National College of Ireland

PREVENTING THE INBOUND MALWARE USING RECOMMENDATION ALGORITHM

MSc Research Project
Master of Science in Cyber Security

Kousic Bhadan Janarthanan
Student ID: x20177313

School of Computing
National College of Ireland

Supervisor: Dr Vanessa Ayala-Rivera

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: …….Kousic Bhadan Janarthanan……

Student ID: ……x20177313……..……

Programme: ……MSc Cyber Security…… **Year:** …2021-2022...

Module: ……MSc Research Project ……..……

Supervisor: ……Dr Vanessa Ayala-Rivera……..……

Submission Due Date: ……16/12/2021……..……

Project Title: ……Preventing the Inbound Malware Using Recommendation Algorithm

Word Count: ……5591…… **Page Count**……20……..……..

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

Signature: ……

Date: ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

Preventing the Inbound Malware Using Recommendation Algorithm

Kousic Bhadan Janarthanan
x20177313

Abstract

Inbound Malware refers to threats that occur as a result of inbound traffic entering your network. Although an anti-virus system can help to safeguard, it cannot always prevent the appearance of an unfamiliar signature. This is because numerous types of malware are specifically developed to infiltrate the insider network server. Due to the high number of traffic data, the difficulties encountered in screening these inbound bad actors are frequently hard to evaluate. Since incoming connections can originate from a range of origins, the best approach to guarantee that inbound limitations are not allowing access to an inside system or protecting inbound security rules from "known-bad" behavior or activity is to utilize a combination of the two. Thus, utilizing the similarity technique with the recommendation system, the behavior activity data can be tracked, which aids in keeping bad actors out.

The model was implemented with the help of a support vector machine and the k-nearest neighbours approach. Then, using a recommendation mechanism prioritizes the level of danger. The attribute-based recommendation system operates here. As a result, these inbound traffic analyses will be used to prevent Malware.

Keywords: Intrusion Prevention System, Inbound Malware, Support Vector Algorithm, KNN Algorithm

1 Introduction

Network resource sharing entails various dangers for both local and wide area networks. The potential hazards associated with data integrity and confidentiality have had major effects in numerous social areas, including health, finance, and education. According to a stealth security white paper, the Zeus malware has infected 3.6 million PCs in the United States alone. [1] Same like this several malware infections such as Nanocore, Dridex SpeakUp, GameOver, Emotet, and many more are posing significant hurdles to security teams, according to the world's largest security consultancy Symantec. As a result, Cybersecurity threats are on-going and increasing year after year.

To keep the scope of this essay limited, we would like to discuss the impact of malicious network packets on network security. For example, if a malicious inbound packet enters the system and is not detected by the Firewall system, it may result in data breaches. By utilizing techniques such as cryptography, the hacker can flexibly modify their malware data packet format in order to avoid detection by the firewall system. Currently, intrusion prevention and detection systems use signature and anomaly-based pattern analysis to examine poisoned data packets. As stated in the title, this paper used machine learning approaches to categorize incoming packets in order to identify Malware. [2]

Many related themes were uncovered when learning about this sector, demonstrating that the usage of machine learning algorithms to detect malware patterns is the current study trend. Unlike previous studies [8] [12], this one focused on exploring and developing pattern recognition approaches to detect malicious data packets based on incoming firewall analysis in order to detect Cybersecurity assaults early. [3] This is not an entirely new approach; it integrates previous research but has been improved [14].

2 Related Work

This section of the article covers the existing work in cyber security challenges and solutions, as well as malware pattern classification. Then detailed how the previous research has been undertaken in the domain of network intrusion detection systems. Also described possible methods to analysing the behaviour in any networking have also been proposed.

The principles of malware pattern recognition are recognized in the early phases of the study; afterwards, the transportation implementation and also the approaches that are included in their operation are recognized from the research. As part of this study, all of the difficulties and obstacles that are recognized during the operation are gathered, and the recommended approaches are offered in the methodologies.

