MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
Federal State Autonomous Educational Institution of Higher Education
“South Ural State University (National Research University)”

School of Electronic Engineering and Computer Science
Department of Computer Engineering

THESIS IS CHECKED
Reviewer,
_____
_____ _. _. _____
“___”_____ 2022 г.

ACCEPTED FOR THE DEFENSE
Head of the department,
Ph.D., Associate Professor
_____ D.V. Topolsky
“___”_____ 2022 г.

Misuse attack detection based on data mining in network functions virtualization

GRADUATE QUALIFICATION WORK
SUSU – 09.04.01.2022.308-643.GQW

Supervisor,
PhD, Associate Professor
_____ D.V. Topolsky
“___”_____2022 г.

Author,
student of the group: CE-228
_____ N.J. AL-Dulaimi
“___”_____2022 г.

Normative control
_____ S.V. Siaskov
“___”_____2022 г.

Chelyabinsk-2022

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
Federal State Autonomous Educational Institution of Higher Education
“South Ural State University (National Research University)”

School of Electronic Engineering and Computer Science
Department of Computer Engineering

ACCEPTED FOR THE DEFENSE
Head of the department,
Ph.D., Associate Professor
_____ D.V. Topolsky
“___”_____ 2022 г.

TASK

of the master graduate qualification work

for the student of the group CE-228
AL-Dulaimi Nebras Jalel Ibrahim
in master direction 09.04.01
“Fundamental Informatics and Information Technologies”
(master program “Internet of Things”)

The topic (approved by the order of the rector from 24.5.2022): “Misuse attack detection based on data mining in network functions virtualization”.

The deadline for the completion of the work: 01.06.2022.

The source data for the work:

3.1. Windows 10 .

3.2. We used the Windows 10 operating system environment because it is the most common and used, the ease of installing the Python language and making the proposed model easy to use and handle by the user.

3.2. The CAIDA “DDoS Attack 2007” Dataset. [Electronic Resource] URL: https://www.caida.org/catalog/datasets/ddos-20070804_dataset/.

3.3. LaTeX, MathCad® syntaxis, MathML, Wolfram Mathematica®.

3.4. We have used the Menedely as references management application.

The list of the development issues:

4.1. To deploy a library that provides Network Function Virtualization (NFV) capability to combat misuse attacks for IoT networks.

4.2. To develop a library that runs on devices that consume low power and have low processing capabilities.

4.3. To deploy a library that trains data mining model on computers with minimal consumption of energy and memory and increased execution speed.

4.4. To employ libraries that apply data mining methods to provide secure IoT data by early and accurate identify of the misuse attack to avoid their consequences.

4.5. To test the library and give an example of how to implement interfaces.

4.6. Conduct experiments to examine the effectiveness of the implemented system and compare the results with other existing methods.

Issuance date of the task: 25.12.2021.

Supervisor _____/ *D.V. Topolsky* /

Student _____/ *N. J. AL-Dulaimi* /

CALENDAR PLAN

Phase	Deadline	Supervisor's signature
Introduction and literature review	10.03.2022	
Development of the model, design of the system	21.03.2022	
Implementation of a system	04.04.2022	
Testing and debugging of the system, experiments	25.04.2022	
Full text, normative control	16.05.2022	
Proposal defense	24.05.2022	

Supervisor _____/ *D.V. Topolsky* /

Student _____/ N. J. AL-Dulaimi /

Acknowledgements

I would like to thank God, for letting me through all the difficulties .for giving me this opportunity to finish my studies successfully.

I would like to acknowledge and give my warmest thanks My supervisor Dmitry Topolsky has been very instrumental, for his excellent comments and his constant support during the work, my department has always been open and available to solving all my educational and non-educational problems. My university has given me a conducive learning environment that I cannot take for granted.

I would also like to thank my husband , my children and my family as a whole for their continuous support and understanding when undertaking my research and writing my project.

Lastly, I would lastly like to thank the Russian Government .

Annotation

N. J. AL-Dulaimi. Misuse attack detection based on data mining in network functions virtualization – Chelyabinsk: SUSU; 2022, 50 p., 20 pic., bibl. – 24.

This thesis consists of six main chapters: Introduction, definition of requirements, design proposed system, implementation of the interfaces, testing methodology, conclusion, and references.

In the first chapter, we will have the subject area analysis briefly then have an overview of analogues and the main technological solutions that I will use will be featured. All the different software platforms to be used will be adequately described.

In the second chapter, there is a description of the data set used in this work and the technique of statistical analysis. This chapter also includes an explanation of machine learning algorithms in detail and the accuracy metrics used to assess the performance of the detection model.

In the third chapter will describe the design and implementation of the software and how the different components will interact with each other as well as the algorithms for tackling the problem and detected the misuse attack.

In the fourth chapter, the implementation of interface for all important python library such as TensorFlow and Pandas library and implementation of the training and testing interface.

In the fifth chapter, the results of each step of the proposed system will be presented, testing the machine learning models and comparing the performance of these algorithms based on accuracy measures.

Finally, in the sixth chapter we will have a conclusion for the thesis, with future improvements to the solution being discussed as well as opportunities.

CONTENTS

INTRODUCTION

IoT technology provides solutions to creating a smarter environment that saves time, reduces energy consumption, and reduces cost [1]. Network Functions Virtualization (NFV) is an essential technology to avoid fundamental alterations in the actual physical components of network systems via supplying network functions by implementing pure software instead of hardware resources by capable of sharing the available resources and inning concurrently on infrastructure via virtualization [2].

Misuse attack can be done in NFV network resources, especially with resources that cannot be shared between multi-user, or the resources that can be shared for a limited number of users, in this attack the attacker free up the resources from other users and using them for its benefits and using it for without sharing it with other users, this attack usually case bottleneck problem, which leads to service delay or even services down of NFV network [3].

There are several advantages to identification model for Misuse attack in NFV [4][5]:

- NFV has many advantages such as; optimizing resource consumption, saving investment cost, increasing operational efficiency, and facilitating network service lifecycle management;

- NFV is not safe from attacks. Since all parts of this NFV network share the same resources, misuse attack is the most common and dangers NFV attack. Therefore, a model with the ability to accurately detect a misuse attack must be built to maintain network resources and security;
- machine learning algorithms can automatically and accurately detect the main differences between normal and aberrant data. Furthermore, because machine learning methods are very generalizable, they can detect previously unknown attacks.

Several firms have also made available various Misuse attack classification algorithms, databases, and templates. These services enable developers to add products like network protection programs from Misuse attacks to prevent an attacker from consuming the resources of the NFV environment by sending the large number of traffic. Consider a cloud that receives a large number of requests to access the services it provides. It will be as easy as using smart systems capable of identifying whether the request is normal or an attacker to direct the cloud by running the appropriate protection programs against the attacker.

The purpose of the research:

To create an efficient model that allows you to run data mining algorithms to detect malicious attacks on Network Functions Virtualization (NFV) is the purpose of the work.

Tasks necessary to achieve the goal:

1. Deploy a library that provides network function virtualization (NFV) capability to combat IoT network misuse attacks.
2. Develop a library that runs on low power, low computing devices.
3. Deploy a data mining model training library to computers with minimal power and memory consumption and increased execution speed.

4. Leverage libraries that use data mining techniques to secure IoT data by early and accurately detecting a misuse attack to mitigate its consequences.
5. Test the library and provide an example implementation of the interfaces.
6. Conduct experiments to test the effectiveness of the implemented system and compare the results with other existing methods.

Structure of the Thesis

This thesis consists of six main chapters: Introduction, definition of requirements, design proposed system, implementation of the interfaces, testing methodology, conclusion, and references.

In the first chapter, we will have the subject area analysis briefly then have an overview of analogues and the main technological solutions that I will use will be featured. All the different software platforms to be used will be adequately described.

In the second chapter, there is a description of the data set used in this work and the technique of statistical analysis. This chapter also includes an explanation of machine learning algorithms in detail and the accuracy metrics used to assess the performance of the detection model.

In the third chapter will describe the design and implementation of the software and how the different components will interact with each other as well as the algorithms for tackling the problem and detected the misuse attack.

In the fourth chapter, the implementation of interface for all important python library such as TensorFlow and pandas library and implementation of the training and testing interface.

In the fifth chapter, the results of each step of the proposed system will be presented, testing the machine learning models and comparing the performance of these algorithms based on accuracy measures.

Finally, in the sixth chapter we will have a conclusion for the thesis, with future improvements to the solution being discussed as well as opportunities.

1. SUBJECT AREA ANALYSIS

Network Function Virtualization (NFV) represents a virtual network whose service is provided by virtual parts of virtual machines. This type of network is simple to execute and update. In addition, NFV leading to low cost due to sharing the same resources. As with other networks, NFV is not safe from attacks. Since all parts of this NFV network share the same resources, misuse attack is regarded to be the most common attack in NFVs, particularly because the attack use one or more of the resources which affect all parts of the NFV [6].

Machine Learning (ML) is classification methods which has the capability of automatically learning and improving through experience, with no need for explicit programming. In other words, Machine Learning is an application which could access data and make use of it learn for self-learning [6].

To do Misuse attack using Machine learning, a programmer should feed data into learning algorithm that discovers the rules in the data, which then builds a model based on the data provided through a process called training and finally data is then run through this model to make classifications, a process called inference [2].

The past few years there are various methodologies that are utilized for the detection and classification of Misuse attack based on machine learning in NFV environments. All these methods use data of the NFV attacks in their proposed models as will be discussed below. In most cases, deep learning or machine learning is used to build systems capable of detecting and identifying NFV attacks, but these systems need to improve accuracy, reduce complexity in computation, and increase execution speed [6].

This thesis will therefore focus on training model using data mining method to extract rules of misuse attack detections. When testing the propose work with a server traffic data having more than 5 million network connections with higher accuracy and faster detection.

1.1. OVERVIEW OF ANALOGUES

1.1.1. Parallel Misuse and Anomaly Detection Model

Goel R. et al. [7] present a hybrid model using C4.5 based binary decision trees are employed misuse and CBA (Classification Based Association) based classifier is used to detect anomalies.Tree C4.5 is efficient, powerful, and popular ML classification method .C4.5 consists of two processes; preparation of decision tree and make the rules (structure and design). Then, compute entropy and information gain with the highest attribute is selected. CBA is classification based on association, also called associative classification, is the application of association rules to classification problems. It generates class association rules (CARs). Classification association rules (CARs) are association rules with the target class on the right hand side of the rules. The important features are:

- the C4.5 based decision tree separates the network traffic into normal and attack categories. The normal traffic is sent to anomaly detector and parallel attacks are sent to a decision trees based classifier for labelling with specific attack type;

- the CBA based anomaly detection is a single level classifier whereas the decision trees based misuse detector is a sequential multilevel classifier which labels one attack at a time in a step by step manner;
- the C4.5 –CBA model is trained and tested on two disjoint datasets provided in the KDD Cup 99;
- the C4.5 –CBA model is implementation using Weka 3 software in Java [7].

