
cm = confusion_matrix(y_test, xgb_pred)
tn, fp, fn, tp = confusion_matrix(y_test, xgb_pred).ravel()
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=target_names)

disp.plot()
plt.show()
```

Figure 9: Confusion Matrix XGBoost

6.3 Experiment 3: Logistic Regression



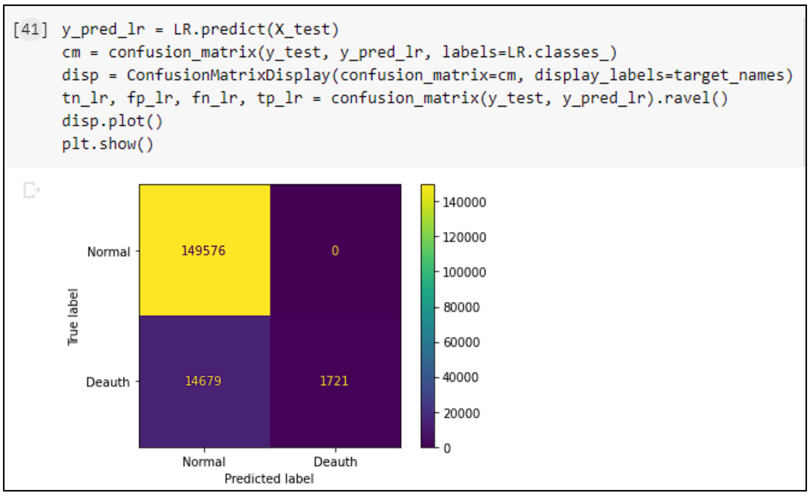
```
train_lg_pred = LR.predict(X_train)
test_lg_pred = LR.predict(X_test)

score_lg_train = round(accuracy_score(y_train, train_lg_pred) * 100, 2)
score_lg_test = round(accuracy_score(y_test, test_lg_pred) * 100, 2)
print("Accuracy of Logistic Regression on training dataset: ", score_lg_train)
print("Logistic Regression Classifier Accuracy: ", score_lg_test)
print(classification_report(y_test, test_lg_pred,target_names=target_names))
```

```
Accuracy of Logistic Regression on training dataset:  91.15
Logistic Regression Classifier Accuracy:  91.16
              precision    recall  f1-score   support

      Normal       0.91      1.00      0.95    149576
      Deauth       1.00      0.10      0.19     16400

    accuracy                           0.91    165976
   macro avg       0.96      0.55      0.57    165976
weighted avg       0.92      0.91      0.88    165976
```

Figure 10: Logistic Regression Classifier

Figure 10 demonstrates that the prediction accuracy was 91.16%, with recall score of 0.10 and F1-Score of 0.19. This indicates that our model has a larger likelihood of forecasting the correct outcome. The method requires an average of 4.65 ms to run.

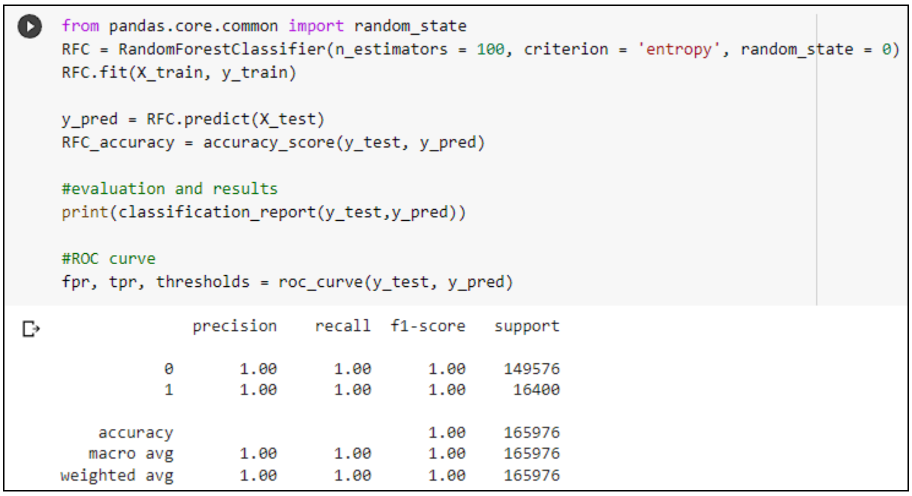
6.3.1 LR: Confusion Matrix