The analysis of the issues that previous techniques encounter, which gave a way to overcome it in the strategy proposed as part of this work. Along with strategies and difficulties which are being faced, this section also provides an opportunity to understand how data analysis methodology was previously used and how standard or the findings that are accessible as parts of a literature study that was completed. Initially, the required metrics were not fully validated that were integrated in assessing the approach; nevertheless, these literature reviews assisted in conducting correctly, which resulted in the establishment of a performance metrics in this study.

2.1 Comparison of Different Malware analysis approach:

In the Cyber Security sector, accurate vulnerability assessment has become a serious problem. As a result, malware protection and detection approaches have evolved. The methodologies outlined in have progressed from signature-based, anomaly-based, and specification-based mitigation to sophisticated behaviour-based detection employing different

machine learning and software-based approaches. Following that, we'll look at the various methodologies used in prior studies.

Bowei et al. [7] proposed an approach about the adversaries that occur in machine learning, as well as a followed by a discussion about adversarial attacks and defensive measures, as well as a discussion about overall adversarial networks (GAN), as well as information more about knowledge, adversarial goals, and capabilities of attackers.

Soodeh et al. [8] proposed a novel machine learning approach for malware detection that combines the ANN, Logistic Regression, and Genetic Algorithm. The associated value sample was extracted as from datasets using Genetic algorithm and the Logistic Regression in the first step. Later Artificial Neural Network is then trained in second step using the GS and PSO algorithms to identify assaults and. The suggested model's efficiency was evaluated using two datasets: KDD cup'99 and NSL-KDD. Although the suggested model has a lower average accuracy, it identifies assaults faster than other ANN-based techniques.

Octavian et al. [9] constructed a framework termed "FAIL" model, that concentrates here on poisoning assault on four separate applications, each of which employs three machine learning techniques to defeat current protections. In this study, they created a stringRay that launches a poison assault on the four machine learning algorithms. Also, make suggestions for future defence.

Faezah et al. [10] decreased the data features by employing a wrapping approach based on the Differential evolution strategy for Intrusion Detection System. Its number of characteristics has been lowered since irrelevant features have an impact on IDS efficiency. The goal is to use differential evolution to choose several of the features from of the NSL-KDD datasets with Extreme Learning Machine to predict the efficiency of the supplied strategy. Differential Evolutionary is repeated until the smallest subset of features with excellent precision is obtained. The suggested model has 80.15 percent recognition accuracy for five classes as well as an 87.3 percent classification rate by reduction in testing and training time.

Matthew et al. [11] proposed an assault on the linear regression method and suggested attack and defensive tactics on such approaches. They picked three datasets: health dataset, load dataset and housing dataset, and they are targeting the linear regression model on all three datasets. The author explains how misinterpretation may result to vulnerabilities; for example, whether it's utilized in a stock market dataset or a medical dataset, it might cause complications.

Iram et al. [12] conducted a research study on pattern recognition classifiers based on Extra tree classifier, Multi-layer perceptron, logistic regression, decision tree, Naive bayes, Random forest, support vector machine, and K-nearest neighbour for such categorization of traffic analysis as anomaly and usual, and also the study's outcomes has been reviewed on four distinct subsets obtained from the NSL-KDD data source. These training sets were pre-

processed depending on major characteristics prior to training a model. The analysis shows using machine learning classifications generate better outcomes for Denial of Service attacks and worse outcomes for user to root assaults, and the model's accuracy level is 99 percent.

Yuyang et al [13] introduced CFS-BA, an accurate IDS based on theoretical selecting features, to minimize data dimensionality depending on feature correlations. Then, during recognition, an ensemble technique comprised of the Random Forest classification algorithm, Forest via penalizes Feature, and C4.5 algorithm was used. Subsequently, the estimated values of the base classifiers were pooled using a voting method to identify the assaults. Its efficiency was 99.8 percent with selection of 10 characteristics chosen from the NSL-KDD sample.

Philipp et al. [14] suggested the NIDS with a lower false alarm rate employing a light weight one-class support vector machine for research analysis. As comparison to prior work, the system was tested with such a harmful data acquired instead of a benign networking sample. Traditional network - based intrusion detection algorithms have been extensively researched and may be used to innocuous traffic analysis. Many anomaly detection methods must be deployed in large-scale networks to safeguard the OpenFlow network.

