COLLEGE OF COMPUTING AND INFORMATICS

UNIVERSITI TENAGA NASIONAL

NETWORK INTRUSION DETECTION MODEL BASED ON REAL AND ENCRYPTED SYNTHETIC ATTACK TRAFFIC USING DECISION TREE ALGORITHM

MIKHAIL AMZAR BIN KAMARUDDIN

2022

NETWORK INTRUSION DETECTION MODEL BASED ON REAL AND ENCRYPTED SYNTHETIC ATTACK TRAFFIC USING DECISION TREE ALGORITHM

by

MIKHAIL AMZAR BIN KAMARUDDIN

Project Supervisor: Nur Shakirah Binti Md Salleh, TS

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF COMPUTER SCIENCE,
COLLEGE OF COMPUTING AND INFORMATICS
UNIVERSITI TENAGA NASIONAL

2022

DECLARATION

I hereby declare that this report, submitted to University Tenaga Nasional as a partial fulfillment of the requirements for the Bachelor of Computer Science has not been submitted as an exercise for a degree at any other university. I also certify that the work described here is entirely my own except for excerpts and summaries whose sources are appropriately cited in the references.

This report may be made available within the university library and may be photocopied or loaned to other libraries for the purposes of consultation.

31 December 2022

MIKHAIL AMZAR BIN
KAMARUDDIN
CS0107477

APPROVAL SHEET

This thesis entitled:

“Network Intrusion Detection Model Based On Real and Encrypted Synthetic Attack Traffic Using Decision Algorithm”

Submitted by:

MIKHAIL AMZAR BIN KAMARUDDIN (CS0107477)

In requirement for the degree of Bachelor of Computer Science, College Of Computing and Informatics, University Tenaga Nasional has been accepted.

Supervisor: Nur Shakirah binti Md Salleh, TS

Signature: …………………………….

Date: 31st December 2022

ABSTRACT

Network intrusion detection systems (NIDS) are essential for defending networks from online dangers. In this research, we suggest a real-time intrusion detection system (NIDS) based on a decision tree algorithm. Our method makes use of a sizable dataset of network traffic with labels specifying whether each instance is a normal traffic or malicious traffic. On this dataset, the decision tree algorithm is trained to identify the patterns and traits of both typical and abnormal behavior. For this paper, focus is given on understanding related topics like machine learning and intrusion detection system by reading several resources such as research journals, online articles, books, and so on. Additionally, exploring numerous possible tools for use in the project to develop the network intrusion detection system. Overall, this study shows the potential of utilizing a decision tree algorithm for NIDS and emphasizes the significance of taking both accuracy and efficiency into account when designing such systems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1 INTRODUCTION

1.1 Background

Currently, we have become so connected to the digital world that the convenience of information is at the tip of our fingers. With the world now being so familiar with the internet, it is logical to look at the security aspects of it as we want to keep enjoying these privileges without being harmed or exploited. One of it being network security. Network security is important as it covers on how to secure our networks which would lead to the security of our digital assets and information. Network intrusion is a general term to describe an unauthorized breach or exploit that affect one’s network. Naturally, we want to avoid this attack from happening to our network which is why technology like network intrusion detection system is developed. Network intrusion detection system will monitor our network traffic and analyze it, informing the person in charge of the network if any malicious activity or anomaly in the traffic is detected. To develop such system, advanced knowledge of network and algorithms are needed. Machine learning algorithms can be useful in this case because we can develop and train the network intrusion detection system to learn from the datasets we provide initially, so that it can decide what to do when receiving real network traffic data. The system will be able to identify feature of a malicious traffic based off its training and will alert network administrators of an intrusion

1.2 Problem Statement

One of the primary concerns surrounding network is security. With the world now being so connected to the internet, naturally there will be increased total of malicious activity committed on networks. One of these malicious activities is a network intrusion. The act of gaining illegal access to a computer network or system, as well as making an unsuccessful attempt to do so, is known as network intrusion. It can be carried out by a single person or by a group, and it can be driven by a wide range of purposes, including the theft of sensitive data, the disruption of the operation of the network, or the gaining of illegal access to resources. It is essential to employ security measures such as firewalls, antivirus software, and frequent security upgrades in order to prevent unauthorized access to a computer network. It is also essential to educate users about the dangers posed by network intrusions, as well as the ways in which such attacks can be recognized and avoided Security systems such as a network intrusion detection system have been developed for many years now. Network intrusion detection system monitors and analyzes network traffic and alerts the responsible party if any anomaly or suspicious activity are detected. The goal for this project is to develop a network intrusion detection model that will help us achieve such results using datasets of real and encrypted synthetic attack traffic while utilizing a decision tree algorithm and will provide user a dashboard containing relevant results concerning data of the detection.

1.3 Objective

The project objectives are:

i. To identify the features of network intrusion based on the existing dataset.

ii. To design a network intrusion detection model that applies Decision Tree algorithm.

iii. To produce a machine learning model and dashboard for network intrusion detection model.

1.4 Scope

1.4.1 User Scope

User of the system would be network administrators.

1.4.2 System Scope

The scope of the system consists of implementing decision tree machine learning algorithm. Additionally, the system will only perform intrusion detection. Dataset that will be used for the training of the algorithm is a publicly available dataset – ALLFLOWMETER_HIKARI2021.csv.

1.5 Expected Outcomes

i. A machine learning model and dashboard must be produced for the network intrusion system.

ii. The network intrusion detection model should apply Decision Tree algorithm.

iii. Using the available datasets, features of a network intrusion should be able to be identified.

CHAPTER 2

LITERATURE REVIEW

2.1 Network Intrusion System

A Network Intrusion Detection System, or NIDS, is a form of security software that monitors a computer network in order to identify and report instances of unauthorized access or malicious behavior to the administrators of the network. It is able to inform administrators when it identifies traffic on the network that could be potentially dangerous because it can monitor network traffic for suspicious behavior or recognize malicious attack pattern. Network Intrusion Detection System is capable of being configured to monitor traffic on a particular network segment or over the entirety of the network. In order to identify potentially malicious behavior, it employs a number of different methods, including signature-based detection and anomaly-based detection. In signature-based detection, established harmful traffic patterns are looked for. This works well for identifying known attacks, but it might miss new or unusual attack types. On the other hand, Anomaly-based detection refers to detecting anomalies based on deviations from typical network behavior. This can be useful for spotting novel attack vectors, but it may also identify legitimate traffic as suspicious, leading to a lot of false positives.

Table 2.1: Intrusion Detection System by implementation

Type	Implementation
Network-based IDS (NIDS)	Placed in strategic points within the network, typically data chokepoints.
Host-based IDS (HIDS)	Runs on the host system it is placed in.

Table 2.2: Intrusion Detection System by detection method

Detection model	Description
Signature-based IDS (SIDS)	Examines network packets and compares them to a database of known attack signatures or features. This type of IDS searches for specified patterns, such as byte or instruction sequences.
Anomaly-based IDS (AIDS)	Detects current network traffic and compares trends to a baseline. It detects malicious activity patterns rather than specific data patterns, going beyond the attack signature approach.

2.2 Machine Learning

The development of algorithms and statistical models that enable computers to learn from data and make judgements based on it without being explicitly programmed is known as machine learning, and it is a subfield of artificial intelligence. In the field of machine learning, an algorithm is taught to make predictions or choices by being exposed to a big dataset during its training phase. For example, a machine learning algorithm might be trained on a collection of consumer data for an online shopping platform, including information about their search history, purchases, and cart contents. The algorithm might therefore be used to anticipate which items the customer might buy next. Machine learning comes in a variety of kinds, including reinforcement learning, unsupervised learning, semi-supervised learning, and supervised learning.

Table 2.3: Types of machine learning methods

Machine Learning Type	Description
Supervised learning	A training dataset is needed for supervised learning, which must include labelled responses or output targets as well as examples for the input. The ML model are then calibrated using the pairs of input and output data from the training set. Following a model's successful training, it can be used to forecast the target output using fresh or previously unobserved data points of the input attributes.

Unsupervised learning	Learning system is expected to identify patterns without any labels or guidelines in place, this is known as unsupervised learning. Training data only consists of variables x with the goal of finding structural information of interest. For example, grouping data of similar attributes or also known as clustering.
Reinforcement learning	We define the system's current state, establish a goal, provide a list of permissible actions and the environmental constraints on their results, and then let the ML model experiment with the process of reaching the goal on its own using the concept of trial and error to maximize a reward.