1.1.2. Self-Adaptive Misuse Attack Detection Model

Self-Taught Learning (STL) is a machine learning framework which is able to exploit unlabeled data with the purpose of improving a supervised classification problem. To design a self-adaptive and autonomous misuse intrusion detection system, Papamartzivanos et al. [8] propose the use of auto-encoder techniques. Specifically, the proposed system is based in four phases, including 1) Monitor, 2) Analyze, 3) Plan, 4) Execute, and 5) Knowledge. The monitor phase determines any alteration event that requires an intrusion detection system adaptation. The analyze phase uses network audit tools (e.g., Argus and CICFlowMeter) to perform the transformation of the raw network traffic into network flows. The plan phase uses a sparse autoencoder to learn representations of the input data. The execute phase is accountable for storing purposes. The important features are:

- Self-Adaptive Misuse Network Intrusion Detection Systems is a combination of Self-taught learning and MAPE-K brings together the benefits of transfer learning from unlabeled data and places this ability in the frame of autonomic computing;
- the study uses two datasets in performance evaluation, including KDDCup’99 and NSLKDD;

- the self-adaptation property implies that an IDS should be capable to adapt itself to the needs of a new environment even without the need of feedback from the administrators;
- the C4.5 –CBA model is implementation using Tensorflow software in python language;
- improving the management of Misuse in the intrusion detection systems (IDSs) [8].

1.1.3. Extract Rules of Misuse Attack Detections Model

Ali K.A. and Bhaya W.S. in [9] using machine learning to extract rules of misuse attack detections. The tree decision C4.5 algorithms has been used to extract these rules, with nine features of network data flow. When testing the propose work with a server traffic data having more than 5 million network connections for the Network Function Virtualization (NFV) which represents a virtual network whose service is provided by virtual parts of virtual machines. The important features are:

- the Extract rules of misuse attack detection is simple to execute and update since it's using NFV environment. In addition, NFV leading to low cost due to sharing the same resources;
- this application with stored data traffic of network attacks can be very helpful to analyze this sort of attack, as well as to extract the rules for discovering any anomaly behavior of NFV when undergoing;
- nine features have been considered, and this number of features helped to produce high quality results;
- TansorFlow is using for the training dataset (around 4.5 million out of 5 million connections) [9].

After comparing between a parallel misuse and anomaly detection model, Self-Adaptive misuse detection model, and extract rules of misuse attack detections model on accuracy value and dataset type, I found out what are their weaknesses are and it helped me to build up on my proposed solution.

First, when on comparing a parallel misuse and anomaly detection model, Self-Adaptive misuse detection model, and extract rules of misuse attack detections models, all these models are quite similar in their offering: misuse attack detection using many different machine learning methods with a good performance. The extract rules of misuse attack detections model continually update their language models to increase accuracy of the detection misuse attack. This continuous development is a strong point for this model, especially selection the important features have been considered, and this number of features helped to produce high quality results. This model has complex operation for found relationship between all feature and select the best of them. However, the performance of the misuse attack detection model will be less accurate if features are selected that do not directly specify the sort of attack [10].

On the other hand, Extract Rules of Misuse Attack Detections Model which is a decision tree C4.5 algorithm of machine learning has been used to analyses a large data of server traffic for a NFV network as well as to extract rules from this data, other contributors so far have a lot of flexibility, especially since it’s employ in IoT devices and can be extended or improved by anyone who dares understand it. It does carry a lot of complexity however, requiring a lot of time and effort to fully learn the quirks.

Using classification accuracy rate, which compares a reference with a hypothesis and is given by (TP+TN / TP+TN+FP+FN) *100, Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives, Parallel Misuse and Anomaly Detection Model has accuracy rate 97.4% [7] , Self-Adaptive Misuse

detection model has accuracy rate 73.37% [8] and extract rules of misuse attack detections has accuracy rate 96.00% [9].

I also compared them in terms of the detection method of misuse attack for instance on detection of misuse in NFV network traffic using C4.5 decision tree method, C4.5 decision tree will be constructed based on one or more factors. Although increasing these factors will increase the complexity of the problem, it usually leads to relatively on accurate results.

Table 1 – Comparison of analogues

Feature	Parallel Misuse and Anomaly Detection Model	Self-Adaptive Misuse Detection Model	Extract Rules of Misuse Attack Detections Models
Language support	Weka 3 software in Java	Weka software in Java	TensorFlow software in python
Dataset Name	KDD Cup 99	KDD Cup'99 & NSLKDD	KDD Cup 99
Detection model	C4.5+CBA (Classification Based Association (CBA))	Self-adaptive and autonomous misuse IDS	C4.5
Domaine of Data	Misuse IDs	Misuse IDs	Misuse Traffic in NFV
Accuracy, %	97.40	77.37	96.00

1.2. ANALYSIS OF THE MAIN TECHNOLOGICAL SOLUTIONS

1.2.1. CAID-DDoS Attack Dataset

The dataset that used in this work called The CAIDA- DDoS Attack 2007 Dataset. The Center for Applied Internet Data Analysis (CAIDA) and a research unit at the University of California San Diego (UCSD) made this dataset available to researchers, and it is managed by The Regents of the University of California [11].

This dataset contains roughly one hour of anonymized traffic traces from a CIAD-DDoS attack on August 4, 2007. This form of denial-of-service attack seeks to prevent access to the targeted server by consuming all computing resources on the server as well as all network bandwidth connecting the server to the Internet. The one-hour trace is divided into 5-minute pcap files. The dataset has a total size of 21 GB. Only attack traffic on the victim and replies to the attack from the victim are included in the traces. Non-attack traffic has as much as possible been removed. A CIAD-DDoS attack dataset include 42 features with 1,072,017 connection of misuse attacks and the remaining are belonging to DDoS traffics and normal traffics [11].

The advantage of this dataset is available for public use -effective to handle large DDoS attacks above 5 Gb -traces can be read on any software reading tcpdump. The tcpdump is a data-network packet analyzer computer program that runs under a command-line interface and the dis-advantage is a non-attack traffic is unavailable does not include payload packets [12].

1.2.2. TensorFlow

TensorFlow is an open-source end-to-end platform for creating Machine Learning applications. It is a symbolic math library that employs dataflow and differentiable programming to accomplish various tasks focused on inference of deep neural networks and training. It enables developers to build machine learning applications by utilizing a variety of tools, libraries, and community resources. [12].

TensorFlow is the best library of all because it is built to be accessible for everyone. TensorFlow library includes different API for building large-scale machine learning architectures such as random forest and linear support vector classification.

TensorFlow is based on graph computation and It enables the developer to visualize the neural network building with Tensor board. This tool is useful for debugging the software. Finally, TensorFlow is built for large-scale deployment. It runs on CPU and GPU.

To offer a concrete example, AI can help Google users do faster and more refined searches. When a user enters a keyword into the search field, Google suggests what the following word may be [13].

Some supported TensorFlow algorithms include:

- Linear regression;
- Classification;
- Deep learning classification;
- Deep learning wipe and deep;
- Booster tree regression;
- Boosted tree classification [14].

1.2.3. Pandas

The Pandas library, under development since 2008, is intended to close the gap in the richness of available data analysis tools between Python, a general purpose systems and scientific computing language, and the numerous domain specific statistical computing platforms and database languages. The library's name derives from panel data, a common term for multidimensional data sets encountered in statistics and econometrics [15]. The main advantages of Pandas library in python are:

- Pandas provide extremely streamlined forms of data representation;
- Pandas helps to shorten the procedure of handling data;
- Pandas provides program with a huge set of important commands and features which are used to easily analyze input data;
- efficiently handles large data;

- makes data flexible and customizable [15].

1.2.4. NumPy Array

The NumPy array is a data structure that efficiently stores and accesses multidimensional arrays (also known as tensors), and enables a wide variety of scientific computation. It consists of a pointer to memory, along with metadata used to interpret the data stored there, notably “data type”, “shape” and “strides” [16].

The data type describes the nature of elements stored in an array. An array has a single data type, and each element of an array occupies the same number of bytes in memory. The shape of an array determines the number of elements along each axis, and the number of axes is the dimensionality of the array [16].

The main advantages of NumPy array library in python are [15]:

- consumes less memory;
- faster as compared to the python List;
- convenient to use.

1.2.5. Programming Technologies

For this project we will use Python as our main programming languages for development. Development that tools are required to develop and test/debug the code include:

- compiler;
- debugger.

1.3. CONCLUSION

The following components are being used for this project. It is urgent and under development right now:

- CAID-DDoS Attack Dataset;
- TensorFlow;

- Pandas;
- NumPy Array;
- Programming Technologies.

2. DEFINITION OF REQUIREMENTS

2.1. FUNCTIONAL REQUIREMENTS

2.1.1. Misuse attack detection security

The misuse attack detection model is a model that uses data mining techniques and methods to improve the security of IoT data in NFV environments. In the case of the cloud, it contains several sensors that receive millions of requests from different parties. When abnormal behavior occurs, the misuse attack detection model will stop the current session and start an analysis study. It varies from the anomaly detecting method, which begins with regular device behavior and then labels all other behavior as irregular. When it comes to misuse identification, something that isn't identified is deemed natural. The usage of attack signatures in an intrusion prevention framework is an illustration of misuse detection. The word "misuse detection" has also been extended to all forms of device misuse.

2.1.2. CAID-DDoS Attack Dataset based on IoT Sensors

CAID-DDoS Attack Dataset is a public online database that has been collected using smart sensors. Sensors are used to track the functioning of devices connected to the Internet of Things in IoT data collection. Smart sensors monitor the state of the Internet of Things network that includes NFV by collecting information on each request in real-time to store or retrieve data from servers in the virtual network. IoT data collection typically involves capturing huge amounts of data streams such as collected data from cloud era, information management has a major role to balance the preservation of data's confidentiality, honesty, and availability (also known as the CIA triad) while concentrating on effective policy execution without undermining company efficiency. Furthermore, several types of attack could affect information security such as misuse, DDoS, SQL Injection, and others else. However, the most common and dangerous one them is the Misuse attack.

2.1.3. Internet independence

The combination between NFV and machine learning model used by three effective data mining methods to detection misuse attack should run on the IoT devices offline.

2.1.4. Noise

Misuse detection model should be able to work well in noisy environments by automatically identification and filtering out the noise.

2.1.5. Processor power

Misuse detection model should run on an edge device with a 64-bit Intel® Core™ i7-8565U microprocessor running at 1.80 GHz of program memory and 8.00GB RAM.

2.2. Core Requirements

2.2.1. Misuse detection model shall be able to classify attack in Real time.

2.2.2. Misuse detection model shall be compatible with different operating system such as Windows 10 Pro.

2.2.3. Misuse attack detection shall be easy to use by developers through bootstrapping of all the required code and shall have comments.

2.2.4. Misuse attack prototype used for this thesis will be limited to CAID-DDoS Attack Dataset to assess how effective the approach taken is, then be updated, by using a new dataset including NFV attacks.