```
[41] y_pred_lr = LR.predict(X_test)
     cm = confusion_matrix(y_test, y_pred_lr, labels=LR.classes_)
     disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=target_names)
     tn_lr, fp_lr, fn_lr, tp_lr = confusion_matrix(y_test, y_pred_lr).ravel()
     disp.plot()
     plt.show()
```

Figure 11: Confusion Matrix LR

6.4 Experiment 4: Random Forest



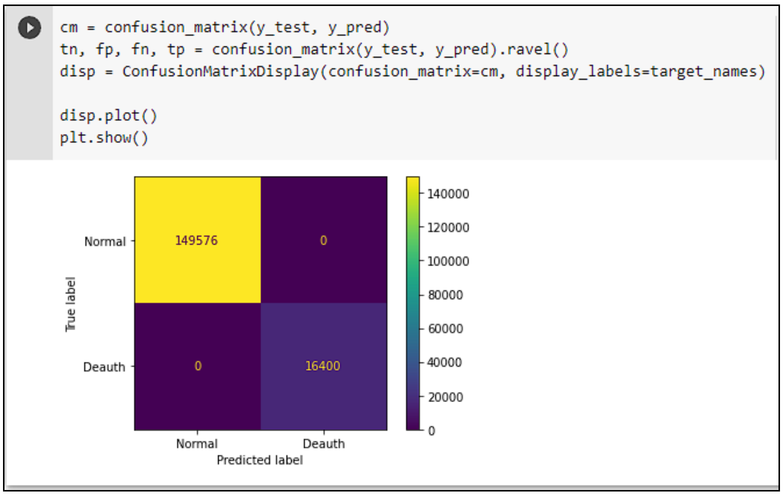
```
from pandas.core.common import random_state
RFC = RandomForestClassifier(n_estimators = 100, criterion = 'entropy', random_state = 0)
RFC.fit(X_train, y_train)

y_pred = RFC.predict(X_test)
RFC_accuracy = accuracy_score(y_test, y_pred)

#evaluation and results
print(classification_report(y_test,y_pred))

#ROC curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    149576
           1       1.00      1.00      1.00     16400

    accuracy                           1.00    165976
   macro avg       1.00      1.00      1.00    165976
weighted avg       1.00      1.00      1.00    165976
```

Figure 12: Random Forest Classifier

Figure 12 demonstrates that the prediction accuracy was 100% for the RF Classifier, the method requires an average of 14724.29 ms to run and has the highest latency compared to the other models.

6.4.1 RF: Confusion Matrix



```
cm = confusion_matrix(y_test, y_pred)
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=target_names)