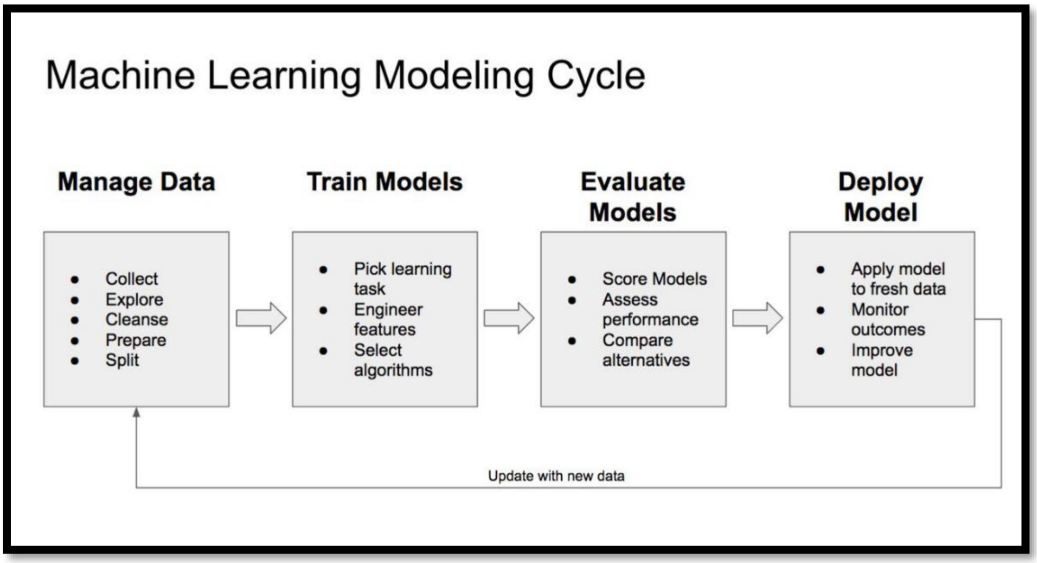


Figure 2.1: Machine Learning Modelling Cycle

2.2.1 Managing Data

The overall machine learning modelling process is a lengthy set of steps that will determine the overall success of the machine learning project. The data that will be used as input for the machine learning algorithm must be prioritized first. Data can be obtained and collected from a variety of sources, including purchasing from vendors, generating synthetic data, open-source datasets, and so on. The datasets should be collected in accordance with the needs of the machine learning project, so careful consideration and research are advised when deciding the data to use. After that, exploring the data which is about analyzing the dataset, identifying any error and values that needs to be labelled and corrected. Following collection and exploration, the dataset must be cleaned because datasets frequently have missing values, labelling errors, and so on. Uncleaned datasets may have an impact on the performance and results of the machine learning model, resulting in a poor machine learning outcome. Cleaning data can be time-consuming, but it is necessary to ensure that the dataset is valid, accurate, complete, consistent, and uniform. Furthermore, a high-quality dataset will greatly improve and accelerate the training process. Following that, feature selection and feature engineering techniques are required to facilitate the process of organizing relevant data for the machine learning model. There are numerous tools available to help speed up and automate the data preparation process. The datasets must then be divided into groups, with each group serving a different purpose. A dataset should essentially be divided into a training set and a test set. This is to evaluate the machine learning model's performance when making predictions based on the dataset. Train dataset is used to train the machine learning model, whereas test dataset is used to evaluate the trained machine learning model.

2.2.2 Model Training

Training the machine learning model consist of several important procedures that needs to be executed correctly in order to satisfy the requirements and objective of the machine learning model. But first, it is important to choose the type of machine learning model that is suitable for the intended task of the project. Starting with the type of learning task, a machine learning task is a form of prediction or inference made based on the problem or query and the available data. The classification task, for example, allocates data to categories, whereas the clustering task groups data based on similarity.

2.2.3 Model Evaluation

Evaluating the machine learning model is essential after training the model to determine if the performance is meeting expectation or not. Additionally, evaluation is important because we need to know whether the machine learning model can make accurate predictions. This phase involves providing the test dataset to the machine learning model. There are several metrics to consider when evaluating models such as classification metrics and regression metrics.

2.2.4 Model Deployment

This step is where the deployment of the finished machine learning model takes place. The process of putting a fully functional machine learning model into production so that it can make data-driven predictions. These predictions are then used by users, developers, and systems to make real business decisions.

However, these models are still being monitored to see if any changes occur in the models. A machine learning model might degrade over time for a variety of reasons. Therefore, machine learning model can undergo improvement by training it with new data.

2.2.5 Decision Tree Algorithm

In the field of machine learning, one type of algorithm that is used for classification and regression tasks is called a decision tree. It is a tree-like structure similar to a flowchart that displays a collection of decisions and the probable outcomes of those actions. The "decisions that need to be made" are represented by the "nodes" in the tree, and the "potential outcomes" of those decisions are represented by the "edges" in the tree. Decision trees are a method for modelling decisions and outcomes, which map decisions using a branching structure. Decision trees are utilized to determine the likelihood of success for various sequence of decisions made to attain a given objective. The concept of a decision tree predates machine learning, as it may be used to manually model operational decisions in the manner of a flowchart. As a technique for analyzing organizational decision making, they are extensively taught and utilized in business, economics, and operations management. In the context of machine learning, decision trees are a type of supervised learning, this means it train models with tagged input and output datasets. The approach is mostly used to handle classification problems, which involve categorizing or classifying an object using a model. Decision trees are also utilized in regression problems, a technique used in predictive analytics to forecast outputs from unknown input.

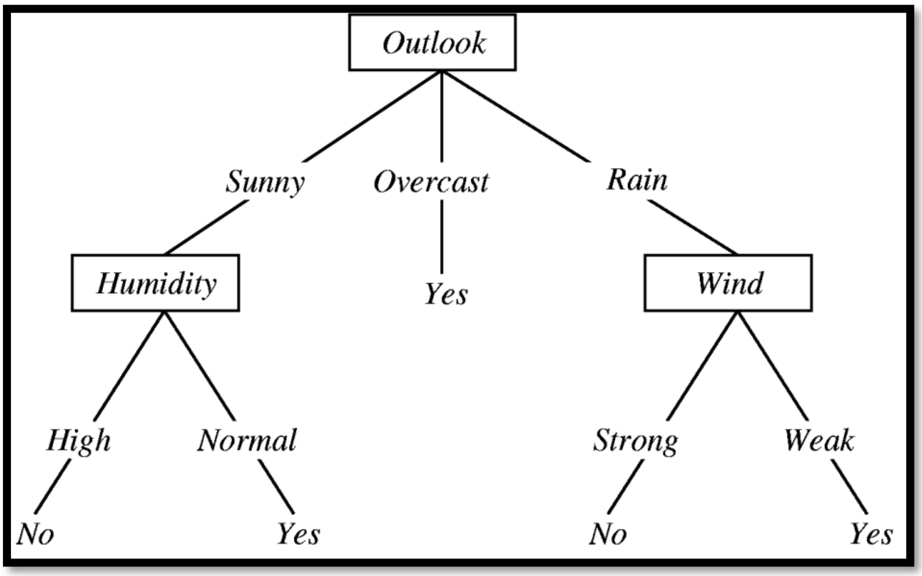


Figure 2.2: Decision tree example

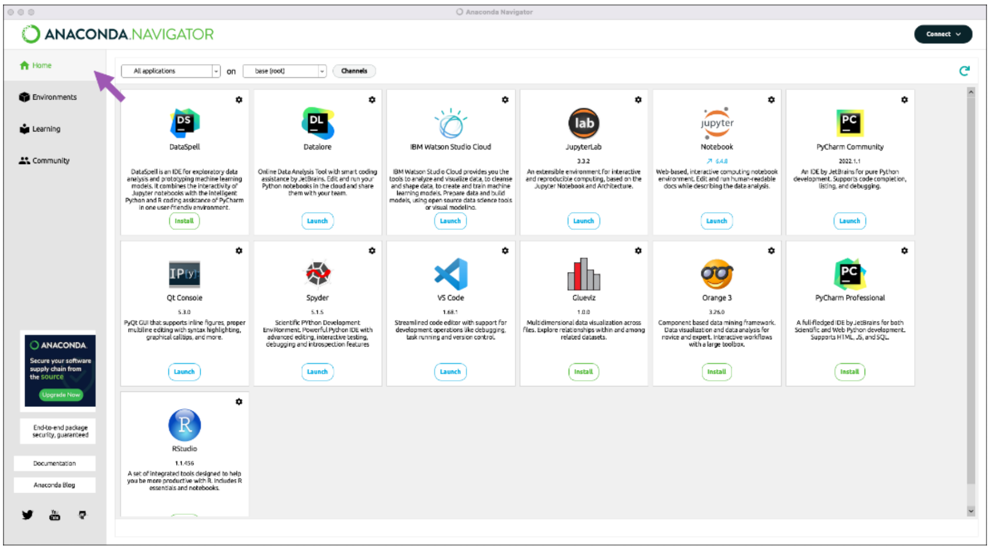
The most typical application of decision trees in machine learning is for classification problems. It is a supervised machine learning issue in which the model is taught to determine whether or not the data belongs to a known object class. Models are trained to label processed data with class labels. In the training phase of the machine learning model lifecycle, the model processes labelled training data to learn the classes. A model needs to comprehend the characteristics that classify a datapoint into the various class labels in order to solve a classification problem. A classification issue might actually arise in a variety of contexts. Document classification, image recognition software, and email spam detection are a few examples.