2.3. Misuse attack detection Requirements

The following guidelines are specific to misuse attack detection:

2.3.1. Misuse attack detection SHALL use only The CAIDA 2007 dataset contains approximately one hour of anonymized traffic traces from Misuse attack and Distributed Denial-of-Service (DDoS) attack.

2.3.2. Misuse attack detection SHALL provide output to indicate the higher accuracy rate of detection of the misuse attack.

2.3.3. Misuse attack detection shall recognize misuse attacks in NFV only for the scope of this thesis.

2.3.4. The dataset that is used in Misuse attack detection stored more than 5 million clients' connections. Each of these connections contained 14 field of information. About 90% of these connections were utilized as training data, whereas the remaining 10% was used as testing data.

2.4. Documentation Requirements

Documentation of this proposed model will be delivered in various formats including pdf, ppt, html and docx and will be English language. This documentation will be the user manual for "misuse attack detection" outlining its various features and how to use the detection model for programming development.

2.5. Conclusion

In this chapter, we defined the various core and misuse attack requirements for misuse attack detection. We also described how the various software components within it will interact with each other. Lastly, we gave the documentation requirements that described among other things the user manual and its contents.

3. DESIGN AND IMPLEMENTATION

3.1. ARCHITECTURE OF THE PROPOSED SOLUTION

The misuse attack detection model is composed of various components, as shown in the image below. First, the sensor device receives different attacks input then using One –Hot Encoding extracts important features while simultaneously reducing the magnitude of features that do not directly indicate the class of attack then using Standardization scales each input feature separately to removing any noise from the data. The inference is then run on the features outputting class probabilities as shown in the diagram below.

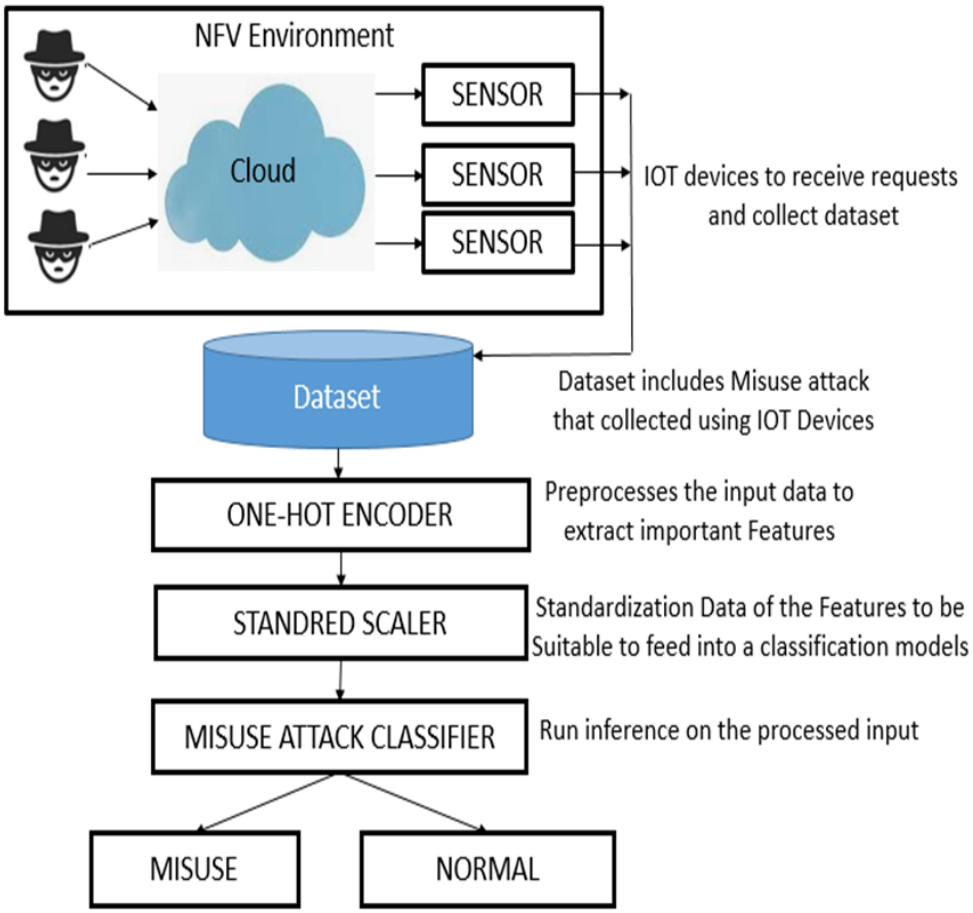

Figure 1 – Misuse Attack Detection Architecture

The model used in this thesis is trained to recognize the misuse attack and normal attack. The model takes data from IOT sensors devices and outputs the probability scores. This data is consumed by the model in terms of analysis and building of classification models.

3.2. ALGORITHMS FOR SOLVING THE PROBLEM

3.2.1. Misuse Attack as Data

The first step to do misuse attack classification for our dataset will be to extract features. In this work, One Hot Encoding is used for dataset preprocessing. The most used coding scheme is One Hot Coding. It compares each category variable level to a fixed reference level. One hot encoding transforms a single variable with n observations and d distinct values, to d binary variables with n observations each. Each observation showing whether the dichotomous binary variable is present (1) or absent (0) [19].

Let assume [0, 0, 0, 1, and 0] will be a valid One Hot Encoding, it would tell you the taxonomy in location 3 or 4 in array indexing is the taxonomy of the object. In invert, [0, 1, 0, 1, 0], and [1, 1, 1, 1, 1] are examples of unacceptable One Hot Encodings. However, it is too difficult to obtain anomaly causing data mostly when these flows are sensible to be used in real attacks like in the case of the Misuse attack. However, the important thing of mechanism is to dealing with the Misuse attacks by using the flows recognized of traffics.

3.2.2. Stander Scaler Data

The second step to do misuse attack classification is stander scaler data in order to define each class within a specific range, like between 0 and 1 or between -1 and +1. Another scaling strategy is standardization, in which the values are centered around the mean and have a unit standard deviation. This indicates that the attribute's mean becomes zero, and the resulting distribution has a unit standard deviation [20].To normalize your data, you need to import the MinMaxScalar from the sklearn library and apply it to our dataset.

3.2.3. Misuse Attack Classification

3.2.3.1. Random Forest Model

A Random Forest is a classifier that consists of several different tree classifiers {h (x, θ _k) k = 1,2,….}, where Θ k are irregularly distributed scattering vectors and each tree selects a unit for the common class at the input x [21]. A Random Forest is created by arbitrating a set of trees.

Breiman used the following steps to create each tree in a Random Forest: if N is the size of the files in the prep set, at this point, N files are displayed especially but by division, from the initial information; this is a bootstrap test. This example will be a preparation for the development of the tree. In cases where there are input elements M, the number m << M is chosen with the ultimate goal that m elements are randomly selected in M in each position, and the best distribution of these m units is to separate the center. The evaluation of m in the background field development is maintained [21].

Thus, many trees are created in the background forest; Ntree size preselects the size of the tree. The amount of variables items (m) selected in each center is indicated as m, t, r, y in the item. The size of the perimeter (for example, the number of events in the leaves, nodes), which is normally set to one, will limit the tree's depth. When the forest is ready or designed as above, to organize another event, it is run with all the trees that are filled. The new event is allocated to each tree, and it is registered as a vote [22]. All of the trees' votes are put together, and the class with the most votes e.g, the largest share of votes is presented as a description of the new event [21].

In the misuse classification model based on Random Forest is follow several steps:

Step 1: Splitting dataset into 70% training and 30% testing;

Step 2: In Random forest n number of random records are taken from the data set having k number of records;

Step 3: Individual decision trees are constructed for each sample;

Step 4: Each decision tree will generate an output;

Step 5: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

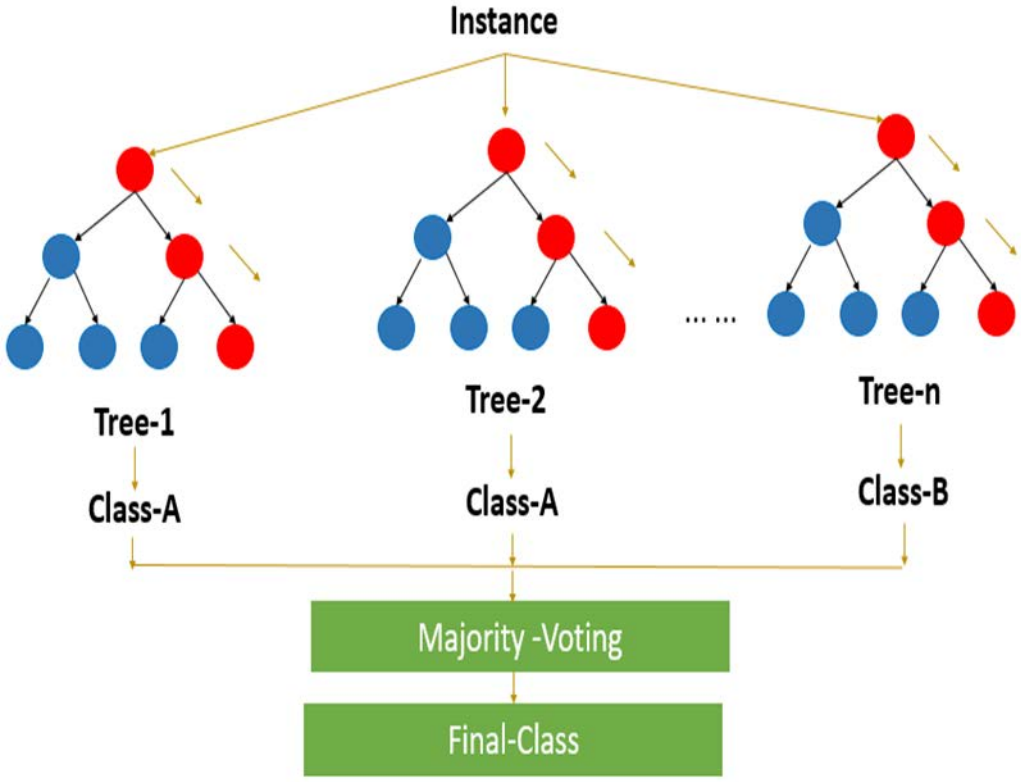

Figure 2 – Workflow of the Random Forest Model

3.2.3.2. C4.5 Decision Tree Model

Root, branches, and leaves make up a traditional tree. Decision Tree follows the same form. There are root nodes, divisions, and leaf nodes in it. Any internal node checks an attribute, the outcome of the evaluation is on the branch, and the class mark as a result is on the leaf node [23]. A root node is the topmost node in a Tree and acts as the parent of all nodes. A decision tree is a tree in which each node (attribute) represents a function, each connection (branch) represents a decision (rule), and each leaf represents an outcome (categorical or continuous value). Since decision trees are built to imitate human reasoning, grabbing data and creating good interpretations is a breeze.

The aim is to construct a tree like this with all of the data and process a single result at each leaf [24]. To allow simple distinctions among the different alternatives, the decision tree allows clear all potential alternatives and tracks each alternative to its conclusion in a single view. One of the greatest characteristics of Decision Tree is its clarity. Another important value is the right to pick the most skewed function and the essence of comprehensibility. It's also easy to categories and grasp [23].