disp.plot()
plt.show()
```

Figure 13: Confusion Matrix RF

6.5 Discussion

The computing power limitation and time-critical needs are significant problems with Raspberry Pi devices. Due to this, Latency is an important consideration for these devices, notwithstanding the conventional machine learning metrics previously discussed. As a result, Table 4 presents each algorithm's prediction in a test dataset at the execution time.

Table 4: Algorithm latency (100 loops each, 7 runs)

Algorithm	**Mean ± Std. Dev.**
Decision Tree	7.97 ms ± 0.068173
XGBoost	32.81 ms ± 0.975412
Random Forest	14724.29 ms ± 16985.79
Logistic Regression	4.65 ms ± 0.189335

In conclusion, the performance data provides an indicator of how well various algorithms may be able to categorise Deauth (DoS) attacks in residential Wi-Fi networks. Only two features — bytes and time delta — that may be derived from network traffic frames and packets are the subject of this study.

7 Conclusion and Future Work

It is a fact that the use of wireless networks brings its users numerous benefits, but one cannot forget to mention its disadvantages, security being one of them. Therefore, while we must take precautions to keep ourselves safe, it will require all of us to effectively safeguard the devices on which we rely.

This work addressed brief concepts of wireless networks, as well as IoT devices used in connection with the Wi-Fi network and demonstrated the vulnerable aspects through deauthentication attack. Although wireless networks have advantages over wired networks in terms of mobility and ease of equipment configuration, they are more susceptible to attacks, since it is not possible to fully control the range of the signal.

In this study, we provided different detection models for De-authentication attack in 802.11 Wi-Fi networks based on machine learning. On a portable Raspberry Pi running KALI Linux, aircrack-ng was used to fire de-authentication packets and Wireshark utility to capture the 802.11 Wi-Fi network traffic. The suggested algorithms on this research have a high detection rate and a low number of false positives for De-authentication attack.

For future work, we plan to embed one of the machine learning models in an IDS configured on a Raspberry PI, detecting Deauthentication attack in real time.

References

Assunção, M. F. A. (2002). *Guia do hacker brasileiro*. Marcos Flávio Araújo Assunção.

Augusto Filho, O. (2021). Securing Home Wi-Fi Networks.

Babar, S., Mahalle, P., Stango, A., Prasad, N. R., and Prasad, R. (2010). Proposed security model and threat taxonomy for the internet of things (iot). In Meghanathan, N., Boumerdassi, S., Chaki, N., and Nagamalai, D., editors, CNSA, volume 89 of *Communications in Computer and Information Science*, pages 420-429. Springer.

Bianchi, G. (2000). Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications*, *18*(3), 535-547.

Bishop, M. (2003). What is computer security?. *IEEE Security & Privacy*, *1*(1), 67-69.

Boeckl, K., Fagan, M., Fisher, W., Lefkovitz, N., Megas, K. N., Nadeau, E., ... & Scarfone, K. (2019). Considerations for Managing Privacy and Cybersecurity Risks in the Internet of Things (IoT).

Br, C. E. R. T. (2020). Security primer for Internet. *Available: http://cartilha. cert. b r [captured August 01, 2022].*

Brownlee, J. (2016). A gentle introduction to xgboost for applied machine learning. *Machine Learning Mastery*, 1-20.

Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE communications surveys & tutorials*, *18*(2), 1153-1176.

COMMUNITIES, C. (2008). *Future networks and the internet: Early challenges regarding the internet of things*. Technical report, CTEC.

Da Silva, F., Monteiro, C., de Boni, M., & Tolentino, C. (2014, July). Network Selection Architecture with Low Computational Resources Consumption for Mobile Devices. In *Anais do II Encontro Nacional de Computação dos Institutos Federais* (pp. 49-52). SBC.

De Magalhaes, G. G. (2016). Security study on the main protocols of the Internet of Things.

De Moraes, A. F. (2010). *Redes sem fio: Instalação, Configuração e Segurança-Fundamentos*. Saraiva Educação SA.

Deng, D. , Chen, X., Zhang, R., Lei, Z., Wang, X., & Zhou, F. (2021). XGraphBoost: extracting graph neural network-based features for better prediction of molecular properties. *Journal of chemical information and modeling*, *61*(6), 2697-2705.

Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on software engineering*, (2), 222-232.

Figueira, V. P. (2016). "Internet of things": a study on security, privacy and infrastructure issues.

Garber, L. (2000). Denial-of-Service Attacks Rip the Internet. *IEEE Computer*, 4(33):12-17.