2.2.6 Tools

Anaconda Navigator

Anaconda Navigator is a graphical user interface (GUI) that allows users to launch applications and manage packages in the Anaconda Python distribution.

It is included with the Anaconda distribution and is designed to make it easier for users to work with Python packages and environments. The GUI can be used to launch Python programs, and operate the conda packages, environments, and channels without using the command line interface commands. Anaconda Navigator is included in the installation of Anaconda Distribution. Anaconda Distribution consist of a wide collection of Python libraries and conda, which is a tool for managing packages. Apart from the numerous open-source packages already available in Anaconda, user can also install thousands of packages from the Anaconda repository and community hub. Anaconda Distribution essentially simplifies the management of Python versions on a single computer and includes a wide number of highly optimized, regularly used data science modules to help users get started quickly. This makes it easier to operate and fulfil scientific packages requirements of Python versions. This software is available to be installed for macOS 10.10 and newer, Windows 10 x86_64 and newer, and Linux with glibc 2.17 or newer.



DataSpell
DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.
Install

Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.
Launch

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
Launch

JupyterLab
3.3.2
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch

Notebook
↗ 6.4.8
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

PyCharm Community
2022.1.1
An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.
Launch

Qt Console
5.3.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

Spyder
5.1.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

VS Code
1.68.1
Streamlined code editor with support for development operations like debugging, task running and version control.
Launch

Glueviz
1.0.0
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

Orange 3
3.26.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

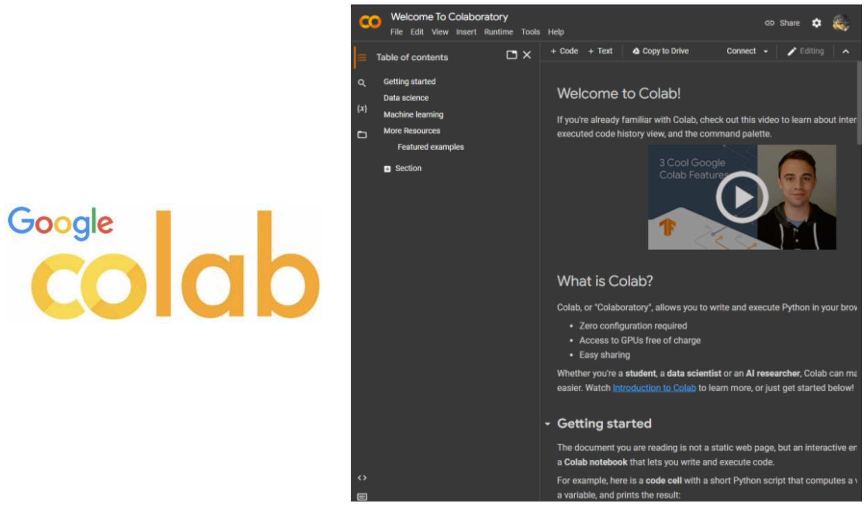
PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.
Install

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Figure 2.3: Anaconda interface and logo.

Google Colab

Google Colab is a product of Google Research. It enables anyone to create and execute arbitrary Python code through the browser. Codes can be executed inside a virtual machine dedicated and private to the user’s account. A great advantage of Colab is that it offers free access to computing resources while requiring no setup to use. Colab can be a great tool for the development of machine learning and data science related projects. Google Colab essentially help users write, test, and execute codes in a practical interactive document called notebooks that features cells which allow the creation of space that user can use to insert codes, texts, and other documentative elements. However, it is important to keep in mind that Colab does not provide unlimited computing resources which means it has the possibility to fluctuate due to usage limit. Computing resources are limited due to various reasons. Colab aim to have flexibility in adjusting usage limited and hardware availability because these are the free version of their service. But if the need to access better and guaranteed resources arises, there is Colab Pro, the paid version for Google Colab, available for subscription by users that are packed with many practical and useful upgrades and advantages over the free version. Colab Pro promises powerful GPUS, larger memory capacity, and 100 compute units. Going further up the tier is Colab Pro+ which offer even better resources and benefits.



Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about inter executed code history view, and the command palette.

What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your brov

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can ma easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive er a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a v a variable, and prints the result:

Figure 2.4: Google Colab logo and interface.

Weka

Weka is a machine learning software that was developed at the University of Waikato in New Zealand. It is available for free and is open source. It offers a collection of algorithms that can be used for various data mining tasks, such as classification, regression, clustering, and learning to associate rules with categories. Weka also features a graphical user interface that gives users the ability to load data, preprocess that data, and apply machine learning algorithms to that data without having to write any code. In general, Weka is a helpful tool for anyone who is interested in machine learning and data mining. It is particularly well-suited for researching and experimenting with a variety of various algorithms and approaches.

2.3 Dashboard

Analytical dashboard provides insights and help display relevant data to be understood easier by humans. By providing an overview of relevant and impactful data to the user. Through the data displayed on a dashboard, user can recognize trends, statistics, and events to make better decisions and analysis. Dashboards are useful when large and broad complex categorized information requires visualization to execute an accurate analysis of created data.

2.3.1 Grafana

Grafana is a free and open-source data visualization and monitoring tool that enables the construction of dashboards to monitor multiple metrics and data sources. It is typically used for visualizing time series data for monitoring infrastructure and application performance, but it may also be used to visualize data from databases and log files. InfluxDB, Prometheus, and Graphite are among the wide range of data sources that Grafana offer support to. It is equipped with great variety of visualization options, alongside support for graphs, tables, heat maps, and more. Grafana's ability to generate dynamic and configurable dashboards is one of its primary advantages. Dashboards can be created by dragging and dropping panels onto a layout and setting the panel to display the data. It is possible to alter the data presented in each panel using Grafana's query editor. Additionally, Grafana has alerting and notification functionalities in addition to visualization and dashboard capabilities, which help to set up alerts based on thresholds or other situations. When an alert is triggered, Grafana can be configured to send notifications via email, Slack, or other messaging platforms.

2.3.2 Google Looker Data Studio

Google Data Studio is a data visualization tool developed by Google that enables users to generate interactive dashboards and reports from a variety of data sources. The application may be accessed through the Google Cloud Platform.

Its purpose is to assist users in transforming data into insights that may be put into action and in communicating those findings to others. Anyone who is interested in understanding the trends and patterns present in their data might benefit from using Google Data Studio, which is a powerful tool for visualizing and analyzing data.

2.3.3 .NET

Microsoft created the software development platform known as .NET. for Windows, Linux, and macOS. It offers a foundation for creating, distributing, and operating applications and services. The.NET Standard Library, a collection of libraries that make up .NET, offers numerous features, including support for networking, data access, and file input/output. The Common Language Runtime (CLR), a runtime environment that executes.NET code and controls the resources of the computer on which the code is running, is also included. C#, F#, and VB.NET are just a few of the programming languages that are supported by .NET, giving developers the option to use the one that best suits their needs. It also offers frameworks for creating web, mobile, desktop, and cloud-based apps as well as tools for designing, debugging, and deploying applications. In general, .NET is a strong platform that makes it easier to create and deploy apps and services, and it is utilized by many developers worldwide.

2.4 Software Development Methodology

Software development methodology is the steps of process involved when developing software and the philosophy involved in the completion of the project. Methodology is important when executing projects as it lays out the path, timeline, objectives that help us stay on track during development process.