The decision tree algorithm use tree representation to solve the problem. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label. Decision Tree operates on the Sum of Product (SOP) form, commonly known as Disjunctive Normal Form. The primary challenge in Decision Tree is identifying the attribute for the root node in each level. This procedure is known as attribute selection. We have two major measures for attribute selection: Gini index, Information Gain see figure 3.

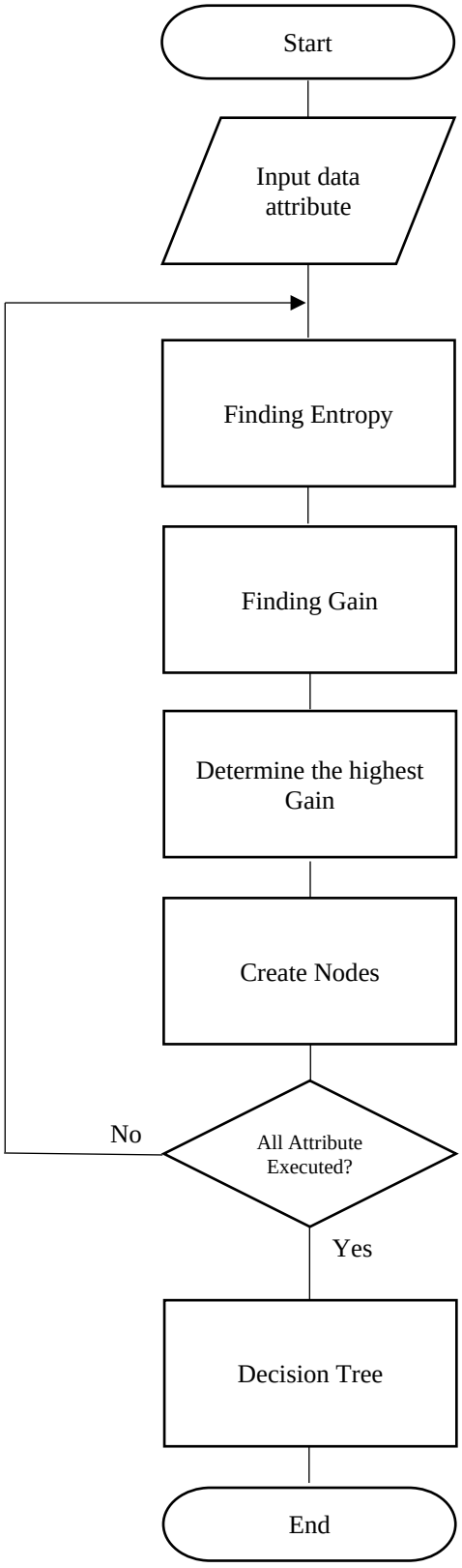

Figure 3 – Workflow of the C4.5 Decision Tree Model

3.2.3.3. Linear Support Vector Classification Model

It is a binary classification that is used for the categorizing of the attacks. If we merge the binary classifier with the decision tree algorithm then we have multi class S Linear SVC. With the help of multi class Linear SVC we can classify attacks of different class [4]. Linear SVC uses nonlinear mapping that maps the real values into higher dimensional feature space. Linear separating hyper plane is used by Linear SVC for the creation of classifier. Through the use of hyper-plane SVC separate the data into different classes. There is an attribute that is called as kernel that Linear SVC uses for solving the problem. User has to provide the kernel function at the training phase of the algorithm. With the help of support vectors, V does the classification. There are many kernel functions like linear, radial basis functions, polynomial, sigmoid [23]. In the Figure 4 given below, the distance between the data and hyper plane is revealed. In the left image, the distance among the data and the hyper plane is small and in the right image the space is larger, which makes classification easy.

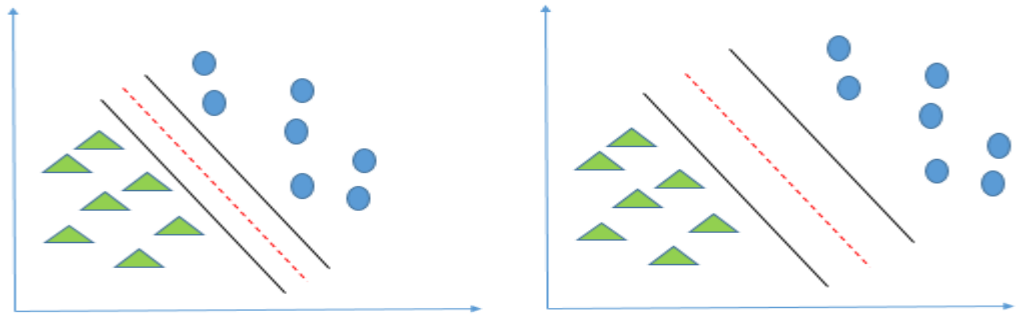

Figure 4 – Hyperplane of Linear SVC Separating the two different Classes

3.2.3.4. Classification

The inputs are flatter one-dimension feature vector for instance then fed to a random forest, C4.5, and Linear SVC to every iteration of training since the input instance has already been converted into a suitable form using one-hot encoder technique. These models are able to distinguish between misuse attack and normal attack classes in instance and classifies them using the machine learning classification technique.

3.3. DESCRIPTION OF DATA

3.3.1. Data Collection

The Network Function Virtualization NFV is network that used in the Internet of Things IoT, so in this project will be used IoT devices linked with group of the sensor to collecting hug data. When the user requests access to the cloud to download or retrieve data, the sensors connected to the cloud will record data about this request and store this data in a dataset called CAID-DDoS attack [17] [18]. These datasets can be utilized in the detection of several types of attacks. Also, the CAID-DDoS dataset contain more than 5 million requests connections. Each one of these connections contained 42 field of information. This dataset has 1,072,017 connection of misuse attacks, and the remaining are belonging to DDoS traffics and normal traffics.

3.3.2. Data Usage

After the primitive data is run through interface model, probabilities for the two classes of data is then computed and displayed in the interface of the model. At this point a command classifier could be programmed to undertake a certain command which the command responder will execute.

3.3.3. Data Storage

The IoT sensor device are generating and storing a various metadata and stored for the purposes of device detects and service improvements. request data inputs being the core piece of Misuse attack detection data, is however not sored because of the size of devices that misuse attack detection is targeting. Attack is processed through the machine learning algorithms and then to extract the user's request. These system employs machine learning techniques to continuously improve itself with each input.

3.3.4. Data Retention

The evolution of the IT industry has also led to an evolution in the approaches and techniques used by attackers so that they can be able to achieve their goals as well. Therefore, Intrusion Detection Systems (IDSs) are constantly developed in response to the continuous development of new approaches of attacks by attackers.

IDSs uses IoT devices to creating datasets that will be used for evaluation of anomaly-based network intrusion detection systems. IoT data collection is the process of using sensors to track the conditions of user request. Devices and technology connected over the IoT can monitor and measure data in real time. The data are transmitted, stored, and can be retrieved at any time.

IoT systems are designed so data collection and related computation occur at the network edge in remote locations away from the data center. Relevant data is sent to a data center or the cloud over a network. The difference is IoT devices are numerous, which imposes large-scale requirements and the need to manage many sensors or devices.

The attack classification and IoT data understanding in the misuse attack detection model are based on machine learning (ML) algorithms. Data sets from real use cases are fed into the various machine learning to build new algorithms and improve existing algorithms.

3.4. CONCLUSION

The model trained in this chapter will then be converted into a misuse attack detection model which can now run based on a machine learning algorithm. The CAIDA- DDoS Attack 2007 dataset and three models of the machine learning will be the main inferencing component of the misuse attack detection interface for different users, the focus of this thesis being to enhance the accuracy of detection misuse attacks in NFV. This model will deliver a seamless and efficient misuse attack service for the CAIDA- DDoS Attack 2007 dataset on NFV. The user will be able to use this model to perform machine learning on their computers with a limited number of resources, such as a battery and processor for real-time misuse attack identification.

4. IMPLEMENTATION

4.1. IMPLEMENTATION OF INTERFACES

Misuse attack detection is a one-step solution for securing cloud computer in NFV environment. It has been implemented in a manner so that it can be deployed in all types of network architectures. The IoT sensors were connected to servers in cloud computer, then it provided IoT data, and stored in dataset.

The misuse detection model library has been added to the TansorFlow Library to Pychram as shown in the list below. Pychram is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers.

