Data Analytics & Cyber Security

Team Project & Group Dynamics Module

Network Intrusion Detection System

Proposal, Technical Project

to be presented on: *24/10/2022*

Ingrid Melin – K00258766

Patryk Kaiser – K00263702

Daniel Mackey – K00263905

Project Supervisor(s): Tom Davis and Aileen O'Mara

TABLE OF CONTENTS

1. INTRODUCTION

1.1 PURPOSE

The purpose of this project is to develop a network intrusion detection system that uses machine learning to analyze network traffic and log cyber-attacks.

We propose to implement this system using supervised machine learning (SML). SML makes use of labeled data to help train an artificial intelligence (AI) which will classify network traffic based on features they share and detect potential cyber-attacks.

Developing and training the model will require us to acquire a dataset of packets representing different types of a cyber-attack. We aim to use the Canadian Institute of Cybersecurity dataset as our research shows that the CIC has some of the most modern data.

Our system will be developed to detect the following common types of attack which we will refer to as anomalies:

- Denial of Service (DoS)
- TCP scan
- UDP scan
- SYN scan
- ICMP scan
- SSH/FTP brute force using dictionary

Once the system has detected an anomaly, it will log details of the anomaly. This will show up as a warning which network administrators will view through a web front end.

The front-end for our system will also include statistical visualization of network traffic and anomaly logs. This will give network administrators an overview of network activity with the ability to filter events by source IP, destination IP, ports, and protocols used.

The main benefit of a network-based intrusion detection system is that large networks can automatically be monitored based on known attack patterns or types. Using analytics, we can perform a risk assessment, quantifying collected attack data to determine which components of a network need more attention based on attack type history. This early detection is crucial so that network specialists can mitigate attacks before they grow and worsen.

1.2 BACKGROUND

Programs like Metasploit are widely available and easy to use and it's critical for networks to have an automatic monitoring mechanism to prevent costly downtime or possibly a more damaging attack where the attacker manages to infiltrate the system, gaining access entirely and under the radar undetected.

Existing products like what we are developing here include Snort, an open-source IDS and IPS written in C/C++. Snort was acquired by Cisco in 2013.

We researched datasets commonly used in IDS projects and found that most open-source IDS implementations use the KDD Cup 1999 dataset. The KDD Cup 1999 dataset has been criticized for being outdated.

We studied various datasets including ones found on Kaggle and concluded that the Canadian Institute of Cybersecurity datasets will be the most modern and therefore more suitable for our project.

Another existing product in this market is Suricata which is also open source.

Heimdal Threat Prevention is a Host Intrusion Detection System which operates on a yearly subscription model.

1.3 NEEDS STATEMENT

We believe that implementing new solutions to cybersecurity problems is key in protecting business and organization assets. While a lot of damage is done using phishing and spear phishing methods, a lot of network systems are still vulnerable against even the least sophisticated kinds of cyber-attacks.

Our group researched the products available on the market today and found that a lot of professional solutions provided inadequate detection. Another problem we have encountered in our research is false positives.

We intend to lower the number of false positives by using a modern dataset and studying various methods for improving our machine learning algorithm's accuracy including but not limited to: Naive Bayes Gaussian Algorithm and a decision tree algorithm.

Our research shows that the decision tree algorithm will be most useful when trying to minimize false positives and improving detection accuracy.

Heimdal Threat Prevention's reviews highlight an inability to detect most malicious files. This is a system that advertises itself as a Host Intrusion Detection System and these apparent shortcomings in detecting malware do not justify the price of the yearly subscription (~$60 per annum).

Snort IDS has a good reputation as being a solid open-source implementation of an Intrusion Detection and Prevention System. We aim to provide basic functionality like Snort which is praised for an easy setup process and real time traffic monitoring.

1.4 SCOPE

The overall scope of our project is to develop a Network Intrusion Detection System which when deployed will analyze network traffic and alert administrators to the following threats:

- Denial of Service (DoS)
- TCP scan
- UDP scan
- SYN scan
- ICMP scan
- SSH/FTP brute force using dictionary

We have chosen these attacks as they are the most common threats facing business and organization networks. These are also attacks we can realistically identify and protect against using our system.

We excluded more complex attacks from the scope of this project. We are open to the possibility of expanding the pool of attacks our system will be able to detect.

We will build an SML model to learn from the training data and labels provided by our completed dataset. We aim to build the model, train the model, and then improve the model's accuracy using fresh training data from the honeypot.

Our initial proposal included the creation of a honeypot to work alongside our detection system. Data from the honeypot was theorized to improve the detection capabilities of our program.

We have decided to focus only on the IDS as we were unable to fit the honeypot idea into the context of the program. We hope that this decision will allow us to put more time into developing the capabilities of our IDS program.

The final part of our project is to create the front-end for our system. We aim to develop a front-end using HTML, CSS, JavaScript, and MySQL.

The main body of the code will be built using Python. We have decided to use Python over R for both the Machine Learning and detection program. The reason for this was that while R is a powerful language, we simply have a lot more networking functionality with Python in addition to the Machine Learning libraries available to us.

