GUI scan system with Machine Learning techniques and CICFlowmeter to analyse Intrusion Detection on Networks

Jawad Mahmud, 001174077

Computer Science, BSc Hons

University of Greenwich

Contact Information:

School of Computing and Mathematical Sciences
University of Greenwich
Old Royal Naval College, Park Row, SE10 9LS
Email: jm6992s@gre.ac.uk

Abstract

Network intrusion detection systems have always had a very crucial role in maintaining the security of the computer network environment against anomalies and other potential malicious behaviour. As a result, several research has been undertaken by many researchers that have tackled many issues in the field. However, they have been able to incorporate machine learning algorithms on various datasets that have received different results overtime. Furthermore, in this project, a GUI system would be implemented using Python and a variety of extensive libraries that would utilise machine learning algorithms and techniques, as well as utilising the CICIDS2017 dataset and CICFlowmeter, which would help display and analyse the data traffic in the network for potential anomalous behaviour.

Introduction

In many years of cyber security, one of the major challenges that impact the modern society is intrusion detection in networks. Across many modern technology such as laptops, phones and other devices, the impact that intrusion detection made has caused many inconveniences, and due to this making a stronger impact on the security of devices, it has been a major security drawback. This has caused many researchers to utilise machine learning techniques to help identify the attack patterns within the system, and assess what the input and the output is regarding the attack. [1] and [2] has mentioned that machine learning techniques such as classification algorithms have been utilised as being the prime examples of the techniques that have been applied for the detection of potential intrusions through the patterns. They have also mentioned that the techniques have had invaluable support for revealing potential anomalies, and they classify/recognise the data traffic through flow and packet analysis.

Project Idea & Objectives

The main idea is to be able to implement an alert system in the form of a GUI for network intrusion detection, with the main objective being that machine learning techniques would be integrated to help detect potential intrusions within the network through live scanning and using the methodology of data preprocessing and feature engineering to collect the traffic data and identifying the relevant column features. The main objectives of the project are the following:

1. When using the GUI, it would enable users to use CICFlowmeter to go through a live detection and view the CSV data for predicted intrusions based on model prediction.
2. When the behaviour is displayed, it would show standardisation scales, that determines the behaviour with standardisation values 0 and 1, indicating malicious and benign behaviour.
3. The machine learning classifiers and model should be able to help analyse and classify the network behaviour for potential intrusion anomalies

Academic Questions

1. **How can intrusion detection help contribute to security in the network infrastructure?**
2. **How can the machine learning algorithms and the model be beneficial for noticing the intrusions in the network environment?**
3. **How can the main approach for the GUI system determine whether the network behaviour is benign or malicious?**

Project Lifecycle

It is essential to consider the major analysis, as well as designing, implementing, testing and evaluating the system, as well as keeping the main methodology in consideration during implementation, as it would be an essential part of the success of the project.

Methodology

The project has utilised data preprocessing, imbalanced learning, feature engineering and model training [4], in which the CICIDS2017 dataset will be utilised. Firstly, data preprocessing has been used to provide a clean and flexible dataset that improves the machine learning model and make it reliable. This is important, as cleaning up the data with data cleaning would help reduce the inaccuracies on the network data traffic. The PCA, or Principal Component Analysis, algorithm had been used for reducing the dataset dimensionality and would be used for the next steps in the process. Furthermore, imbalanced learning uses a balance of positive and negative cases within the dataset to provide feature engineering methods such as feature selection, which is a selection of a subset in the original dataset that would help avoid the complexity of the data dimensionality, which was then followed by feature extraction, which uses the PCA algorithm to create the model, and use that model for the intrusion predictions.

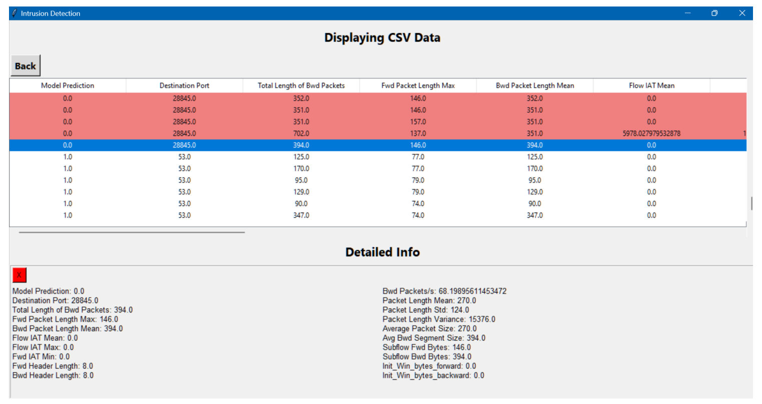
Moreover, it was proven essential that experiments must be conducted with various machine learning algorithms to get a better accuracy for the model, which was then integrated onto the interface. These steps are maintained to ensure that the system is able to determine intrusions efficiently and that the ML techniques make it accurate enough for the user to know that there is anomalous behaviour.

Prototype

The prototype is comprised of a GUI enables the application to use CICFlowmeter for a live detection and a 'use file' option that would display the selected network packet csv output to determine the behavioural patterns. The network intrusion detection GUI alert system was implemented in Python using libraries such as NumPy, Sklearn, Pandas, os and sys which is required for implementing the machine learning techniques, and Tkinter, which was required for implementing the user interface, and this is to ensure that the model has been integrated onto the main GUI to monitor the behaviour.