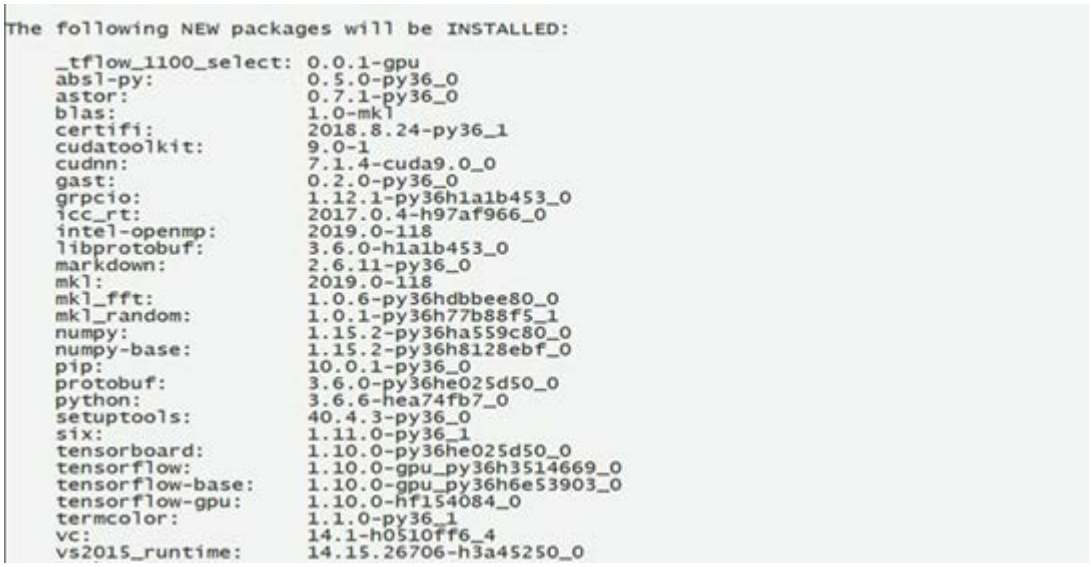

Figure 5 – Install TensorFlow Entry in the Pychram Library

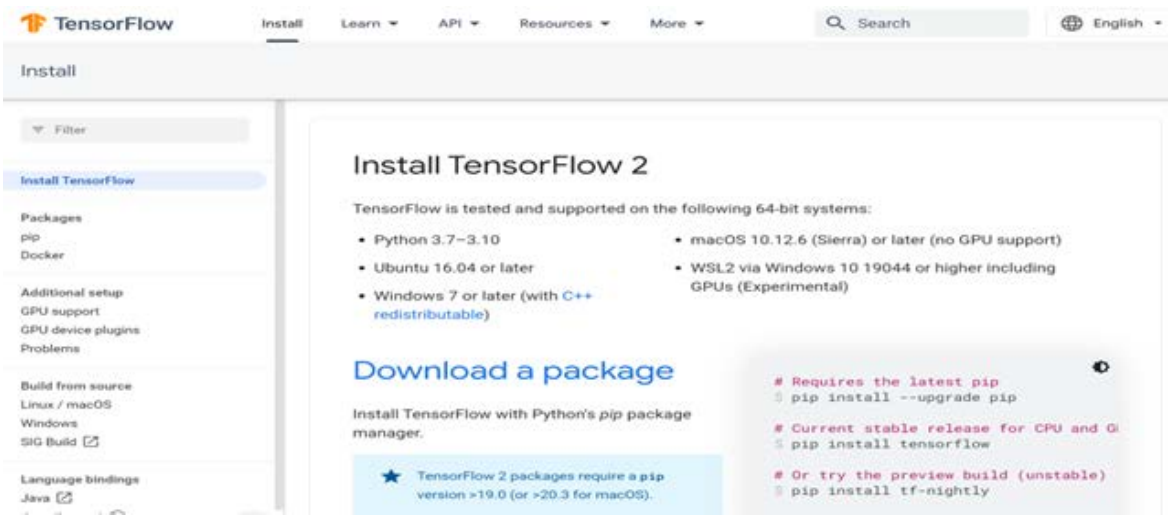

Figure 6 – Interface of the Install TensorFlow

The misuse detection model library has been added to the Pandas Library to Python as shown in the figure below.

```
bash-3.2$ conda install pandas
Collecting package metadata: done
Solving environment: done

## Package Plan ##

  environment location: /anaconda3

  added / updated specs:
    - pandas


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ca-certificates-2019.1.23  |                0         126 KB
    certifi-2018.11.29         |            py36_0         146 KB
    conda-4.6.4                |            py36_0         1.7 MB
    openssl-1.1.1a             |       h1de35cc_0         4.6 MB
    pandas-0.24.1              |   py36h0a44026_0        10.1 MB
    ------------------------------------------------------------
                                           Total:        16.6 MB
```

Figure 7 – Install Pandas in the Python Library

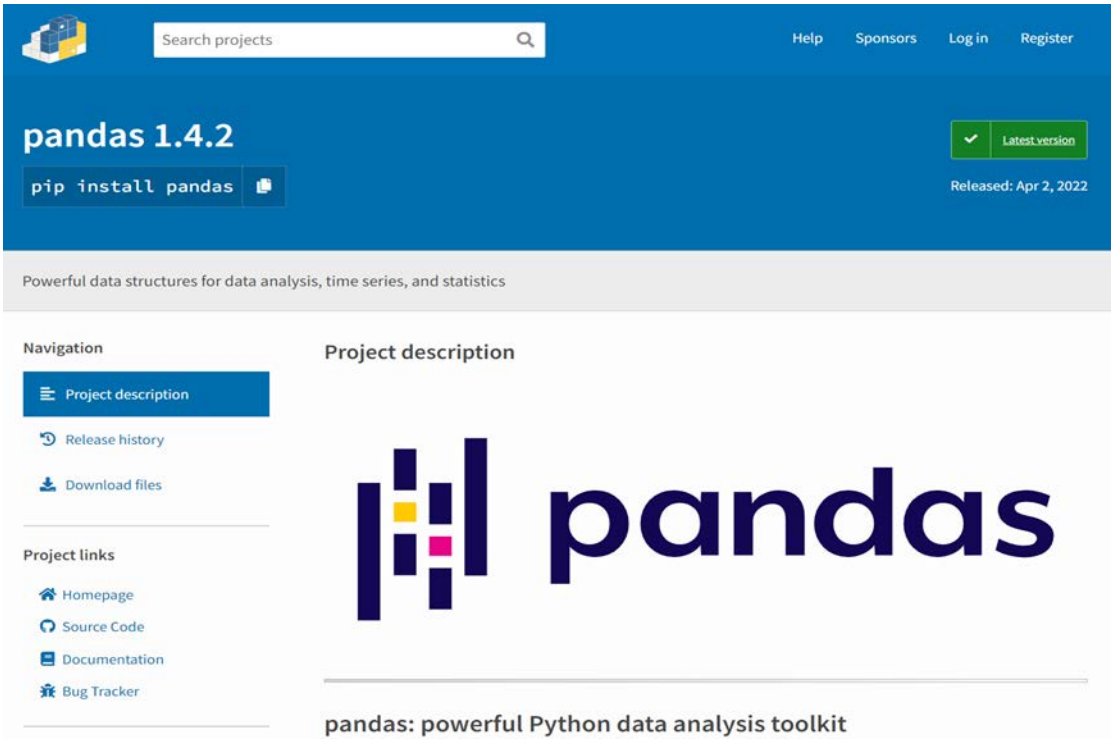

Figure 8 – Interface of install Pandas Library

The misuse detection model installs Numerical Python or shortly Numpy library to python language to deal with arrays as clarify in figures bellow.

```
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\acer>pip install numpy
Collecting numpy
  Downloading https://files.pythonhosted.org/packages/ce/ad/2e88f36b56f64f70c081b32fa5512dacedf12005ccb0c2d300d44dcc1215
/numpy-1.17.4-cp37-cp37m-win32.whl (10.7MB)
     |████████████████████████████████| 10.7MB 467kB/s
Installing collected packages: numpy
Successfully installed numpy-1.17.4

C:\Users\acer>_
```

Figure 9 – Install NumPy in the Python Library

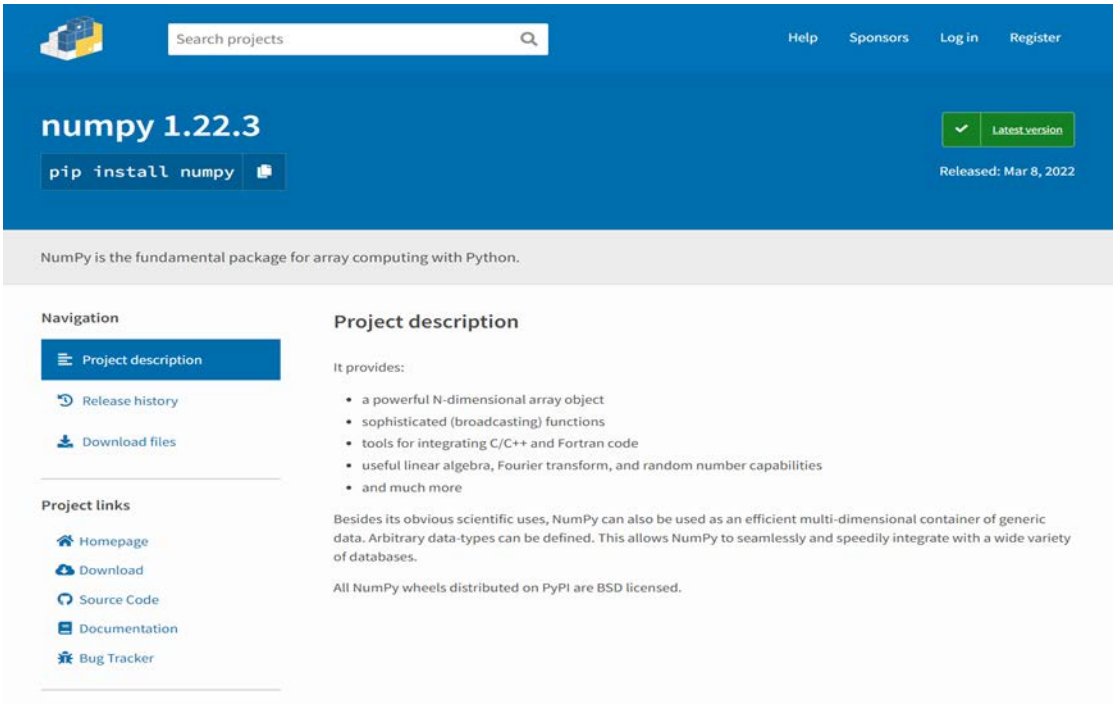

Figure 10 – Interface of install NumPy Library

In the implementation of the proposed misuse detection model, firstly running the training the input CAID-DDoS attack dataset using training interface as illustrated in figure 11.

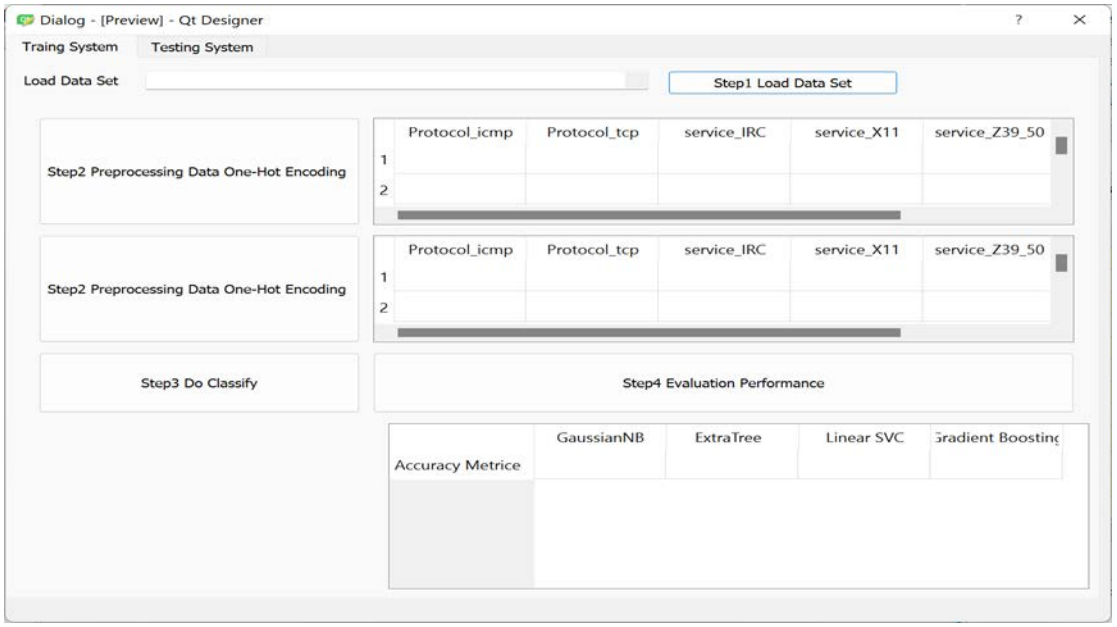

Figure 11 – Implementation of the Training Interface

After training dataset then testing the proposed model by input sample from the testing dataset to detection the input sample if it is normal attack or misuse attack as shown in figure 12.