1.5 PROJECT MEMBERS

Team Member	Role	Contact Information	Responsibilities
Patryk	Champion	K00263702@student.lit.ie	Keep the team motivated. Liaise with outside parties such as supervisors.
Ingrid, Daniel, Patryk	Stakeholder	K00258766@student.lit.ie	Has a stake in the project.
Ingrid	Project Manager	K00258766@student.lit.ie	Set deadlines, distribute workload, GitHub repository management.
Daniel, Patryk	Architect	K00263905@student.lit.ie	Design the structure and system architecture of our project.
Ingrid, Daniel	Analyst	K00263905@student.lit.ie	Preprocess and analyse our data.
Ingrid, Daniel, Patryk	Developer	K00263702@student.lit.ie	Develop and train the machine learning model. Develop the IDS system to use our trained model.

2. PROPOSED TECHNICAL APPROACH

2.1 REQUIREMENTS

Our IDS requires a dataset which will be used to train the SML model.

An SML model trained on our chosen dataset will be required for our system to detect network anomalies.

The detection program is required to sniff packets on the network and using the SML model it would predict if the packet captured is an anomaly. The use case diagram for the main functionality of the IDS is depicted in figure 1.

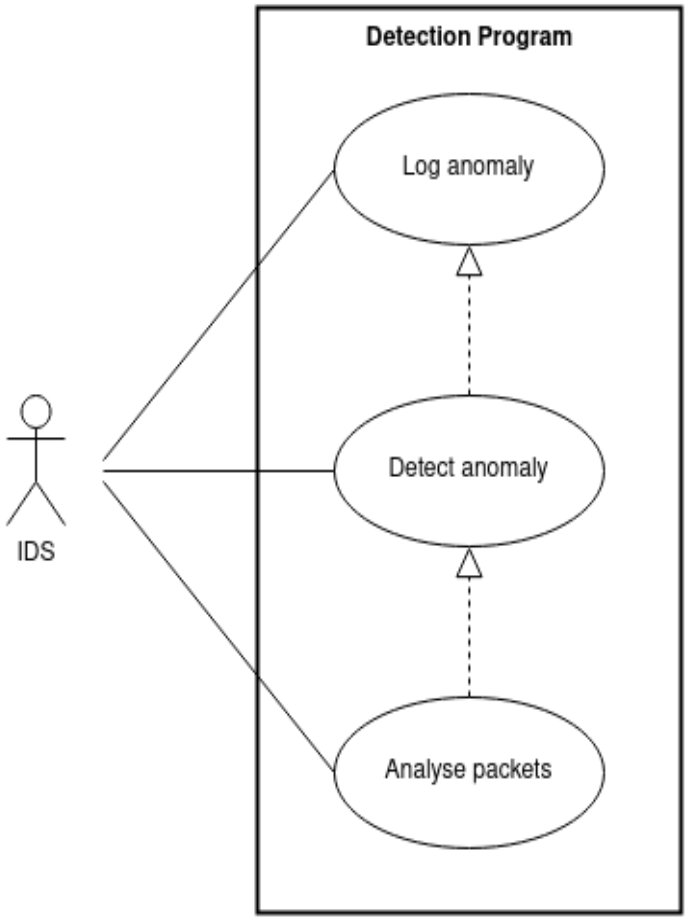

Figure 1: Main functionality of our detection program illustrated in a use case diagram.

A database holding the detected anomaly logs is required.

A front-end website is required to allow administrators to log in and view anomaly logs and network statistics. The use case diagram for our front-end can be seen in figure 2.