2.4.1 Waterfall Model

This model is linear and sequential. Each phase must be completed before proceeding with the next phase to ensure organized workflow and no overlapping occur. Consider the image below to understand the Waterfall Model:

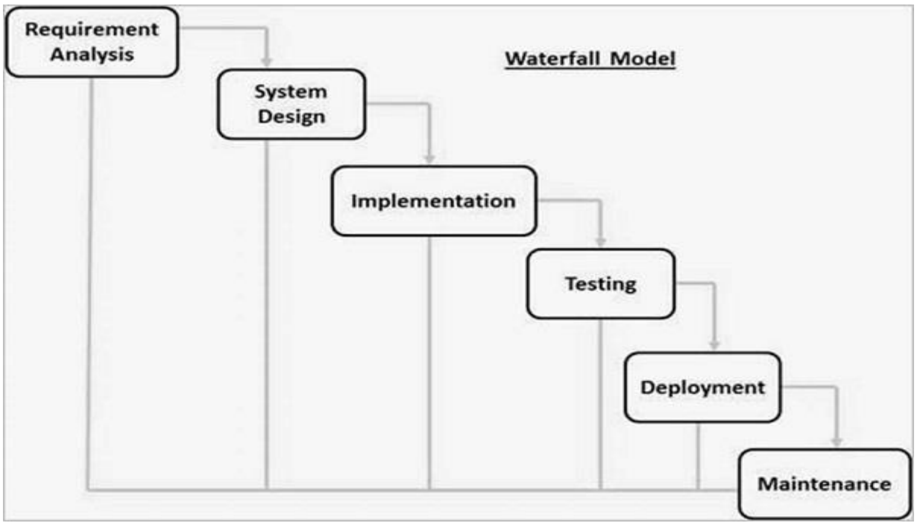


Figure 2.5: Waterfall model diagram

i. **Requirement Gathering and analysis**

During this phase, all potential system requirements are identified and recorded in a specification document.

ii. **System design**

This phase studies the specifications from the previous phase hence initiating system design. This system design aids in determining hardware and system requirements, as well as the overall system architecture.

iii. **Implementation**

The system is first built-in discrete programs called units, with input from the system design phase, and then combined in the following phase. Unit undergo unit testing to test for functioning after it is developed.

iv. **Integration and testing**

Following unit testing, all units generated during the implementation phase are integrated into a system. Following integration, the complete system is tested for flaws and failures.

v. **Deployment**

After the completion of functional and non-functional testing, the product is deployed in the client environment or released to the market.

vi. **Maintenance**

There may be problems or bugs that arise in the client environment. Patches are published to address these vulnerabilities. To improve the product, newer versions are published. Maintenance is performed to implement these modifications in the released software.

Waterfall models are suitable to be applied to projects where the requirements are determined, product definition is stable, expected timeline is relatively short, and the resources and expertise are available.

2.4.2 Agile

Software development using the agile methodology is flexible, iterative, and places a focus on quick prototyping, frequent releases, and ongoing improvement. It is predicated on the Agile Manifesto, a set of guidelines that emphasises customer happiness, functional software, and teamwork and communication. Agile approaches place a heavy emphasis on teamwork and communication among members and are designed to be flexible and responsive to shifting requirements and priorities. Teams are able to quickly react to new knowledge and changes in the business environment as a result, and they are able to provide value to customers fast and effectively. To aid teams in planning, monitoring, and managing their work, agile procedures are frequently used in conjunction with agile project management tools and techniques, such as burndown charts and agile boards. Scrum, Lean, and Extreme Programming are a few of the various agile approaches that have been created over time (XP). Although each of these methodologies has unique procedures and methods, they all follow the guidelines of the Agile Manifesto.

2.5 Similar System

In terms of concept and functionality, there are systems similar to this project such as Splunk and FireEye.

2.5.1 Splunk

Splunk is a software platform that searches, analyses, and visualizes machine-generated data collected from IT infrastructure and business's websites, applications, sensors, and devices. Analyzing these data allow organizations to come up with solutions related to certain aspects of their business. It also provides better understanding of the customer and the service they provide. The data provide an opportunity to be aware of issues regarding their system as well as improving the quality of service or functionality of systems. Machine data are typically complex because it may be presented in an unstructured format which means analyzing it becomes a challenge. Splunk works in a way that it essentially extracts machine data for users in a readable form. With these data user can monitor and record system performance to search for any failure conditions, investigate any occurring events, checking the business matrix, visualizing data to display results and dashboards, and storing data for future reference. Additionally, the benefits of Splunk are why many organizations use this software. Searching for data in Splunk is easy because of the Search Processing Language and the ability to input search data in any form. Splunk also does not require a backend database system due to it using its own file system to store data.

2.5.2 FireEye

FireEye Network Security is a cyber threat solution. Aimed at helping organizations manage risk of breaches. It does this by identifying and halting advanced, targeted, and other evasive attacks hidden in Internet traffic. With real proof, actionable intelligence, and response process integration, it enables quick resolution of reported security events in minutes. FireEye are capable of achieving wide range of detection, prevention, and response measures. Featuring Multi-Vector execution engine, or MVX as they call it and numerous machine learning, AI and correlation engines. FireEye halts the infection of the cyber-attack by detecting previously unseen exploits. Additionally, rule-based analysis based on real-time insights gained on the front lines from thousands of hours of incident response experience, makes it possible for FireEye to identify and stop obfuscated, targeted, and other tailored assaults. FireEye Network

Security alerts offer actual, tangible evidence that may be used to respond to targeted and recently identified attacks. For contextual support, threats can also be mapped to the MITRE ATT&CK framework.

CHAPTER 3

METHODOLOGY

3.1 Chosen Software Development Methodology

The chosen methodology is Agile. A flexible and efficient workflow should be the priority when deciding the methodology to be adopted for the project. Agile is the preferred choice as it emphasizes flexibility and rapid iteration. It is intended for complicated and rapidly changing projects, and it is especially well-suited for projects with a high level of uncertainty or ambiguity. Agile enables rapid iteration and delivery: Agile divides work into tiny, focused iterations called "sprints," which typically run two to four weeks. This enables quick deployment of working software and receive feedback from clients early in the process. Any improvement or required changes can be identified and applied after reviewing the results of each sprints. Apart from that, Agile values collaboration, transparency, and communication among developers. This can lead to more informed decisions and a more transparent approach. It would be very beneficial to the development as the deliverables and outcomes can be discussed with the project supervisor throughout the project. Agile is more flexible than the waterfall model, which is more rigid and sequential, because it is built to handle project that might experience multiple changes in design and requirements. Developer can thus respond swiftly and apply changes according to the project's needs. Since Agile enable rapid delivery and continual improvement, it can be less expensive and time-consuming compared to the waterfall model. Typically, in a waterfall model, no finished product can be delivered until the project is nearing the end of its development cycle, which is during the deployment of the system. The Agile methodology will not constraint the progress and work deliverables of the project into strict phases as seen in a waterfall model as well. In a waterfall model, the linear and sequential flow of development means that a phase must be completed first before moving on to work on the next development phase. This element restricts the development of the software in some way as the development of this intrusion detection system comprises of many technical and complicated aspect such as programming, machine learning algorithm implementation, dashboards, and datasets to manage. It is believed that a complicated project would be easier to manage if a methodology like Agile is adopted. Scrum, an implementation of Agile would allow the tasks to be divided to smaller chunks or sprints. When tasks are broken into smaller units, more focus can be put to complete the tasks. Moreover, in an Agile development, a small product or deliverable can be consistently put out after each sprint. This product can be evaluated so that any improvement or change in requirements can be identified to allow applying necessary changes in the product. Division of task should also focus on the features of the system. That way the features can be easily reassessed and improved if needed. Furthermore, Kanban may be used to visualize the workflow of the entire development cycle. A Kanban board should be created to list the task tasks to be done, tasks in progress, and finished task. Work in progress limits is set for the board to avoid having too much task to keep track of.

3.2 Proposed System Features

The network intrusion detection model based on real and encrypted synthetic attack traffic using decision tree algorithm offer a few primary features that outlines the model's primary operations. Firstly, the ability for users to insert input or provide data, which is the key features extracted from the traffic data, into the model to be analyzed, and classified by the decision tree model to be either natural traffic or malicious traffic. Second, the capability of the model to analyze traffic data by recognizing unusual patterns or features that may point to malicious activity on the network. This leads to an output that gives an alert and classifies the traffic data in question as malicious traffic. The third feature is a graphical and analytical dashboard that provides the functionality of reporting and displaying the outputs back to the user in a visual form that is organized, practical, and easy to comprehend. The dashboard may include charts, statistics, and other useful analytical features. A key function of the dashboard is to make it easier for users to identify malicious traffic and alerts. Additionally, it offers a better user experience, which should enhance user performance when utilizing the model to analyze data and make decisions.