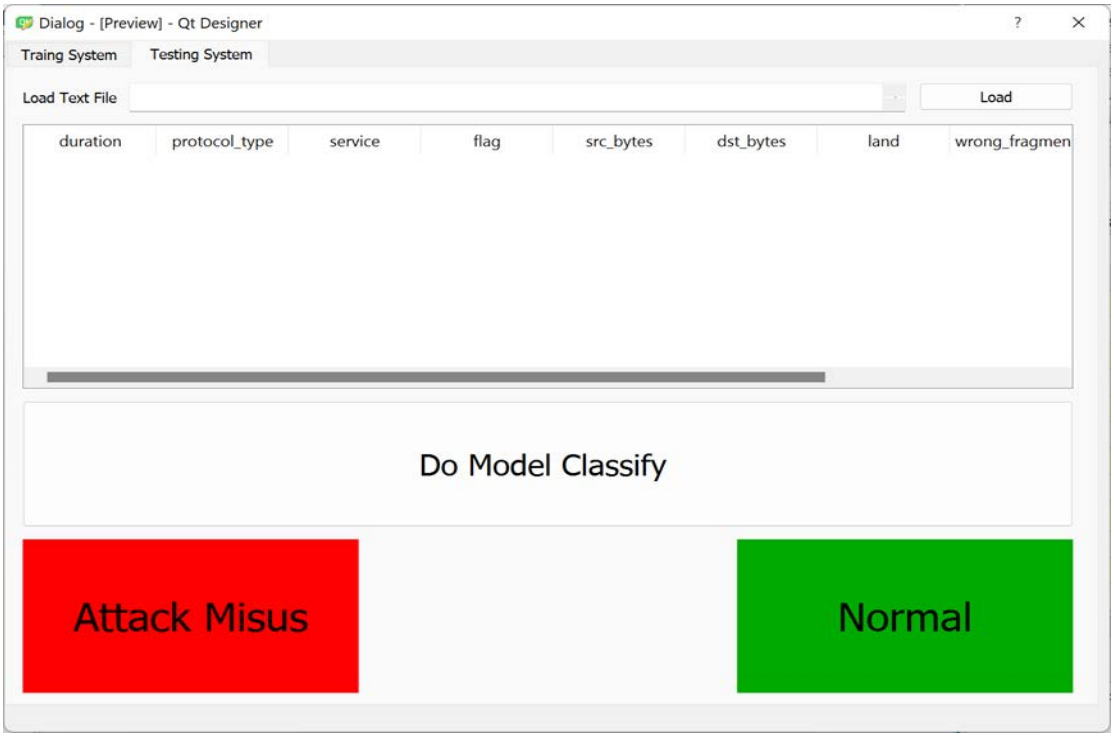

Figure 12 – Implementation of the Testing Interface

4.2. CONCLUSION

In this chapter, we explained how to add the libraries required by this project for use in the training and testing process, these libraries are TensorFlow, Pandas, and Numpy. The implementations of the interface for training and testing are presented in this chapter. These interfaced used to display the results and show it to admin, so that the admin can monitor the movement of requests arriving to the cloud if they are from an attacker, the proposed system will ignore the request or accept it.

5. TESTING

5.1. TESTING THE DETECTION MODEL

The methodology of this project involves applying three machine learning algorithms : random forest , C4.5 ,and Linear SVC onto stored data of misuse attack .The proposed model studies data behaviors by find the relationship between attribute and attack to understanding the correlation between them as shown in figures 13 ,14, and 15 will be studies data by analysis the relationship between each three attributes and attacks such as the relationship between 3 attributes(diff_srv_rate, same_srv_rate ,and srv_error_rate) with (misuse and normal attack) and so on.

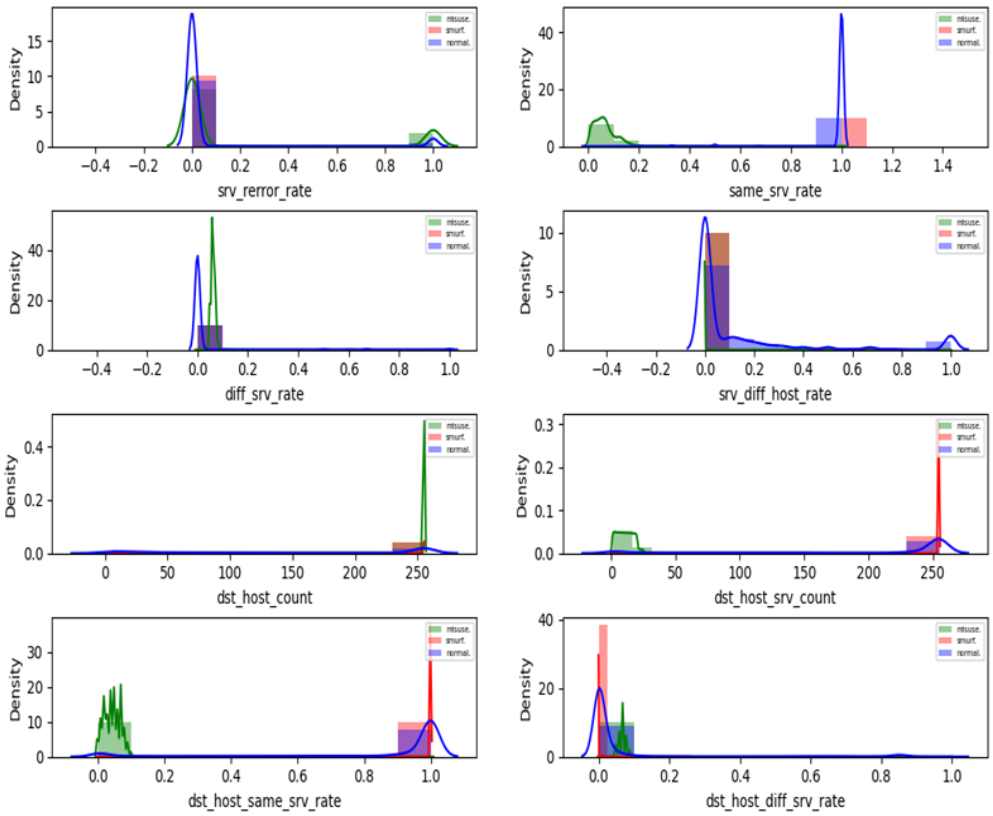

Figure 13 – One Dimensional Histogram of the Relationship Between Attributes (sev_error-rate, same_srv_rate,diff_rate, rv_diff_host_rate, dst_host_count, dst_host_ srv _count, dst_host_same_srv_rate, dst_host_diff_srv_rate) and misuse attack

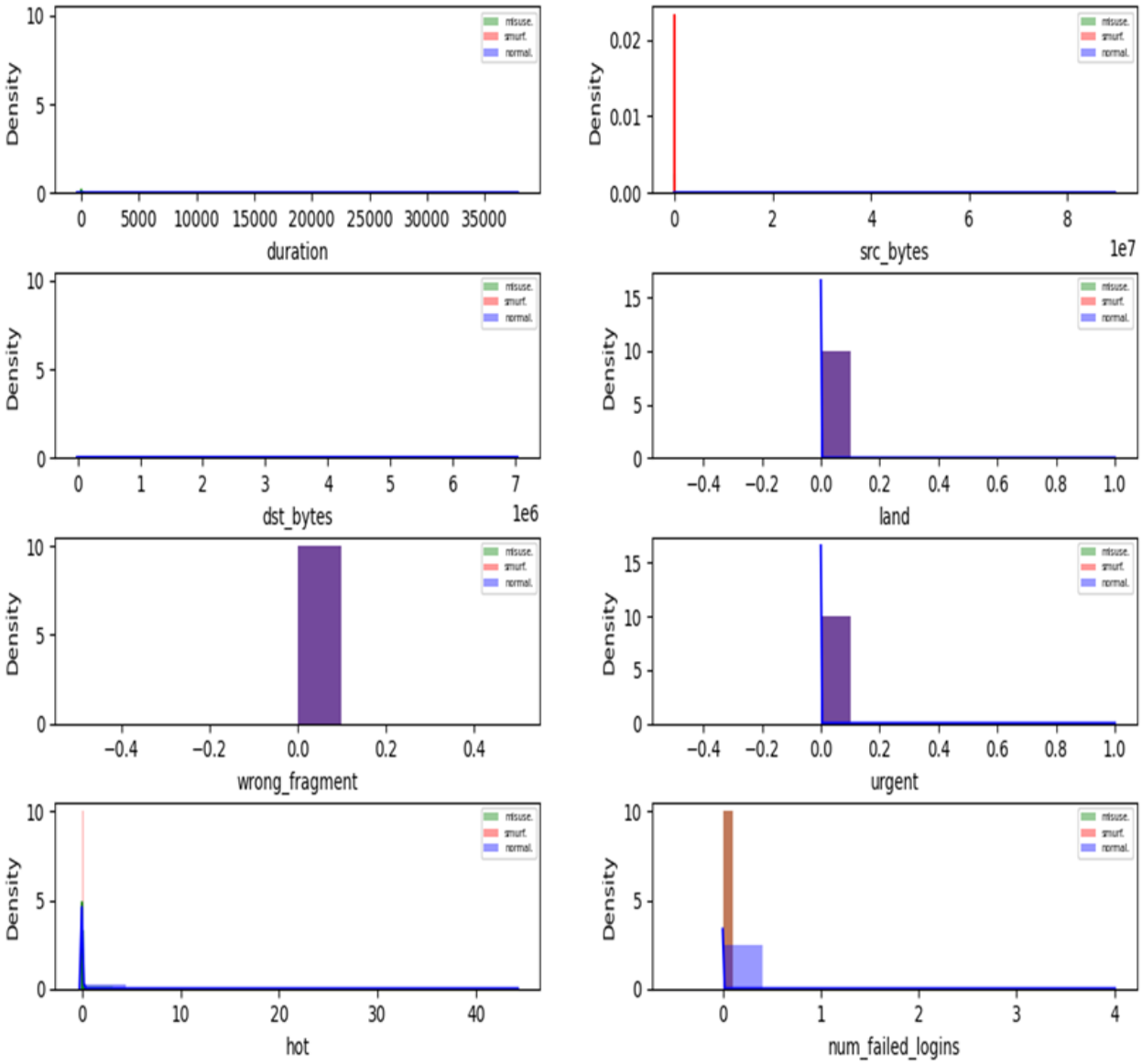

Figure 14 – One Dimensional Histogram of the Relationship Between Attributes (duration,scr_bytes,dst-byte,land,wrong_fragment, urgent,hot,num_failed_logins) and misuse attacks