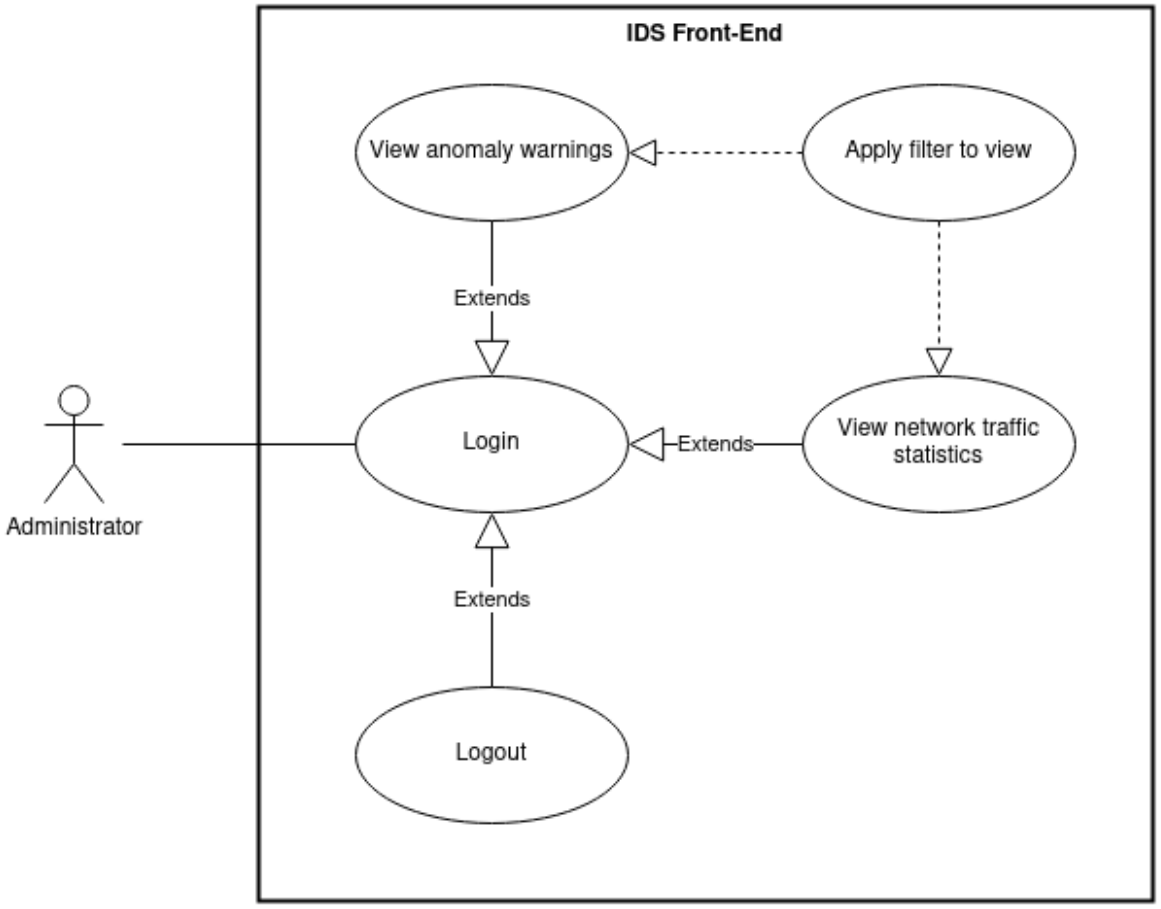

Figure 2: The front-end website's functionality captured in a use case diagram.

2.2 SYSTEM MECHANICS

The IDS program will be running on a host machine and analyzing packets. Once an anomaly is detected the system will match it to known attack types and log the anomaly in the database.

The website front-end will fetch anomaly logs from the database and visualize them as well as display the most recent anomalies detected.

The administrator will log into the website and a session will be established.

2.3 PROTOTYPE/STORYBOARD

We have worked on a prototype for our machine learning prediction model using Scikit-learn.

To avoid redundant data, we removed null feature entries for analysis purposes.

```
y = df[['Label']]
print(y)
```

```
     Label
0        0
1        0
2        0
3        0
4        0
..     ...
547      1
548      1
549      1
550      1
551      1
```

Figure 3: Label column, or classifier which is also the dependent variable. Responsible for ultimately deciding detection system decision making. 0 is good, 1 is bad. These may also represent subsets of a larger category of attack types and be assigned to them later.

```
x = df.drop(['Label', ], axis = 1)
print(x.head())
```

```
   Duration_window_flow  Avg_delta_time  Min_delta_time  Max_delta_time  \
0              0.000125        0.000013             0.0        0.000125
1              0.000150        0.000015             0.0        0.000150
2              0.004210        0.000421             0.0        0.003305
3              0.003275        0.000327             0.0        0.001006
4              0.002161        0.000216             0.0        0.000954

   StDev_delta_time  Avg_pkts_lenght  Min_pkts_lenght  Max_pkts_lenght  \
0          0.000042           1445.4             1008             1494
1          0.000050           1404.4              598             1494
2          0.001094            874.2               60             1494
3          0.000378             60.0               60               60
4          0.000375             60.0               60               60

   StDev_pkts_lenght
0         153.686694
1         283.340078
2         675.330010
3           0.000000
4           0.000000
```

Figure 4: Drop everything but the label feature, the independent features which will decide the class above.