3.3 Chosen Machine Learning Tool

When developing a network intrusion detection model that implements machine learning, a tool that is both powerful and efficient in order to simplify the process of developing machine learning models and Python codes is required. Therefore, the tool for machine learning that will be used is going to be Google Colab. Seamless Python documentation and coding in a single document are available in an interactive environment through Google Colab, which makes it convenient and practical to test codes. The integration of Google Colab with Google Drive offer support to import data from the Drive as well as saving Colab notebooks into the Drive to make sure progress can be saved quickly. Importing Python libraries into Google Colab requires only a few lines of code and does not require the libraries to be downloaded in advance. This makes it possible to import Python packages that are widely used in machine learning projects. Furthermore, Google Colab enables users to share notebooks with one another and work together on those notebooks. This can be especially helpful when working on projects as part of a group or when seeking input from other people in the same field. Most importantly, access to sophisticated hardware for running machine learning models is available through Google Colab and TensorFlow included. This hardware includes GPUs and TPUs, both of which are able to considerably speed up the training process.

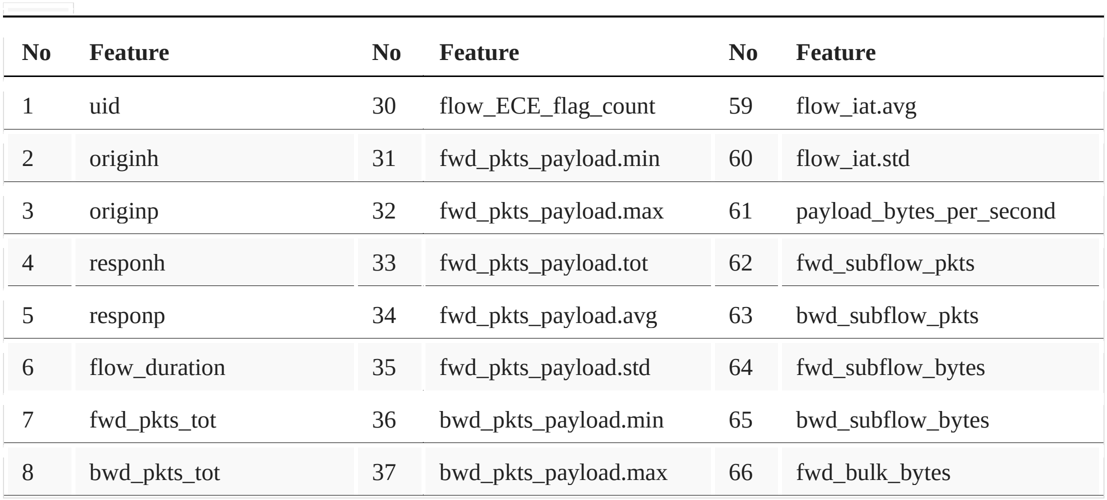
3.4 Chosen Dashboard Tool

This network intrusion detection model comes equipped with a dashboard, which is one of its most important features. The user of the network intrusion detection model should be able to make precise analytical observations and decisions regarding the traffic data with the help of the dashboard which facilitates the visualization of data and results from the machine learning algorithm of the network intrusion detection model. In order to facilitate effortless adoption and utilization on the part of users and developers alike, it is important that the dashboard tool be easy to understand and straightforward in both its configuration and its operation. As a result, following careful consideration of the available choices among the many dashboard tools, the decision was made to use Google Looker Studio as the dashboard tool. Google Looker Studio was chosen for several reasons. Looker can quickly connect to and query a broad variety of data sources. These data sources can range from databases and data warehouses to cloud storage and online storage. To better examine and comprehend the network intrusion detection model data, Looker offers a broad variety of visualization choices, such as charts, graphs, maps, and tables, that can be used to build interactive dashboards. Looker's intuitive interface also allows altering the visual style of the data to suit preferences. When it comes to building and developing the dashboard, the Looker feature that enables collaboration with other users makes it easy to effectively apply modifications and encourage teamwork and communication between developers. This feature is beneficial towards the development and continuity of the dashboard as it invites additional input and suggestions that can be used to improve the dashboard. Furthermore, Looker gives the ability to connect to and analyze data from a wide variety of Google Cloud data sources because the Looker platform has been connected with the Google Cloud platform. In addition to this, a large number of data sources from a variety of services are made available with connector support, which will result in an increased flexibility and variety of options pertaining to data sources that may be used in the project.

3.5 Dataset

A dataset is a crucial component of a machine learning model since it contains the data from which the model will learn. The dataset will be used by the model to uncover patterns and relationships in the data, which will then be used to generate predictions. The quality of the dataset is an important consideration when picking which one to use. The quality of the data in the dataset is key to the machine learning model's performance. The model's predictions or conclusions may be unreliable if the data is incomplete, imprecise, or biased. For this network intrusion detection model, HIKARI-2021 was chosen as the dataset to train the machine learning model. HIKARI-2021 consist of two types of traffic, which are synthetic benign traffic and attacker traffic. HIKARI-2021 were created using a combination of ground-truth data and data that was missing from previous existing IDS datasets. Furthermore, HIKARI-2021 was produced based on a variety of new dataset construction requirements, such requirements are anonymization, payload, ground-truth data, encryption, and practical implementation technique. The entire dataset consists of numerous files, including pcap files from background traffic and synthetic attacks. Features in HIKARI-2021 traffic data are presented in **Table 3.1**.

Table 3.1: HIKARI-2021 traffic data features



No	Feature	No	Feature	No	Feature
1	uid	30	flow_ECE_flag_count	59	flow_iat.avg
2	originh	31	fwd_pkts_payload.min	60	flow_iat.std
3	originp	32	fwd_pkts_payload.max	61	payload_bytes_per_second
4	responh	33	fwd_pkts_payload.tot	62	fwd_subflow_pkts
5	responp	34	fwd_pkts_payload.avg	63	bwd_subflow_pkts
6	flow_duration	35	fwd_pkts_payload.std	64	fwd_subflow_bytes
7	fwd_pkts_tot	36	bwd_pkts_payload.min	65	bwd_subflow_bytes
8	bwd_pkts_tot	37	bwd_pkts_payload.max	66	fwd_bulk_bytes

9	fwd_data_pkts_tot	38	bwd_pkts_payload.tot	67	bwd_bulk_bytes
10	bwd_data_pkts_tot	39	bwd_pkts_payload.avg	68	fwd_bulk_packets
11	fwd_pkts_per_sec	40	bwd_pkts_payload.std	69	bwd_bulk_packets
12	bwd_pkts_per_sec	41	flow_pkts_payload.min	70	fwd_bulk_rate
13	flow_pkts_per_sec	42	flow_pkts_payload.max	71	bwd_bulk_rate
14	down_up_ratio	43	flow_pkts_payload.tot	72	active.min
15	fwd_header_size_tot	44	flow_pkts_payload.avg	73	active.max
16	fwd_header_size_min	45	flow_pkts_payload.std	74	active.tot
17	fwd_header_size_max	46	fwd_iat.min	75	active.avg
18	bwd_header_size_tot	47	fwd_iat.max	76	active.std
19	bwd_header_size_min	48	fwd_iat.tot	77	idle.min
20	bwd_header_size_max	49	fwd_iat.avg	78	idle.max
21	flow_FIN_flag_count	50	fwd_iat.std	79	idle.tot
22	flow_SYN_flag_count	51	bwd_iat.min	80	idle.avg
23	flow_RST_flag_count	52	bwd_iat.max	81	idle.std
24	fwd_PSH_flag_count	53	bwd_iat.tot	82	fwd_init_window_size
25	bwd_PSH_flag_count	54	bwd_iat.avg	83	bwd_init_window_size
26	flow_ACK_flag_count	55	bwd_iat.std	84	fwd_last_window_size
27	fwd_URG_flag_count	56	flow_iat.min	85	traffic_category
28	bwd_URG_flag_count	57	flow_iat.max	86	Label
29	flow_CWR_flag_count	58	flow_iat.tot		