Results

The experiments conducted through the project implementation with the dataset include four algorithms and a main GUI that would help determine the network behaviour. Figure 1 shows the model prediction uses standardisation scaling that would help understand what the behaviour is, in this instance, if the model returns 0, it means that there is malicious activity present, whereas 1 has been able to indicate that benign behaviour has been detected. In addition to this, the data would be presented when the user selects a valid CSV file that would help determine what the behaviour is, and when a user selects the row, it would display the activity information and determine what behaviours have been identified, e.g. packet length, subflow of bytes, total length of packets and the destination port it came from.



Model Prediction	Destination Port	Total Length of Bwd Packets	Fwd Packet Length Max	Bwd Packet Length Mean	Flow IAT Mean
0.0	28845.0	352.0	146.0	352.0	0.0
0.0	28845.0	351.0	146.0	351.0	0.0
0.0	28845.0	351.0	157.0	351.0	0.0
0.0	28845.0	702.0	137.0	351.0	5978.027979532878
0.0	28845.0	394.0	146.0	394.0	0.0
1.0	53.0	125.0	77.0	125.0	0.0
1.0	53.0	170.0	77.0	170.0	0.0
1.0	53.0	95.0	79.0	95.0	0.0
1.0	53.0	129.0	79.0	129.0	0.0
1.0	53.0	90.0	74.0	90.0	0.0
1.0	53.0	347.0	74.0	347.0	0.0

Model Prediction: 0.0
Destination Port: 28845.0
Total Length of Bwd Packets: 394.0
Fwd Packet Length Max: 146.0
Bwd Packet Length Mean: 394.0
Flow IAT Mean: 0.0
Flow IAT Max: 0.0
Fwd IAT Min: 0.0
Fwd Header Length: 8.0
Bwd Header Length: 8.0

Bwd Packets/s: 68.19895611453472
Packet Length Mean: 270.0
Packet Length Std: 124.0
Packet Length Variance: 15376.0
Average Packet Size: 270.0
Avg Bwd Segment Size: 394.0
Subflow Fwd Bytes: 146.0
Subflow Bwd Bytes: 394.0
Init_Win_bytes_forward: 0.0
Init_Win_bytes_backward: 0.0

Figure 1: The system running

Four algorithms have been implemented and experimented on and these are the main results on how efficient they are:

Classifier	Accuracy	Precision	Recall
KNN	0.99	1.00	1.00
Logistic Regression	0.881388	0.89	0.88
Naive Bayes	0.760484	0.77	0.76
Decision Tree	0.99817	1.00	1.00

Table 1: Classifier results

The table for each of the algorithms shows that Decision Tree classifier runs the best as it provides a better average accuracy compared in contrast to the other algorithms. Furthermore, this has been achieved by testing the model with a test validation set as well as separating the dataset based on label values to test prediction.

Evaluation

1. The research of this project has proved the success in which CICFlowmeter can be applied to the GUI via live scanning.
2. The GUI implementation was straightforward with basic logic implemented to give an idea on how the next stages would be carried out, which would also help to integrate the main components such as the classifiers and model prediction.
3. The main challenges involves with installing CICFlowmeter and whilst it was successful, setting it up has been time consuming, and could only be used through terminal.
4. Moreover, during algorithm experimentation, it was crucial to be understand which of the machine learning algorithms are necessary for the main process, as it also depends on the model accuracy.

Conclusions

- The system is able to access and display the CSV data onto the interface through a table of columns with detailed information. When information is displayed, the activity has been labelled as malicious, with the red rows showing that the behaviour is malicious, and white rows have indicated that the behaviour is benign. This has been achieved through the use of model prediction.
- Despite some difficulties with CICFlowmeter for the GUI, it can be used the same way through a terminal and it can be able to carry out the necessary features.

Limitations & Future Work

It would have been useful if CICFlowmeter had been integrated for the live scan onto the interface. It is important that the CSV files that the user wants to display must have valid values based on the columns. In future, these findings by ensuring that many features can be integrated that would enhance the accuracy of machine learning techniques even further. Moreover, to ensure further use, it would be beneficial to consider the nature of the machine learning process, and recent datasets will gradually degrade the accuracy, precision and recall, with potential new threats that may cause challenges with the security overtime.

References

[1] Dylan Chou and Meng Jiang. A survey on data-driven network intrusion detection. *ACM Computing Surveys (CSUR)*, 54(9):1–36, 2021.

[2] Mario Di Mauro, Giovanni Galatro, Giancarlo Fortino, and Antonio Liotta. Supervised feature selection techniques in network intrusion detection: A critical review. *Engineering Applications of Artificial Intelligence*, 101:104216, 2021.

[3] Gints Engelen, Vera Rimmer, and Wouter Joosen. Troubleshooting an intrusion detection dataset: the cicids2017 case study. In *2021 IEEE Security and Privacy Workshops (SPW)*, pages 7–12. IEEE, 2021.

[4] Zhen Yang, Xiaodong Liu, Tong Li, Di Wu, Jinjiang Wang, Yunwei Zhao, and Han Han. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, 116:102675, 2022.