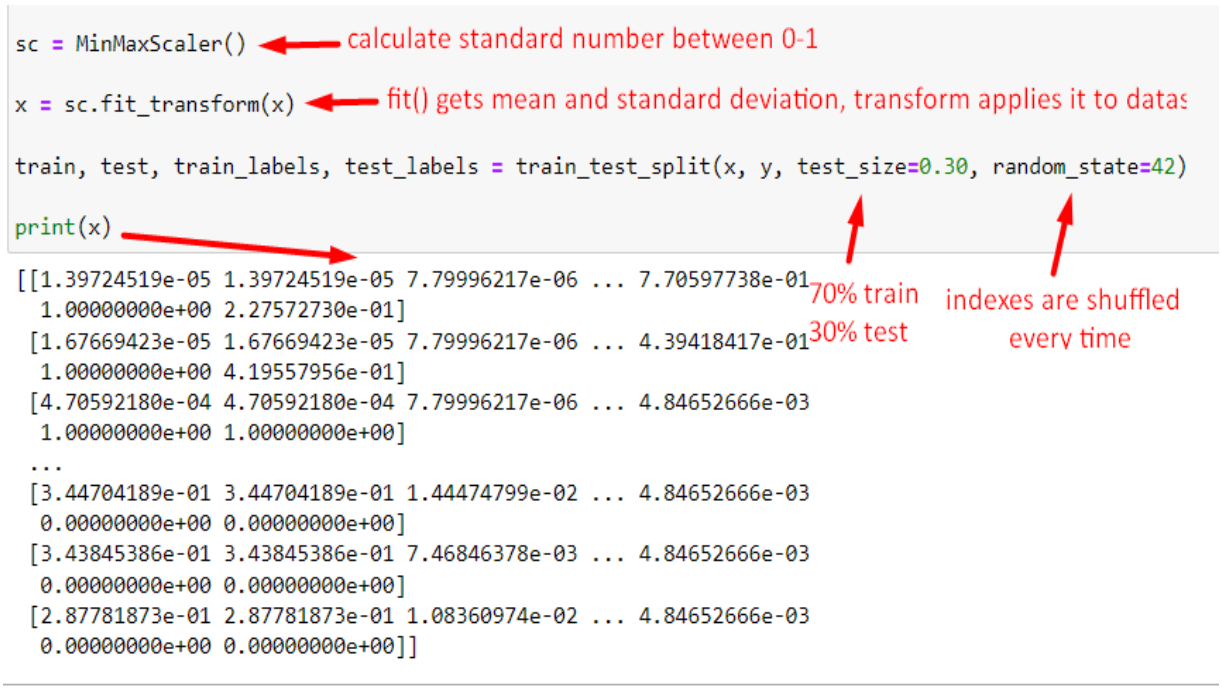

Figure 5: The train_test_split function produces 4 parameters, depending on the value of x, y, and test size. Train and test are the original data, or 70%. Train_labels and test_labels are the test data, or 30%. Produces a unique identifier for every packet or packets in each timeframe.

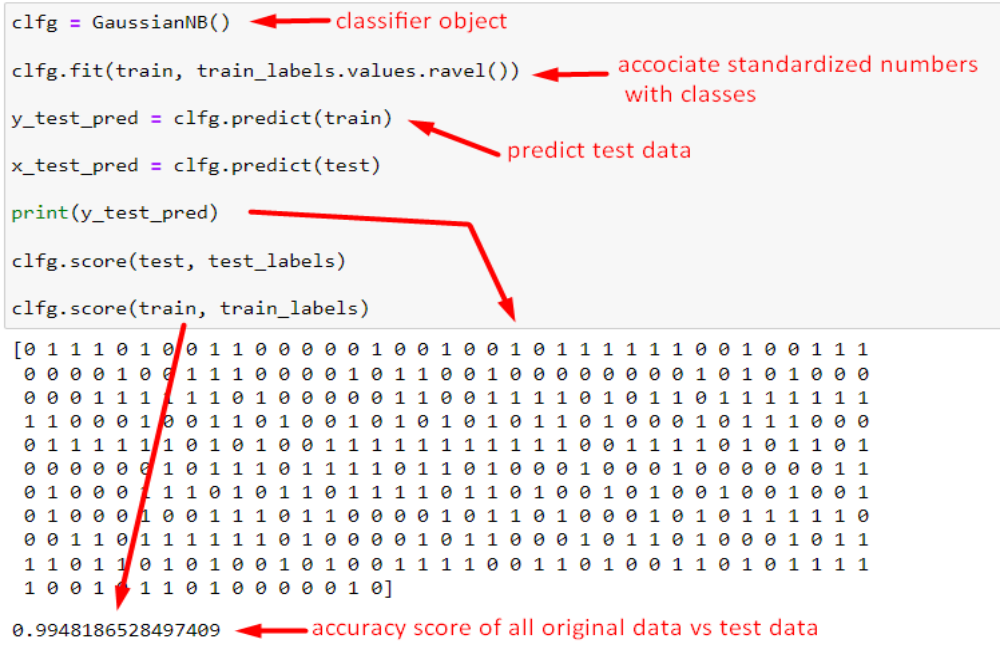

Figure 6: The Naive Bayes Gaussian Algorithm, provides supervised learning or probabilistic classification to ensure our incident detection system identifies the threat.

2.4 ARCHITECTURE DESIGN

The SML will be done using Python's TensorFlow and Scikit-learn libraries.

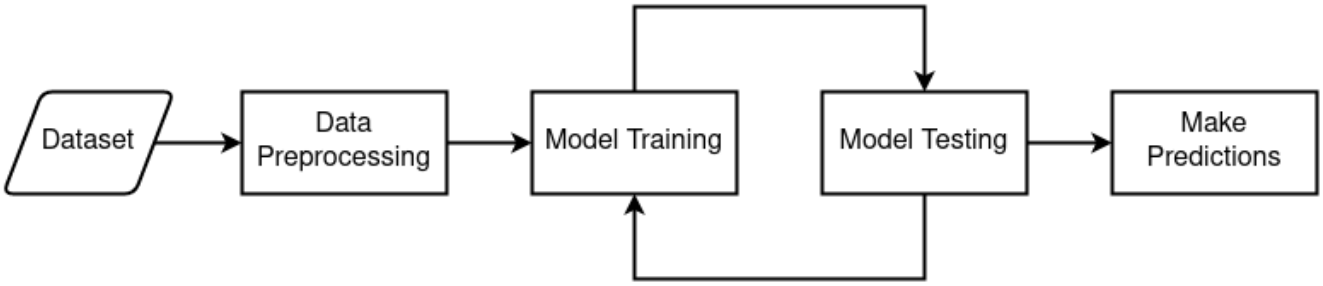

Figure 7: The architecture our SML model will be using. The dataset is preprocessed and fed into the training and testing sets. The model will be scored on the accuracy of the predictions and finally we can use the model to make predictions once a satisfactory score is reached.

The proposed location of our IDS within a business/organization network is between the router and firewall. Our research shows that this is the most lucrative area for detecting anomalies as it allows the administrators to add rules to the firewall according to the detections made by our system.