CHAPTER 4

DESIGN

4.1 System Architecture

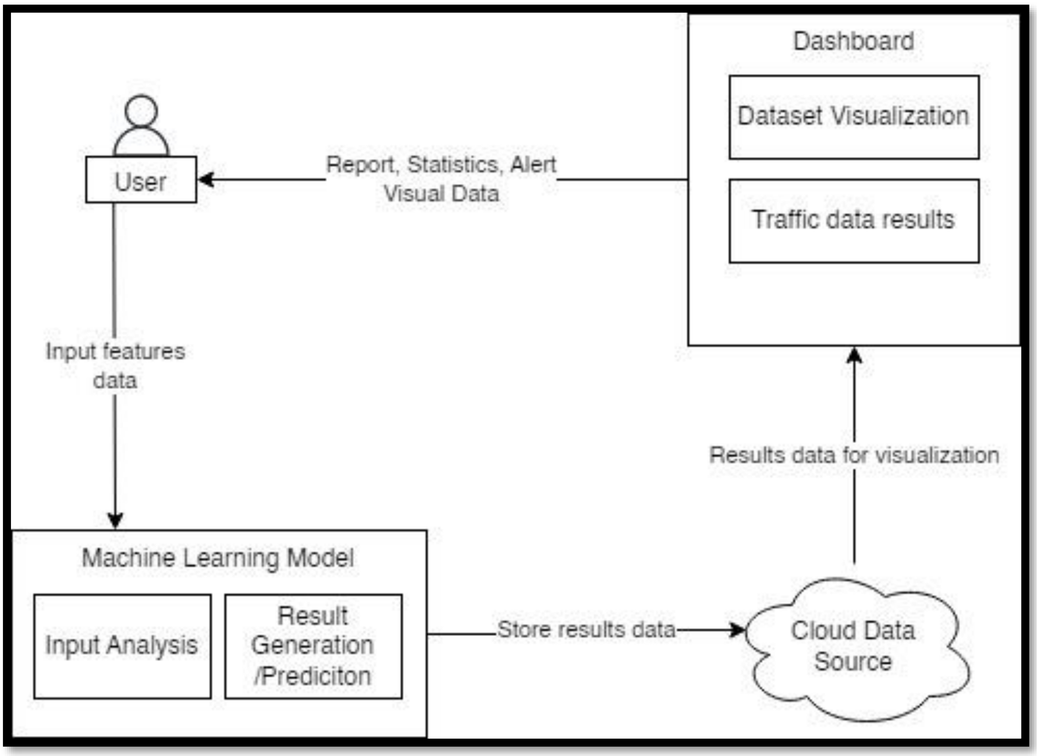


Figure 4.1: Network Intrusion Detection model architecture

The user has an interaction with the network intrusion detection model based on the system architecture shown in **Figure 4.1**. This interaction takes place when the user supplies the model with traffic data. The network intrusion detection model will use the provided traffic data to make predictions and conduct analysis on the features identified in each traffic based on what it learned during the training phase of the model. These predictions and analyses will be based on the features identified in each traffic. The outcomes of the predictions need to be sent to a data source to allow the storage of result data. The data source needs to be linked up to the dashboard platform, which in this case is Google Looker Data Studio. This will allow the data to be formatted into graphical representations and visualized before being shown to the user. User can view the dashboard by heading into Google Looker Data Studio and choosing the dedicated dashboard file of the network intrusion detection model.

4.2 Dashboard

The dashboard will perform the function of serving data visualization, alerts, and statistics to the end user using the results of analyzed traffic from the network intrusion detection model with decision tree machine learning algorithm. Google Looker Data Studio will be used to create the dashboard. Google Looker Data Studio will be linked to a data source that houses the network intrusion detection model's result data in order to enable result data to be shown in the dashboard.

4.3 Flowchart

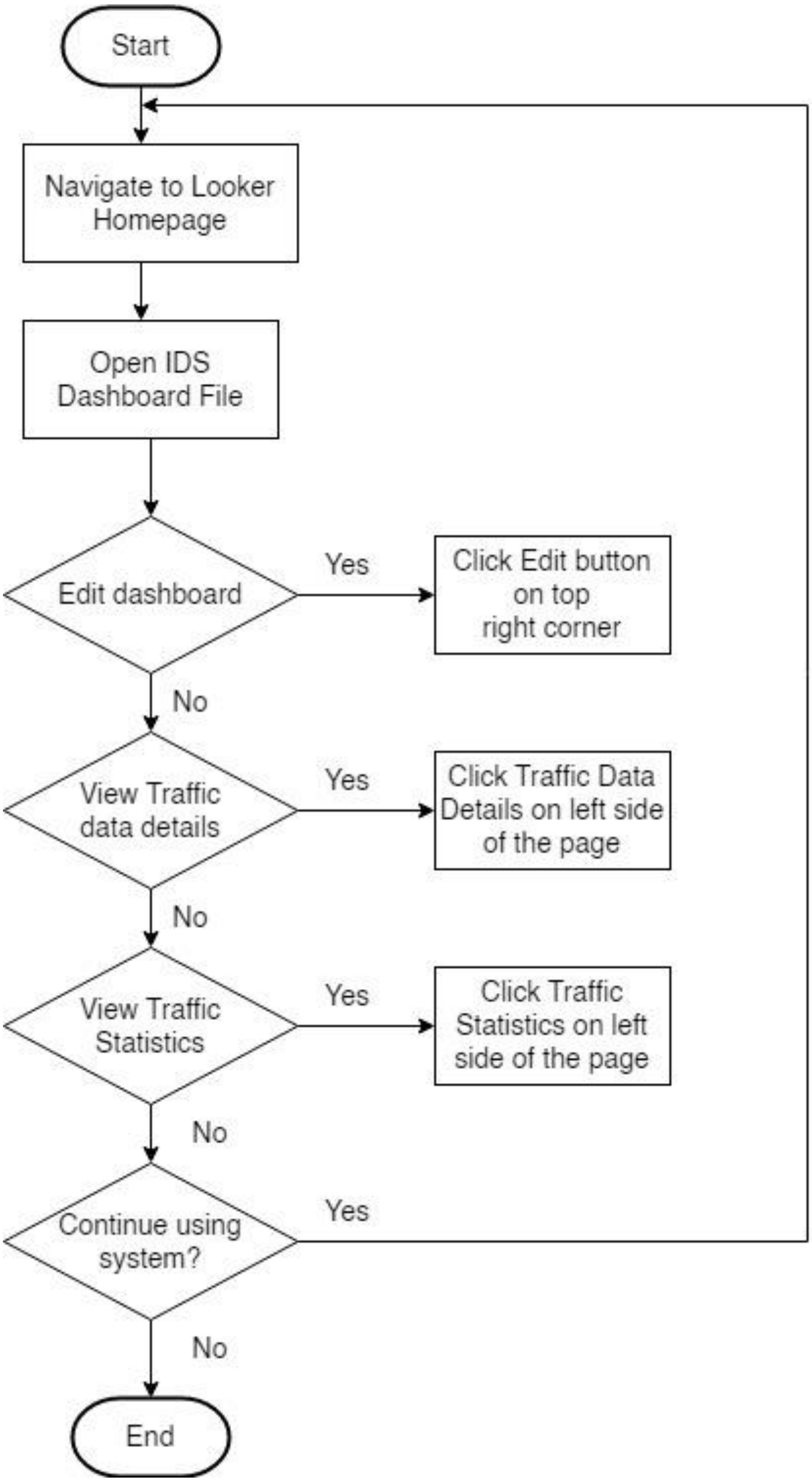
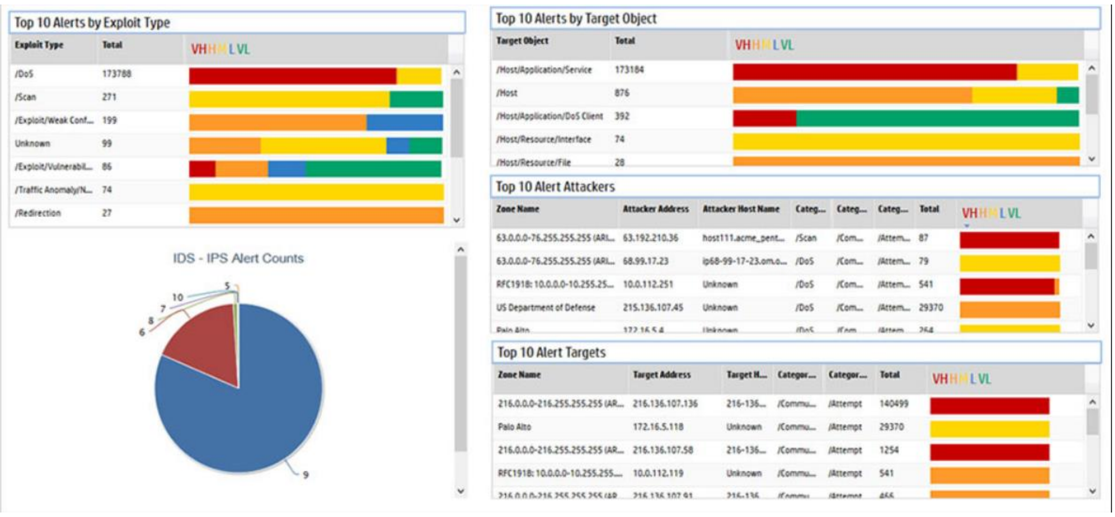

Start
Navigate to Looker Homepage
Open IDS Dashboard File
Edit dashboard
Yes
Click Edit button on top right corner
No
View Traffic data details
Yes
Click Traffic Data Details on left side of the page
No
View Traffic Statistics
Yes
Click Traffic Statistics on left side of the page
No
Continue using system?
Yes
No
End

Figure 4.2: Dashboard Flowchart

The flowchart in **Figure 4.2** displays the overall flow of interaction and navigation when the user accesses the dashboard through Google Looker Studio.