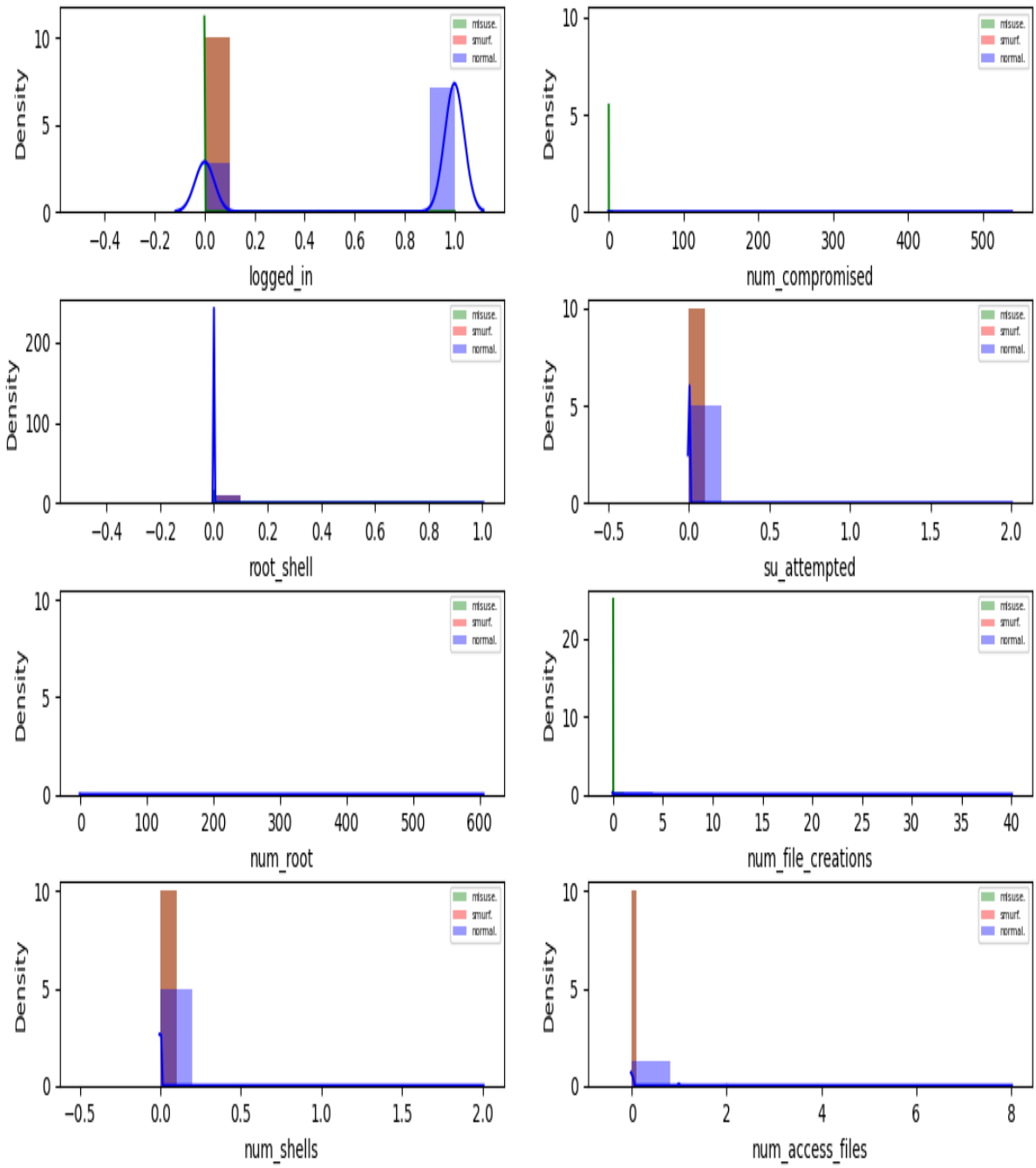

Figure 15 – One Dimensional Histogram of the Relationship Between Attributes (logged_in,num_compromised,root_shell,su_attepted,num_root,num_file_creations, num_hells, num_access _files) and Misuse Attack

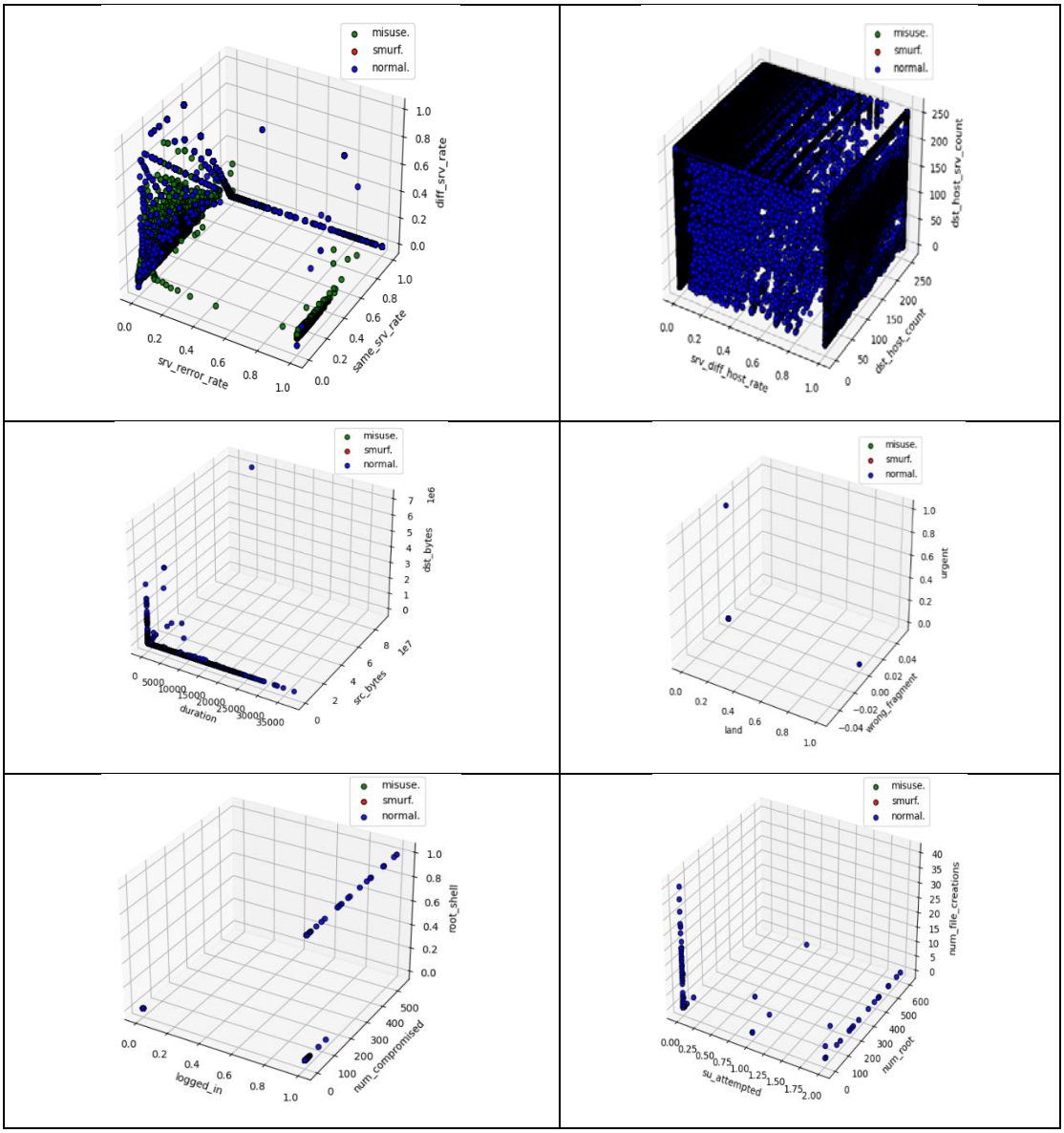

Figure 16 – Three Dimensional Histogram of the Relationship Between Attributes and Misuse Attack in the Dataset

The implementation classification algorithms are needs to splitting dataset into 70% training and 30% testing. The misuse detection model is classified by starting with preprocessing data and normalization data from the training step and applying algorithms (Random forest, C4.5 decision tree ,and Linear SVC) on the entered test samples. Where accuracy is used for performance evaluation.

The accuracy of the model was measured as the percentage of the right classification of the input sample if its belongs to normal or misuse attack.

A confusion matrix is a summary in table form of the classification results on a classification model. The number of right and wrong classifications are summarized and divided by class using count values.

For Misuse attack detection, the following were the results of testing the model: The Random Forest has a 99.90% accuracy, the C4.5 decision tree has a 99.87 % accuracy, and the Linear SVC has a 99.85 % accuracy. The confusion matrix for each machine algorithms are shown below:

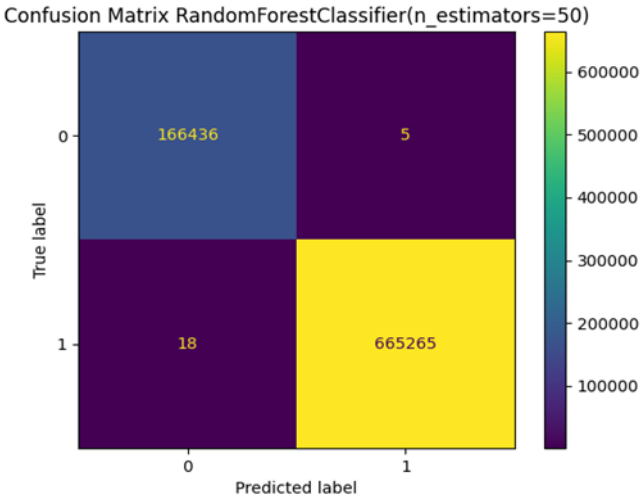

Figure 17 – Confusion Matrix of the Random Forest Algorithm

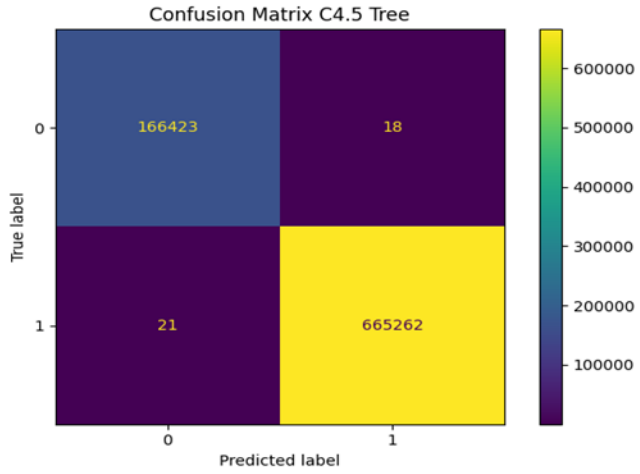

Figure 18 – Confusion Matrix of the C 4.5 Tree Algorithm

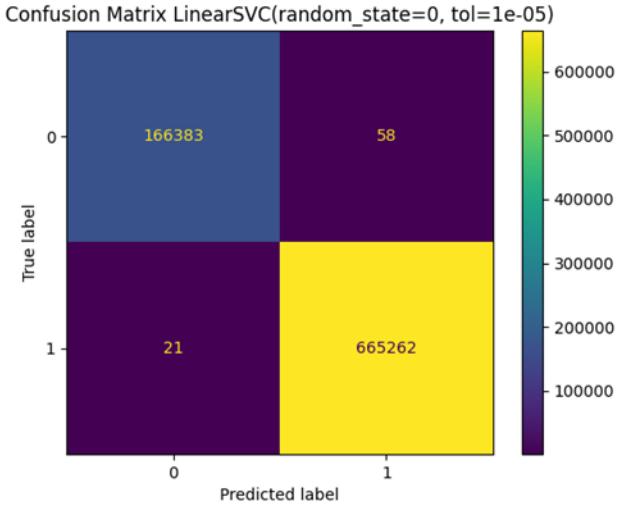

Figure 19 – Confusion Matrix of the Linear SVC Algorithm

As shown in confusion matrix overall machine learning algorithms have excellent performance to classification misuse attacks. Figure 20 shows a comparison between Random forest, C4.5 decision tree ,and Linear SVC algorithms based on accuracy values . In this figure illustrates the Random forest algorithm has better performance with accuracy=99.90% while the C4.5 tree algorithm has accuracy 99.87% and Linear SVC has accuracy 99.85%.