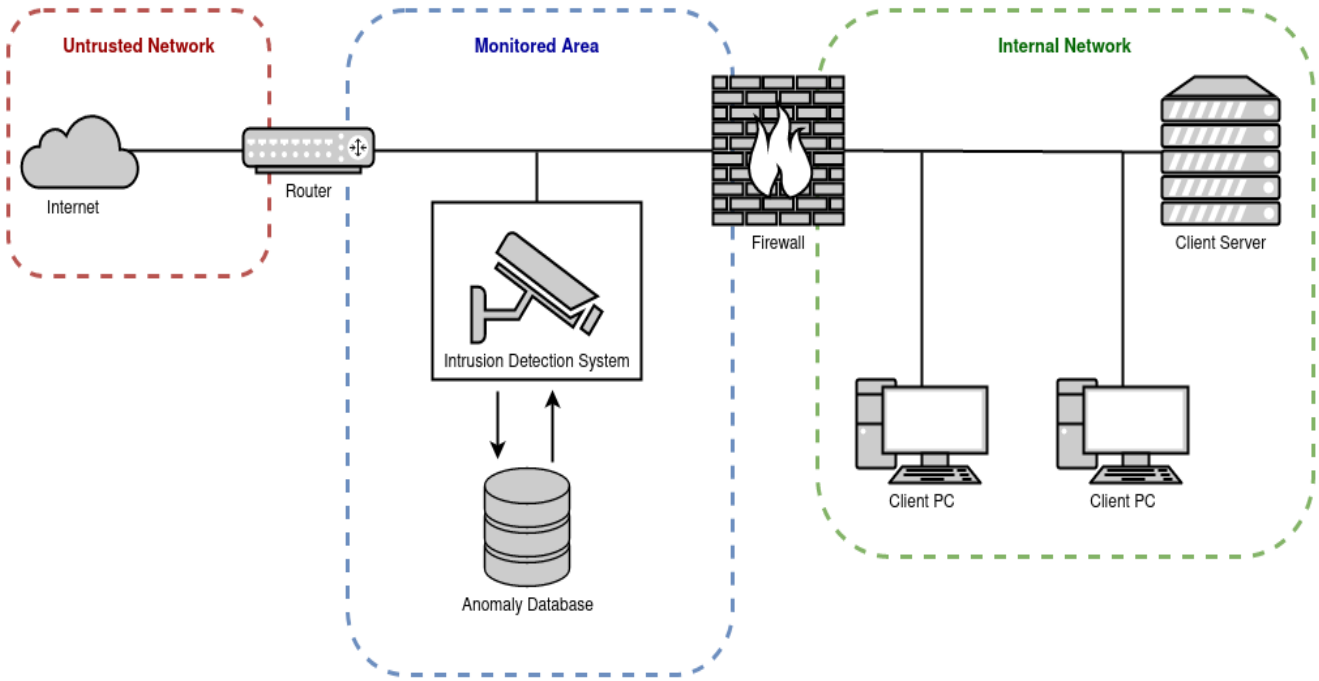

Figure 8: The proposed location of our intrusion detection system within a small network.

We are looking to implement our IDS by running it on a Raspberry Pi host. The advantages of using a Raspberry Pi are that they are lightweight and cost-efficient devices.

The downside of using Raspberry Pi to host our IDS is that they might lack processing power. We can mitigate this problem by transferring the machine learning process over to a computer running more powerful hardware or even to host it on the cloud using one of many machine learning cloud deployment services.

We have designed a basic flowchart diagram to show the 'main system loop'. This will be the programs running state. The program will sniff packets, add them to a buffer and then analyse them one by one. Normal traffic will be ignored, and any anomalies detected will be logged into the database and a notification will be sent to the front-end.