4.4 Dashboard Design

Google Looker Studio offers many templates for an interactive dashboard that can be used by users to quickly and easily setup their dashboard. This would allow to save time when deciding design aspects while focusing effort on choosing the important data and statistic to display. The dashboard for this network intrusion detection model will also adopt common designs taken from other existing intrusion detection systems as these existing dashboards serves as a good reference when developing a functioning dashboard for network intrusion. The output traffic data should be listed and sorted by categorical features such as the port origin, IP address, and so on. The inclusion of visual components such as pie charts is beneficial in displaying the volume of traffic recorded and classifying it as regular or malicious traffic. Additionally, Google Looker Data Studio allows the creation of pages to ease in organizing the data and visualizations according to category. This makes sure the data are displayed in a manner that is easy for the user to observe and analyze.



Top 10 Alerts by Exploit Type

Exploit Type	Total	VH H M L VL
/DoS	173788	
/Scan	271	
/Exploit/Weak Conf...	199	
Unknown	99	
/Exploit/Vulnerabil...	86	
/Traffic Anomaly/N...	74	
/Redirection	27	

Top 10 Alerts by Target Object

Target Object	Total	VH H M L VL
/Host/Application/Service	173184	
/Host	876	
/Host/Application/DoS Client	392	
/Host/Resource/Interface	74	
/Host/Resource/File	28	

Top 10 Alert Attackers

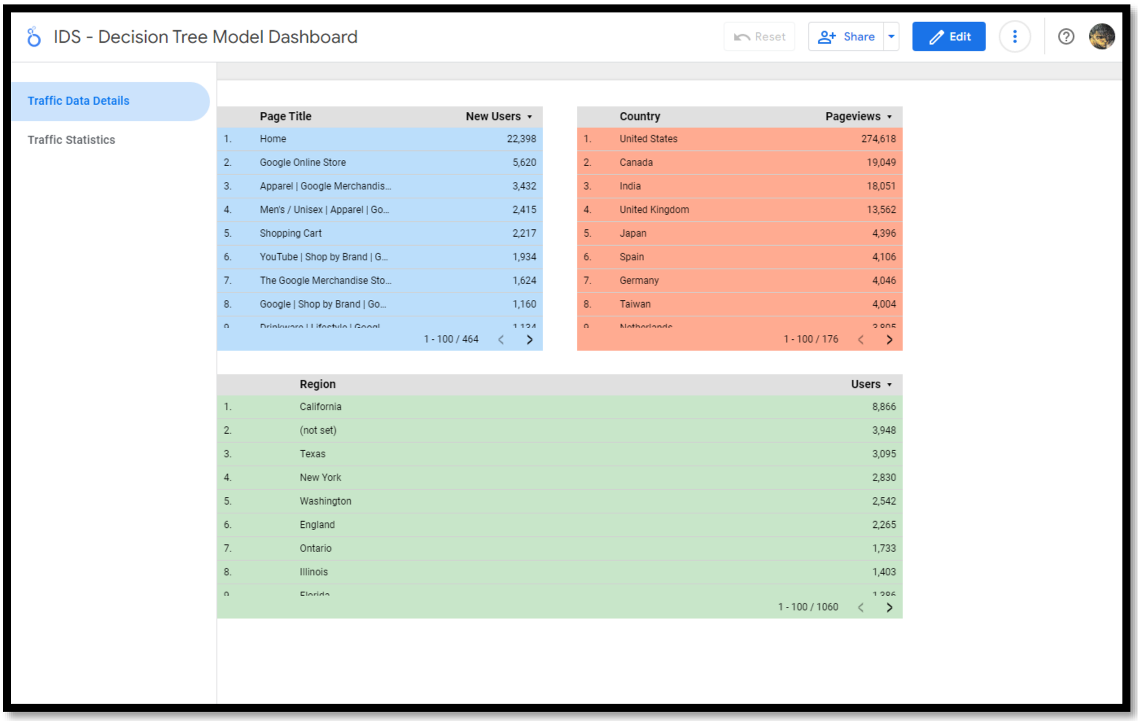
Zone Name	Attacker Address	Attacker Host Name	Categ...	Categ...	Categ...	Total	VH H M L VL
63.0.0.0-76.255.255.255 (ARI...	63.192.210.36	host111.acme_pent...	/Scan	/Com...	/Attem...	87	
63.0.0.0-76.255.255.255 (ARI...	68.99.17.23	ip68-99-17-23.om.o...	/DoS	/Com...	/Attem...	79	
RFC1918: 10.0.0.0-10.255.25...	10.0.112.251	Unknown	/DoS	/Com...	/Attem...	541	
US Department of Defense	215.136.107.45	Unknown	/DoS	/Com...	/Attem...	29370	
Palo Alto	172.16.5.4	Unknown	/DoS	/Com...	/Attem...	264	

Top 10 Alert Targets

Zone Name	Target Address	Target H...	Categor...	Categor...	Total	VH H M L VL
216.0.0.0-216.255.255.255 (AR...	216.136.107.136	216-136...	/Commu...	/Attempt	140499	
Palo Alto	172.16.5.118	Unknown	/Commu...	/Attempt	29370	
216.0.0.0-216.255.255.255 (AR...	216.136.107.58	216-136...	/Commu...	/Attempt	1254	
RFC1918: 10.0.0.0-10.255.255....	10.0.112.119	Unknown	/Commu...	/Attempt	541	
216.0.0.0-216.255.255.255 (AR...	216.136.107.91	216-136...	/Commu...	/Attempt	466	

Figure 4.3: IDS Dashboard Example

Figure 4.3 shows an example of a dashboard for an intrusion detection system that contains visual elements to represents relevant data and statistics.



	Page Title	New Users ▾
1.	Home	22,398
2.	Google Online Store	5,620
3.	Apparel | Google Merchandis...	3,432
4.	Men's / Unisex | Apparel | Go...	2,415
5.	Shopping Cart	2,217
6.	YouTube | Shop by Brand | G...	1,934
7.	The Google Merchandise Sto...	1,624
8.	Google | Shop by Brand | Go...	1,160
9.	Drinkware | Lifestyle | Googl	1,134

1 - 100 / 464 < >

	Country	Pageviews ▾
1.	United States	274,618
2.	Canada	19,049
3.	India	18,051
4.	United Kingdom	13,562
5.	Japan	4,396
6.	Spain	4,106
7.	Germany	4,046
8.	Taiwan	4,004
9.	Netherlands	3,805

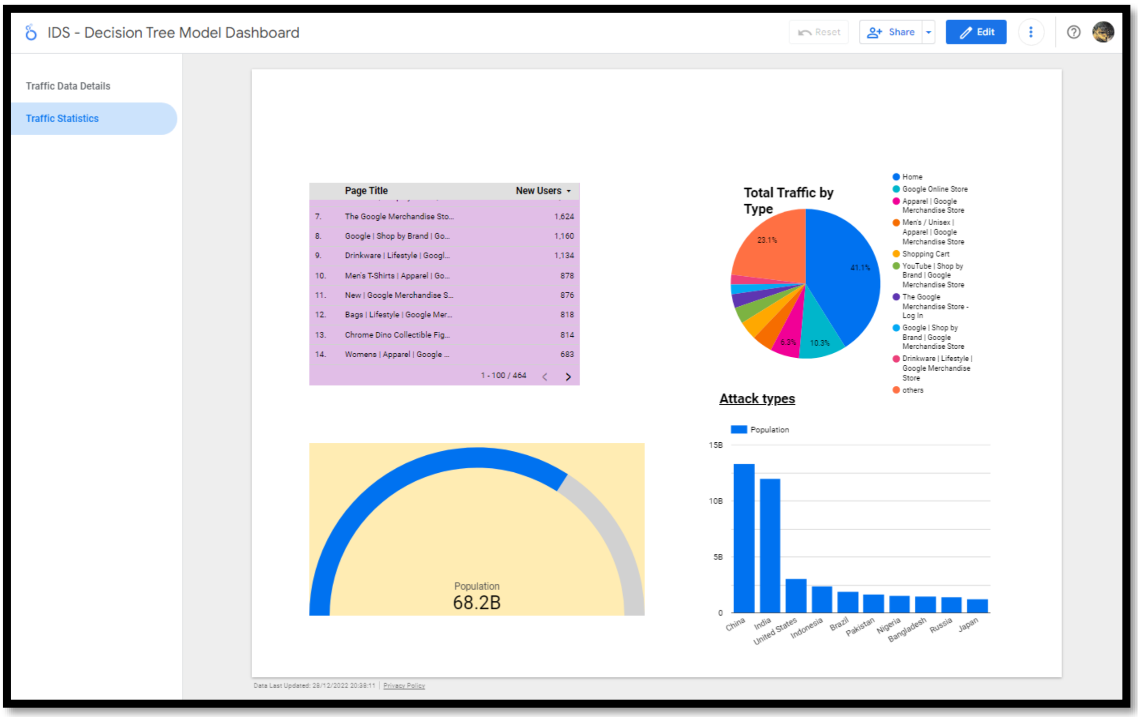
1 - 100 / 176 < >

	Region	Users ▾
1.	California	8,866
2.	(not set)	3,948
3.	Texas	3,095
4.	New York	2,830
5.	Washington	2,542
6.	England	2,265
7.	Ontario	1,733
8.	Illinois	1,403
9.	Florida	1,386