Gates, C., & Taylor, C. (2006, September). Challenging the anomaly detection paradigm: A provocative discussion. In *Proceedings of the 2006 workshop on New security paradigms* (pp. 21-29)

Géron, A. (2017). Hands-on machine learning with scikit-learn and tensorflow: Concepts. *Tools, and Techniques to build intelligent systems*.

Gil, A. C. (2008). *Métodos e técnicas de pesquisa social*. 6. ed. Ediitora Atlas SA.

GONZALEZ, L. D. A. (2018). Logistic regression and its applications.

Guttman, B., & Roback, E. A. (1995). Sp 800-12. an introduction to computer security: The nist handbook.

Haykin, S., Xue, Y., & Davidson, T. N. (2008, October). Optimal waveform design for cognitive radar. In *2008 42nd Asilomar Conference on Signals, Systems and Computers* (pp. 3-7). IEEE.

Hassija, V., Chamola, V., Saxena, V., Jain, D. , Goyal, P., & Sikdar, B. (2019). A survey on IoT security: application areas, security threats, and solution architectures. *IEEE Access*, *7*, 82721-82743.

Henke, M., Santos, C., Nunan, E., Feitosa, E., dos Santos, E., & Souto, E. (2011). Machine learning for security in computer networks: Methods and applications. *Minicursos do XI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg 2011)*, *1*, 53-103.

Jain, A. (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res*, *12*, 2825-2830.

Jobstraibizer, F. (2010). *Unraveling wireless networks*. Universo dos Livros Editora.

Kaspersky, E., & Furnell, S. (2014). A security education Q&A. *Information Management & Computer Security.*

Kombo, C. A. (2019) Intrusion monitoring, attack simulation of a honeypot server in a controlled environment.

Kumar, P. A. R., Selvakumar, S. (2011). "Distributed Denial of Service Attack Detection Using an Ensemble of Neural Classifier", *Computer Communication*, Vol. 34, pp. 1328-1341.

Kuncheva, L. (2004). Combining pattern classifiers methods and algorithms. john wiley&sons. *Inc. Publication, Hoboken*.

Kurose, J. F., & Ross, K. W. (2005). Reti di calcolatori e Internet.

Landwehr, C. E. (1981). Formal models for computer security. *ACM Computing Surveys (CSUR)*, *13*(3), 247-278.

Landwehr, C. E. (2001). Computer security. *International journal of information security*, *1*(1), 3-13.

Likarish, P., Jung, E., & Jo, I. (2009, October). Obfuscated malicious javascript detection using classification techniques. In *2009 4th International Conference on Malicious and Unwanted Software (MALWARE)* (pp. 47-54). IEEE.

Lima, J. T., & Panham, A. (2019). Risk management in smart home -Smart Home projects. *Scientific journal e-Locution*, *1*(16), 21-21.

Lorenzett, C. D. C., & Telöcken, A. V. (2016). Comparative Study between Random Forest and J48 Data Mining algorithms in Decision Making. In Symposium on Research and Development in Computing (SPDC), vol. 2, no. 1.

Magrani, E. (2018). *The internet of things*. Editora FGV.

Mateti, P. (2005). Hacking Techniques in Wireless Networks: Forged Deauthentication. *Department of Computer Science and Engineering, Wright State University*.

McCullagh, D. (2013). Feds push web companies for encryption master keys. *CNET* .

Mirkovic, J., Dietrich, S., Dittrich, D., & Reiher, P. (2004). *Internet denial of service: attack and defense mechanisms (Radia Perlman Computer Networking and Security)*. Prentice Hall PTR.

Misra, S., Li, H., & He, J. (2020). Noninvasive fracture characterization based on the classification of sonic wave travel times. *Machine Learning for Subsurface Characterization*, 243-287.

Mitchell, T. M. (1997). Does machine learning really work? *AI magazine*, *18*(3), 11-11.

Moreno, D. (2016). *Pentest in wireless networks*. Novatec Editora.

Nakamura, E. T., & de Geus, P. L. (2007). *Network security in cooperative environments*. Novatec Editora.

Prodanov, C. C., & De Freitas, E. C. (2013). *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição*. Editora Feevale.

Santos, B. P., Silva, L. A., Celes, C. S., Borges Neto, J. B., Peres, B. S., Vieira, M. A. M., & Loureiro, A. A. (2016). Internet of things: from theory to practice.

Silva, D. F. B. F. D. (2021). *Pré-processamento de Dados e Comparação entre Algoritmos de Machine Learning para a Análise Preditiva de Falhas em Linhas de Produção para o Controlo* (Doctoral dissertation).

Spafford, G., Schwartz, A., & Garfinkel, S. (2003). *Practical Unix and Internet Security*. O'Reilly.

Stoneburner, G., Goguen, A., & Feringa, A. (2001). Special Publication 800-30. *Risk Management Guide, NIST.*

Wadlow, T. A. (2000). Segurança de Redes: Projeto e gerenciamento de redes seguras. *Rio de Janeiro: Campus*.
Wright, J. D. (2005). The constitutional failure of gang databases. *Stan. JCR & CL*, *2*, 115.