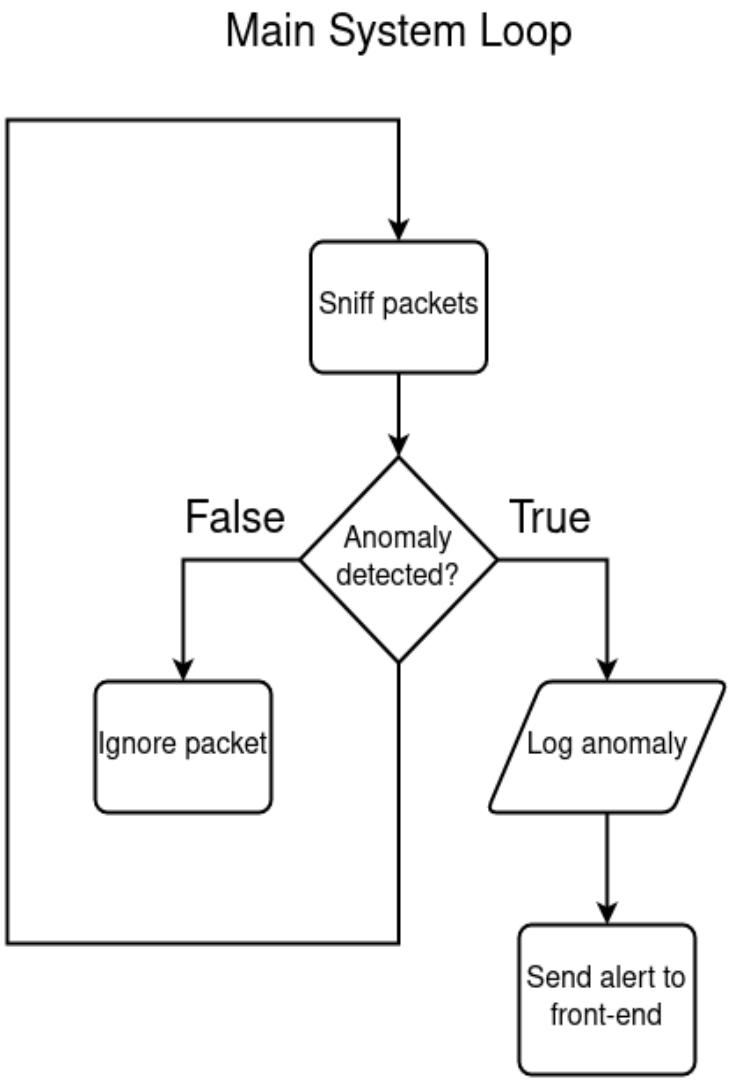

Figure 9: This flowchart illustrates the running loop of our IDS. The system sniffs packets on a network and uses the SML model to predict if it is an anomaly. If an anomaly is detected the system will log the anomaly and send an alert to the front-end website.

Our front-end website will be remotely hosted and accessible through the browser. We will be fetching data from the anomaly log database and populating tables, graphs, and charts with the added functionality of filtering by time, source, destination, port, protocol, etc.

2.5 IMPLEMENTATION

For this project we propose to implement the use of Agile methodology. Our reason for this being that our project manager has used Agile methods such as Scrum and Kanban before in a working environment and it has proven to be very beneficial.

Scrum is a methodology aimed at simplifying a project. We intend to break down the workload of this project into manageable parts and distribute these parts evenly among the team members. Once this is done, we will take note of each task and what member was assigned what tasks, then we will put them into a Kanban board.

A Kanban board is an agile method that allows someone to track their progress on a project with given deadlines. The team members can see how close they are to completing that project by moving their tasks into the correct categories.

Our team has an advantage of having a project manager with experience in using the Agile methodology to lead a project.

The entire team has good knowledge of the ‘make it work’, ‘make it better’ and ‘make it best’ approach when developing software. We aim to apply this approach to our group project by first creating a working IDS and front-end, then improving our system’s accuracy and finally making the best system we can by adding extra features.

2.6 QUALITY ASSURANCE PLAN

The main risk is impeding the performance of the network. We intend to manage this risk by performing the network sniffing and prediction in a non-intrusive way.

We do not intend to allow our program to stop any traffic, the IDS will only alert the administrators of anomalies detected and the administrators will have to react.

Another potential risk we have identified is the possibility of false positives. To lower the risk of false positives we will train our SML model with a high-quality dataset containing data on modern implementations of cyber-attacks.

Our system will not be able to detect zero-day exploits and new kinds of attacks. The pool of attacks we will be able to detect using our system will be small. More research will have to be conducted to manage this risk.

3. EXPECTED PROJECT RESULTS

3.1 MEASURES OF SUCCESS

Our main measure of success is achieving an accuracy rating of ~95% for detections/predictions using our machine learning algorithm.

We have accepted the fact that false positives will be impossible to wipe out fully therefore we agreed upon setting an acceptable limit to the number of false positives our system can give. We are looking to produce no more than one false positive per day of 24-hour surveillance on a network using our IDS. Further research will need to be undertaken to apply a realistic limit to this measure of success.

We will be classifying packets depending on the packet signature, identified by its payload size, flags, and whether it's a TCP or UDP packet, its corresponding total length among others, and especially the calculated variance of these factors such as duration window flow indicating an overload in transmission, too much for the receiver to handle, against other incoming packets, harmful packets can be distinguished between normal ones and a decision must be made.

However, to categorize these connections properly and avoid mitigating legitimate traffic an accuracy score will be determined using a classification algorithm such as either a decision tree, gaussian naive baye, or random forest.

We will be experimenting with various classification algorithms.

Once a model is created by transforming, fitting, splitting, training, and testing data, an accuracy score is made by comparing the output of the predicted test data with the original test data and assigned a percentage out of 100. We would be looking for an accuracy score above 90, or ideally 99 percent accuracy score. 100 may not be attainable especially if the dataset is large.

4. PROJECT MANAGEMENT

4.1 DEVELOPMENT METHODOLOGY

For this project we propose to implement the use of Agile methodology. Our reason for this being that our project manager has used Agile methods such as Scrum and Kanban before in a working environment and it has proven to be very beneficial.

Scrum is a methodology aimed at simplifying a project. We intend to break down the workload of this project into manageable parts and distribute these parts evenly among the team members. Once this is done, we will take note of each task and what member was assigned what tasks, then we will put them into a Kanban board.

A Kanban board is an agile method that allows someone to track their progress on a project with given deadlines. The team members can see how close they are to completing that project by moving their tasks into the correct categories.

Our team has an advantage of having a project manager with experience in using the Agile methodology to lead a project.