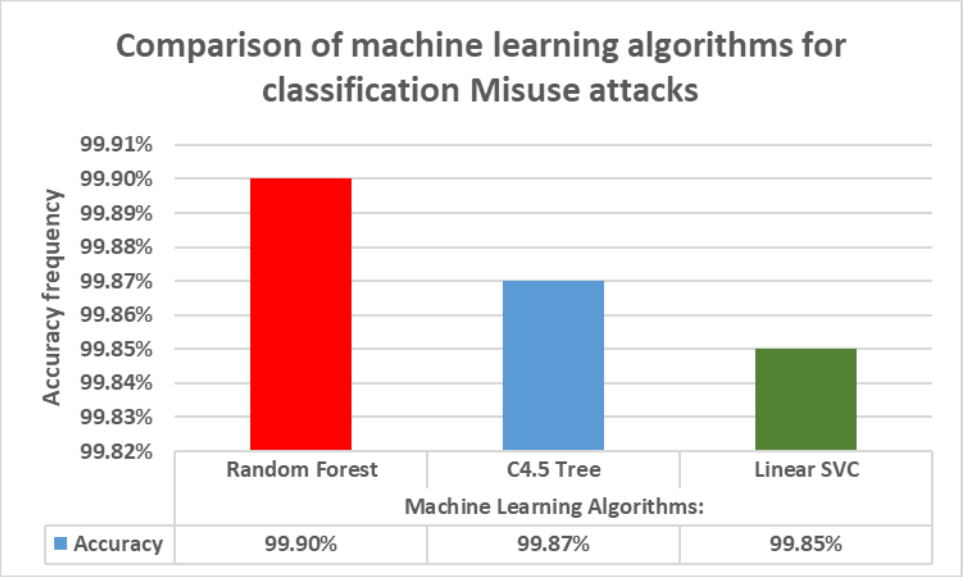

Figure 20 – Comparison of Machine Learning Algorithms for Classification Misuse Attacks

5.2. Conclusion

In this chapter we have conducted testing for the three machine learning models which returned a best accuracy values. The Random forest algorithm has better performance with accuracy 99.90% while the C4.5 tree algorithm has accuracy 99.87% and Linear SVC has accuracy 99.85%. The detection model is now ready for use by other developers.

6. CONCLUSION

This project undertakes a viable solution for the need for machine learning at the very basic level, that is, in low-powered limited resources devices. this makes this project able to keep the security of IoT data in NFV networks by identifying and diagnosing with high accuracy the most dangerous type of attack, which is the misuse attack in cloud services. It basically uses the CAIDA- DDoS Attack 2007 dataset, which is carefully chosen and ideal for robotic applications.

The tasks solved in this thesis include:

1. Deploy a library that provides Network Function Virtualization (NFV) capability to combat misuse attacks on IoT networks. This is by developing a machine learning model that can identify misuse attacks in the CAIDA- DDoS Attack 2007 dataset.
2. Development of a library that runs on devices that consume low power and have low processing capabilities by converting the machine learning model into a TensorFlow model that can run in very small devices.
3. Testing of the library has been done using accuracy metrics and an example of how to implement interfaces is added into the Library and can be accessed using the Python.
4. Employed libraries that apply data mining methods to provide secure IoT data by early and accurate identification of the misuse attack to avoid their consequences. This is used one-hot encoding for analysis dataset and three effective and three effective machine learning models for detection misuse attacks.
5. Test three classification models random forest, and C4.5 decision tree, and Linear SVC based on the accuracy measures and compare the results of these models

and choose the best model that got the higher accuracy in the detection of misuse attack.

6. Conduct experiments to examine the effectiveness of the machine learning models are employed in the proposed system and compare them based on accuracy value.

Due to the successful running of the misuse attack detection model, the next steps will be to constant updating of the feature database to cope with attacks from variant malware. Improvements as well will be done to tune the performance of the misuse attack detection model as shown below:

6.1. Improving Latency

Model architectures are complex and time-consuming to build, but there have lately been significant advancements in automating the process, such as data mining models that use deep learning techniques or machine learning to enhance network designs. These are still far from being able to completely replace humans.

I'm looking forward to using a ready machine learning models like random forest, C4.5 decision tree, and Linear SVC, and developing them using an effective data mining technique that allows users to avoid many of the gritty details of training, design the best possible model for our data, and solve latency issues through efficiency trade-offs.

6.2. Improving Power Usage

To do so, I will attempt to evaluation how much power the model utilized on diverse devices by calculating the latency for running one sample, and then multiplying the average power usage of the system for that period to obtain the energy usage. I can predict how long a model will take to execute if I know how many arithmetic operations it requires and how many operations per second a processor can accomplish

6.3. Improving Model and Binary Size

Weights are now kept as floating-point numbers in training, and each one takes up four bytes of memory. Because space is such a limitation for embedded devices, I'll utilize TensorFlow's compression utility to assist machine learning. TensorFlow is cross-platform, meaning it can operate on popular operating systems like Windows. TensorFlow Distributed Execution Engine can run training models on numerous computers at the same time, reducing training time. TensorFlow is a machine learning framework that may be used to understand the laws of abuse behavior and abnormalities. TensorFlow code is considerably easier to execute in a distributed way across a cluster of computers when utilizing GPUs since the execution mechanism is represented by graphs. Each of these codes is stored in a single byte, and arithmetic operations are performed on them.

REFERENCES

1. Guizani, N. A network function virtualization system for detecting malware in large IoT based networks / N. Ghafoor // IEEE Journal on Selected Areas in Communications. – 2020. – V.38. – P. 1218-1228.
2. Mazher, A.N. The Security Threats and Solutions of Network Functions Virtualization: A Review / A.N. Mazher, J. Waleed, A.T. MaoLood // Journal of Al-Qadisiyah for computer science and mathematics. – 2020. – V. 12. – P. 38.
3. Alhebaishi, N. Modeling and mitigating security threats in network functions virtualization (NFV). In IFIP Annual Conference on Data and Applications Security and Privacy / L .Wang, S. Jajodia. – Springer: Cham, 2020. – P. 3-23.
4. Liu, H. Machine learning and deep learning methods for intrusion detection systems: A survey / H. Lang // Applied sciences. – 2019. – V.9. – P. 4396.
5. Kocher, G. Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges / G. Kumar // Soft Computing-2021. – V. 25. – P. 9731-9763.
6. Park, Y. Distributed security network functions against botnet attacks in software-defined networks / Y. Park, N.V. Kengalahalli, S.Y. Chang // IEEE Conference on Network Function Virtualization and Software Defined Networks. – 2018. – P. 1–7.
7. Goel, R. Parallel Misuse and Anomaly Detection Model / R. Goel, A. Sardana, R. C. Joshi // Int. J. Netw. Secur. – 2012 – V. 14. – P. 211-222.
8. Papamartzivanos, D. Introducing deep learning self-adaptive misuse network intrusion detection systems / D. Papamartzivanos, F.G. Mármol, G. Kambourakis // IEEE Access. – 2019. – V. 7. – P. 13546-13560.

9. Ali, A. K. Detection of Misuse Attack in NFV Networks Using Machine Learning. In Journal of Physics / A.K. Ali, W.S. Bhaya // Conference Series. – 2021. – V. 1818. – P. 12123.
10. Adilova, L. System misuse detection via informed behavior clustering and modeling / L, Adilova, L. Natious, S. Chen, etc. // In 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops. – 2019. – P. 15-23.
11. The CAIDA “DDoS Attack 2007” Dataset. [Electronic Resource] https://www.caida.org/catalog/datasets/ddos-20070804_dataset/.
12. Alzahrani, S. Generation of ddos attack dataset for effective ids development and evaluation / S. Alzahrani, L. Hong // Journal of Information Security. – 2018. – V. 9. – P. 225-241.
13. Nguyen, G. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey / G. Nguyen, S. Dlugolinsky, M. Bobák, etc. // Artificial Intelligence Review. – 2019. – V. 52. – P. 77-124.
14. Abadi, M. {TensorFlow}: A System for {Large-Scale} Machine Learning / M. Abadi, P. Barham, J. Chen, etc. // USENIX symposium on operating systems design and implementation. – 2016. – P. 265-283.
15. McKinney, W. Pandas: a foundational Python library for data analysis and statistics / W. McKinney // Python for high performance and scientific computing. – 2011. – V. 14. – P. 1-9.
16. Harris, C. R. Array programming with NumPy / C.R. Harris, K.J. Millman, S.J. Van Der Walt, etc. // Nature. – 2020. – V.585. –P. 357-362.
17. Scarfone, K. Guide to intrusion detection and prevention systems (idps) / K. Scarfone, P. Mell // NIST special publication. – 2007. – V. 800. – P. 94.

18. Modi, C. A survey of intrusion detection techniques in cloud / C. Modi, D. Patel, B. Borisaniya, etc. // Journal of network and computer applications. – 2013. – V. 36. – P. 42-57.
19. Potdar, K. A comparative study of categorical variable encoding techniques for neural network classifiers / K. Potdar, T.S. Pardawala, C.D. Pai, etc. // International journal of computer applications. – 2017. – V. 175. – P.7-9.
20. Ali, P. J. M. Data normalization and standardization: a technical report / P.J.M. Ali, R.H. Faraj, E. Koya, etc. // Mach Learn Tech Rep. – 2014. – V. 1. – P. 1-6.
21. Negandhi, P. Intrusion detection system using random forest on the NSL-KDD dataset / P. Negandhi, Y. Trivedi, R. Mangrulkar // In Emerging Research in Computing, Information, Communication and Applications. – Springer: Singapore. – 2019. – P. 519 –531.
22. Resende, P. A. A. ACM Computing Surveys (CSUR) / P.A. Resende, A.C. Drummond // A survey of random forest based methods for intrusion detection systems. – 2018. – V. 51. – P. 1-36.
23. Koshal, J. International Journal of Computer Network and Information Security / J. Koshal, M. Bag // Cascading of C4. 5 decision tree and support vector machine for rule based intrusion detection system. – 2012. – V. 4. – P. 8.
24. Rai, K. Decision tree based algorithm for intrusion detection / K. Rai, M.S. Devi, A. Guleria // International Journal of Advanced Networking and Applications. – 2016. – V.7. – P. 2828.