1 - 100 / 1060 < >

Figure 4.4: Google Looker Dashboard Concept Page 1

Figure 4.4 shows the usage of table elements in Google Looker Data Studio to organize data in a list and how its placed in one page dedicated to only tables. The tabs on the left side can be used by the user to navigate to different pages. Configurations of the dashboard can be performed by clicking the Edit button which allow user to enter edit mode.


IDS - Decision Tree Model Dashboard
Reset
Share
Edit
Traffic Data Details
Traffic Statistics
Total Traffic by Type
41.1%
23.1%
10.3%
6.3%
Home
Google Online Store
Apparel | Google Merchandise Store
Men's / Unisex | Apparel | Google Merchandise Store
Shopping Cart
YouTube | Shop by Brand | Google Merchandise Store
The Google Merchandise Store - Log In
Google | Shop by Brand | Google Merchandise Store
Drinkware | Lifestyle | Google Merchandise Store
others
Attack types
Population
15B
10B
5B
0
China
India
United States
Indonesia
Brazil
Pakistan
Nigeria
Bangladesh
Russia
Japan
Population
68.2B

	Page Title	New Users ▾
7.	The Google Merchandise Sto...	1,624
8.	Google | Shop by Brand | Go...	1,160
9.	Drinkware | Lifestyle | Googl...	1,134
10.	Men's T-Shirts | Apparel | Go...	878
11.	New | Google Merchandise S...	876
12.	Bags | Lifestyle | Google Mer...	818
13.	Chrome Dino Collectible Fig...	814
14.	Womens | Apparel | Google ...	683

1 - 100 / 464 ‹ ›

Data Last Updated: 28/12/2022 20:38:11 | Privacy Policy

Figure 4.5: Google Looker Dashboard Concept Page 2

Figure 4.5 shows the conceptual page in Google Looker Data Studio that provides visual representations of analyzed data in forms that are easy to observe such as pie charts, gauge, bar graph, and table.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

Machine learning is a powerful that can be used to enhance various application in numerous fields of technology. With regards to network security and network intrusion detection system, machine learning has the potential to dramatically increase the performance of intrusion detection systems by allowing computers to learn from data and make predictions or choices based on that learning. However, applying machine learning in IDS systems can be difficult due to the time and effort required for proper planning, data administration, model selection, and training. It is also worth mentioning that the usage of recent datasets to train machine learning models has various advantages. Organizations can increase the accuracy and effectiveness of their models by training them on data that is current and relevant to the problem at hand. Furthermore, training a model on recent data can assist ensure that it can react to changing conditions or patterns over time. Based on the activities and research conducted in this project so far, numerous machine learning concepts and information on network intrusion detection systems were studied to provide understanding of the topic. However, there is still much more to be studied and explored as the project continue in its development cycle due to the vast subject that is machine learning. It is crucial that knowledge on the particular subject to be greatly understood in order to establish solid reasoning and ensure success in the project. Overall, leveraging recent datasets to train machine learning models can help organizations make better informed and accurate decisions, and can be a critical aspect in the success of a machine learning project. The application of machine learning in network intrusion detection system may also prove to be beneficial as it can improve the detection of malicious traffic which will led to the improvement of network security.

5.2 Future Works

The future work of the project will be focusing on the development of the model which will first cover the management and implementation of the datasets. This involves preparing the dataset by completing several important steps. These steps include exploring the dataset and identifying any errors and important details. After that, cleaning the datasets to correct any errors and missing values, alongside deciding which features to extract from the traffic data. Additionally, the dataset should be split into training set and testing set for the machine learning model's training and testing. After data management, focus should be put into the development of the machine learning model. In this project, further research on decision tree implementation using Python will be conducted to ensure great understanding and progress are achieved when writing the machine learning model code. This is also to ensure the machine learning model can satisfy the expected outcome in terms of the performance of the model. As mentioned before, Google Colab will be the platform to write and execute the model written in Python. Furthermore, integration between the network intrusion detection model and dashboard will be explored more. This is to make sure the dashboard can achieve great results in displaying the required results and data from the network intrusion detection model. To sum up, future works would be the development of the machine learning model for the network intrusion detection model and the dashboard.

REFERENCES

[1] "Agile Methodologies: A Comprehensive Guide." Atlassian. Internet: https://www.atlassian.com/agile

[2] "The Agile Manifesto." Internet: https://agilemanifesto.org

[3] Machine Learning Mastery. Internet: https://machinelearningmastery.com

[4] "Applications of Machine Learning" Medium. Internet: https://medium.com/swlh/applications-of-machine-learning-9f77e6eb7acc

[5] Yasir Hamid1, M. Sugumaran and V. R. Balasaraswathi . 2016. IDS Using Machine Learning – Current State of Art and Future Directions. *British Journal of Applied Science & Technology 15(3): 1-22, Article no.BJAST.23668 ISSN: 2231-0843, NLM ID: 101664541*

[6] "Decision Tree." GeeksforGeeks. Internet: https://www.geeksforgeeks.org/decision-tree/

[7] "Decision Trees." sklearn documentation. Internet: https://scikit-learn.org/stable/modules/tree.html

[8] Google Colab "Getting Started". Internet: https://colab.research.google.com

[9] Grafana "Everything You Should Know About It". Internet: https://scaleyourapp.com/what-is-grafana-why-use-it-everything-you-should-know-about-it/

[10] What Is Splunk? A Beginners Guide To Understanding Splunk. 2022. Internet: https://www.edureka.co/blog/what-is-splunk/

[11] FireEye Network Threat Prevention Platform. Internet: https://www.fireeye.com/content/dam/fireeye-www/products/pdfs/pf/web/fireeye-network-threat-prevention-platform.pdf

[12] Google Data Studio Support. Internet: https://support.google.com/looker-studio/answer/6283323?hl=en

[13] Anaconda Documentation. Internet: https://docs.anaconda.com/navigator/index.html

[14] What is Weka. Internet: https://www.tutorialspoint.com/weka/what_is_weka.htm

[15] What is .NET. Internet: https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet

[16] M. Somvanshi, P. Chavan, S. Tambade and S. V. Shinde, "A review of machine learning techniques using decision tree and support vector machine," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), 2016, pp. 1-7, doi: 10.1109/ICCUBEA.2016.7860040.

[17] A. Navada, A. N. Ansari, S. Patil and B. A. Sonkamble, "Overview of use of decision tree algorithms in machine learning," 2011 IEEE Control and System Graduate Research Colloquium, 2011, pp. 37-42, doi: 10.1109/ICSGRC.2011.5991826.

[18] SDLC – Waterfall Model. Internet: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

[19] Abid Ali Awan, 2022. The Machine Learning Life Cycle Explained. Internet: https://www.datacamp.com/blog/machine-learning-lifecycle-explained

[20] Jafar Alzubi et al 2018 J. Phys.: Conf. Ser. 1142 012012

[21] Ferriyan, A.; Thamrin, A.H.; Takeda, K.; Murai, J. Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic. Appl. Sci. 2021, 11, 7868. https://doi.org/10.3390/app11177868

[22] Seldon, 2021, Decision Tree In Machine Learning. Internet: https://www.seldon.io/decision-trees-in-machine-learning \

[23] Bhuvaneswari Gopalan, 2020. “Is Decision Tree a classification or regression model?” Internet: https://www.numpyninja.com/post/is-decision-tree-a-classification-or-regressionmodel