The entire team has good knowledge as a whole of the ‘make it work’, ‘make it better’ and ‘make it best’ approach when developing software. We aim to apply this approach to our group project by first creating a working IDS and front-end, then improving our system’s accuracy and finally making the best system we can by adding extra features.

4.2 SCHEDULE

The project plan we have come up with follows two Agile methodologies called Kanban and Gantt.

A Gantt chart is a brilliant tool for visualizing tasks and for project timelines and a Kanban board is excellent for displaying the statuses of the project's tasks. Both tools work well on their own, but they work even better together. That is how we propose to use these tools in our project.

Figure 10 depicts a screenshot image of our project’s Kanban board. This board will help us get an idea for the completion status of each task.

As you can see from figure 10 two sprints on the board in various stages of completion. The first sprint indicated by the green cards has two tasks.

The first task has been moved into the “Done” category, meaning that the task is finished. The second and final task of the first sprint is under the “In Progress” category, meaning that this task is in the process of being finalized.

The second Sprint on the board indicated by the yellow cards, are lined up under the “To Do” category. This means that once the second task is finished and has been moved in to the “Done” category, the second sprint can be moved onto the “In Progress” category.

Figure 10: Screenshot of our Kanban Board with tasks. Each task has a title, description, deadline date, estimated time of completion and Assignee.

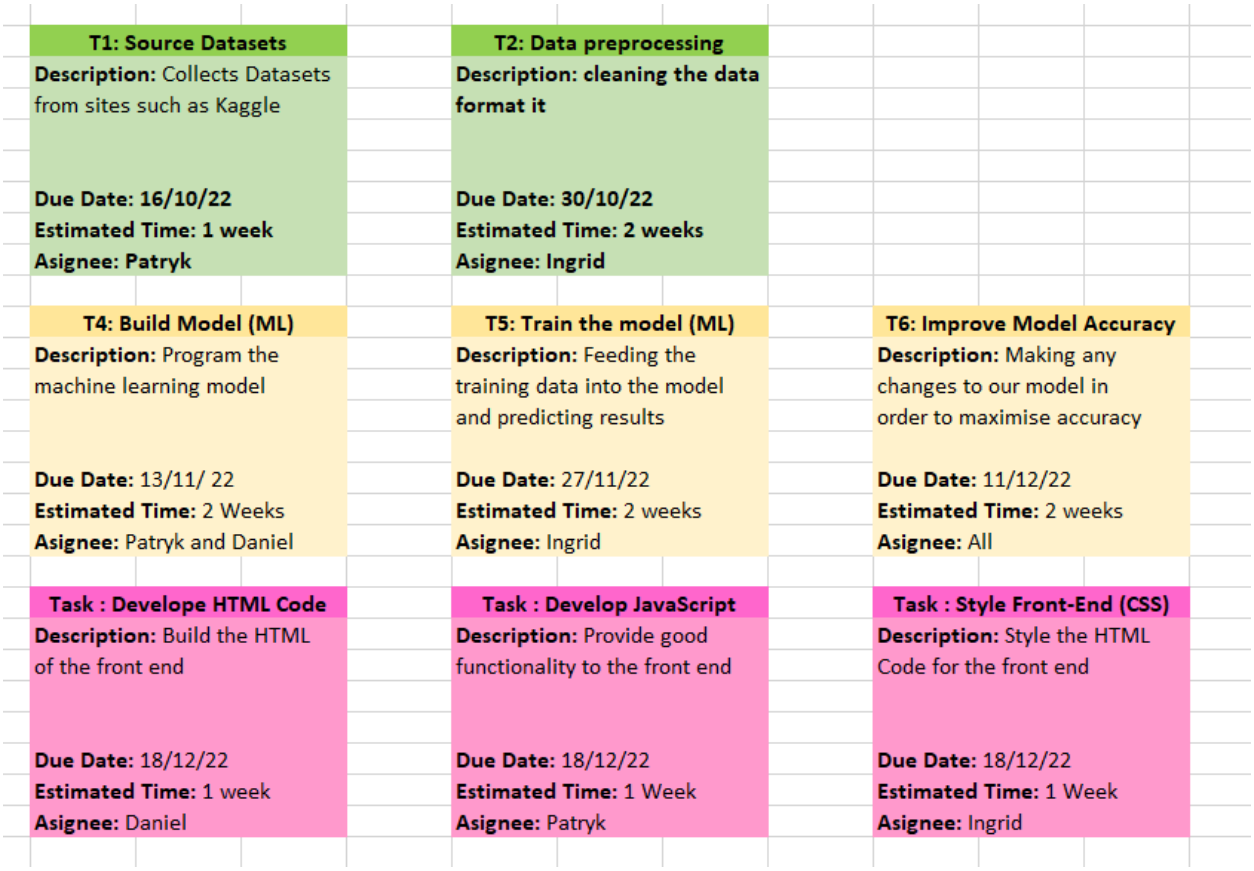

Figure 11: Screenshot of the project's Kanban cards. Each colour represents a different sprint. These will all eventually be put on to the Kanban board.

Figure 12 below shows a screenshot of our project's Gantt Chart. This chart will help to give the team a general overview of tasks completed and what tasks are not completed on a much clearer scale as you can see months in advance.

From the screenshot you can see that we have the tasks listed on the left side of the chart. The various dates for each task are displayed across the top. Next to each task and in conjunction to their respective completion dates are coloured bars that represent the estimated time delegated to each task. As you can see, they all line up with each other providing a very clear view of the progression of the project.

We as a team have decided to give ourselves until our last week in December to have the “Make it work” part of the project complete. During and after the holidays we have agreed to try and achieve the “make it better” part of the project. Then finally towards the end of our final deadline on the 10th of February, we as a team hope to have achieved the “Make it best” part of this project.

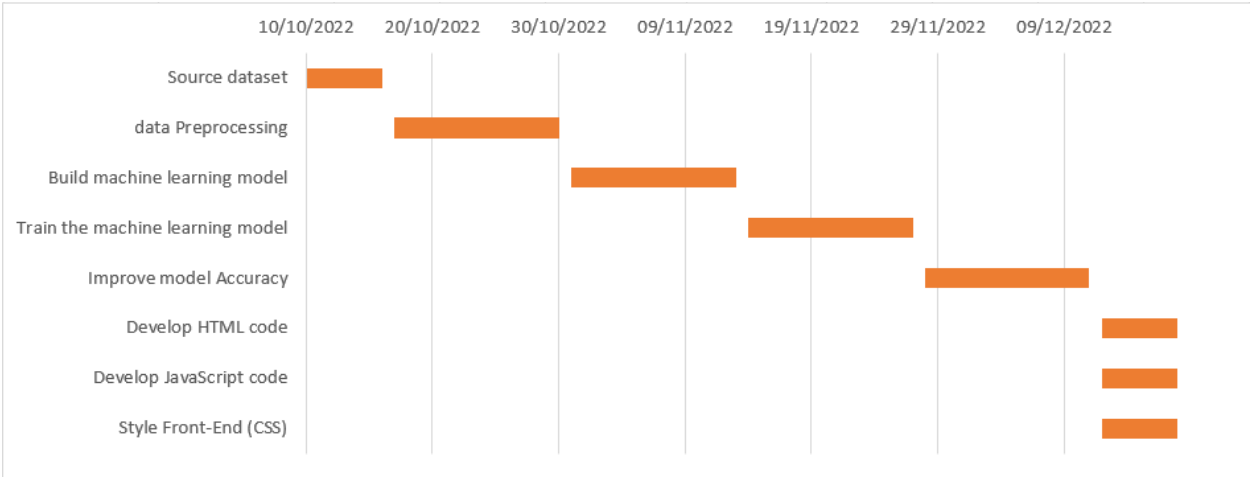

Figure 12: Screenshot of Gantt chart. Complete with task list and estimated time of completion.

4.3 BUDGET

Approximately €50 for Raspberry Pi 4 8GB RAM version.

Domain/cloud hosting ~€25 per month to host our web front-end.

Alternatively, we can use free hosting.

Our budget is not planned to exceed €100.

4.4 COMMUNICATION & COLLABORATION PLAN

Our communication plan is a simple process to follow, we have created a Discord group chat where we can quickly and easily share ideas with each other. We also intend to use this group chat to hold our daily scrum meetings. At the beginning of each day, we have agreed to meet for 15 minutes, and each member answers these three simple questions. 1) What have you done so far? 2) What are you currently working on? 3) Do you have any problems? Answering these questions daily we can all get a good sense of the progress of the project and help each other out if needs be.

We have also agreed to go on a call once to twice a week to discuss the week's progress and to help each other out with any tasks that may have an upcoming deadline and are not complete. We intend to use the class time designated to the group project for the same reasons.

Each member has also created a GitHub account. The team's project manager has created a repository on their GitHub and has given the other team members access to this repository. We are going to use the GitHub repository to share any code we create. This will allow our lecturers to see our progress as well as allowing each other to see our progress.

Our main method for keeping track of our project and making sure that we meet deadlines are using collaboration tools called Kanban and Gantt. Traditionally these tools are used separately, but for this project we have decided as a team to use both together. The Kanban board serves to help us visualize the status of the tasks that we have created to complete the project. The Board consists of three categories, “To Do”, “In Progress” and “Complete”. We place each task under the relevant category heading and move them accordingly. The Gantt chart provides a similar view of the project, but it allows us to see a clearer timeline of the project's deadlines. The Chart allows us to see months in advance with each task's deadlines clearly marked.

We have a process for delegating tasks among our group members. Once the entire team is all together, we go through any tasks that are completed, we discuss any tasks that need to be finished and we also discuss any tasks that may not be on our Kanban board or our Gantt Chart. During this discussion we talk through each task and then the project manager asks who would like to take that task. Sometimes a task can take more than one person. We follow this process for each task, thus delegating and assigning each task to a team member equally and fairly.

5. REFERENCES

https://www.unb.ca/cic/datasets/index.html

https://github.com/christianversloot/machine-learning-articles/blob/main/how-to-predict-new-samples-with-your-keras-model.md

https://www.researchgate.net/publication/350151991_Real-Time_Network_Intrusion_Prevention_System_Based_on_Hybrid_Machine_Learning

https://esource.dbs.ie/bitstream/handle/10788/4251/msc_chudasma_ph_2020.pdf?sequence=1&isAllowed=y

https://www.youtube.com/watch?v=czaGYvE1UJ4

https://github.com/gfek/Real-CyberSecurity-Datasets