In this research, we apply machine learning classification techniques to discover anomalies. For identifying assaults, six essential characteristics are chosen: dst host, src bytes, rootshell, protocol type, duration, and dst bytes. So the fundamental distinction among our approach and previous articles is that we apply symmetric pre-processing and feature extraction in the framework of inbound firewall.

3 Research Methodology

In this study, KDD (Knowledge Discovery and Data Mining) is used. Knowledge discovery and data mining are currently important aspects of Data Science. To make any service better and more efficient, we require a lot of data. Data can be submitted in any raw condition or format, but it must be pre-processed according to the criteria before being analysed further. Following the processing of the data, several data mining techniques are used to identify any existing patterns in the data, which are then assessed and examined.

This study approach is divided into five stages, as illustrated in the picture below: data collection, data pre-processing, data transformation, data modelling and conversion, evaluation, and findings.

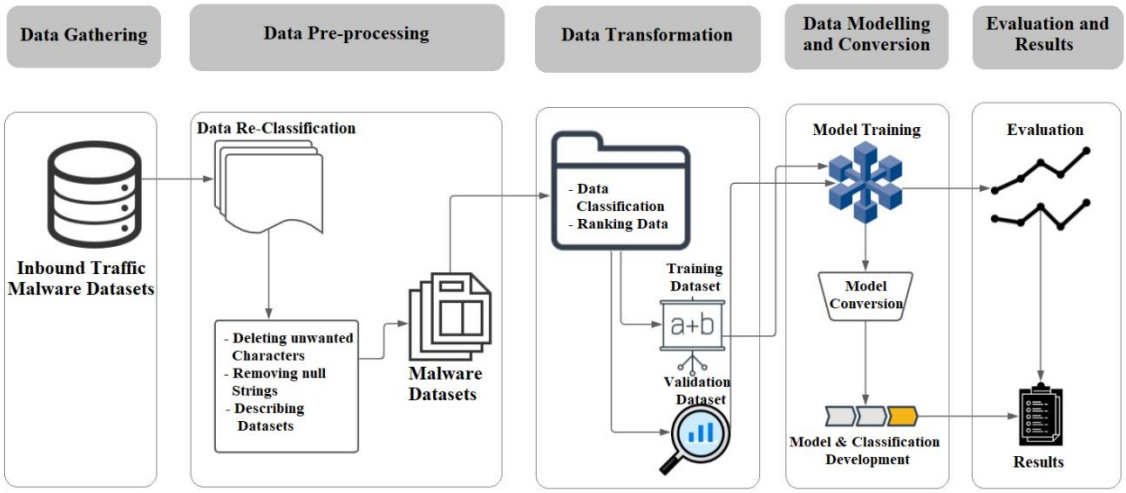

Figure 1: Methodology Proposal for Research

The next step, we performed Data pre-processing that includes data dividing, plot the Correlational study, remove the unnecessary characters, locate data imbalance, balancing data, and detailing data column. In each category, all malware data was manually checked for misclassified data. Data that had been misclassified was reclassified. Unnecessary detail, such as character and empty space, was removed from the data.

The third step includes data classification and ranking of the dataset, which is then divided into training and validation datasets. In order to train a model, data classification requires similarity analysis and prioritization of the dataset. There are five types of vulnerabilities in the malware analysis dataset, as well as one normal dataset. The classified dataset was then split in most for training and rest for validation in an 80:20 ratio.

Model training, model conversion, and model and classification development are all part of the Data Modeling and Conversion stage. The models were trained and tested using the splitted training dataset and validation dataset. Then, based on similarity, a classification development is proposed. To calculate the similarity between the attributes, all of the datasets were first imported into a distance formula. The k-nearest neighbors and support vector machine algorithm was then projected into the class, which made extracting potential malware pattern from the dataset more efficient. The potential malware pattern was predicted based on similarity on this basis.

Finally the stage evaluation and results reconstruct and provide the output result in a format of classification and prioritization. This prioritization enables to alert the vulnerable data.

4 Design Specification

Malware sample classification poses significant challenges in security design as well as monitoring. To identify the large number of malware variants, a malware classification system that is able to support a huge samples and able to adapt to predict variations at running time is required.

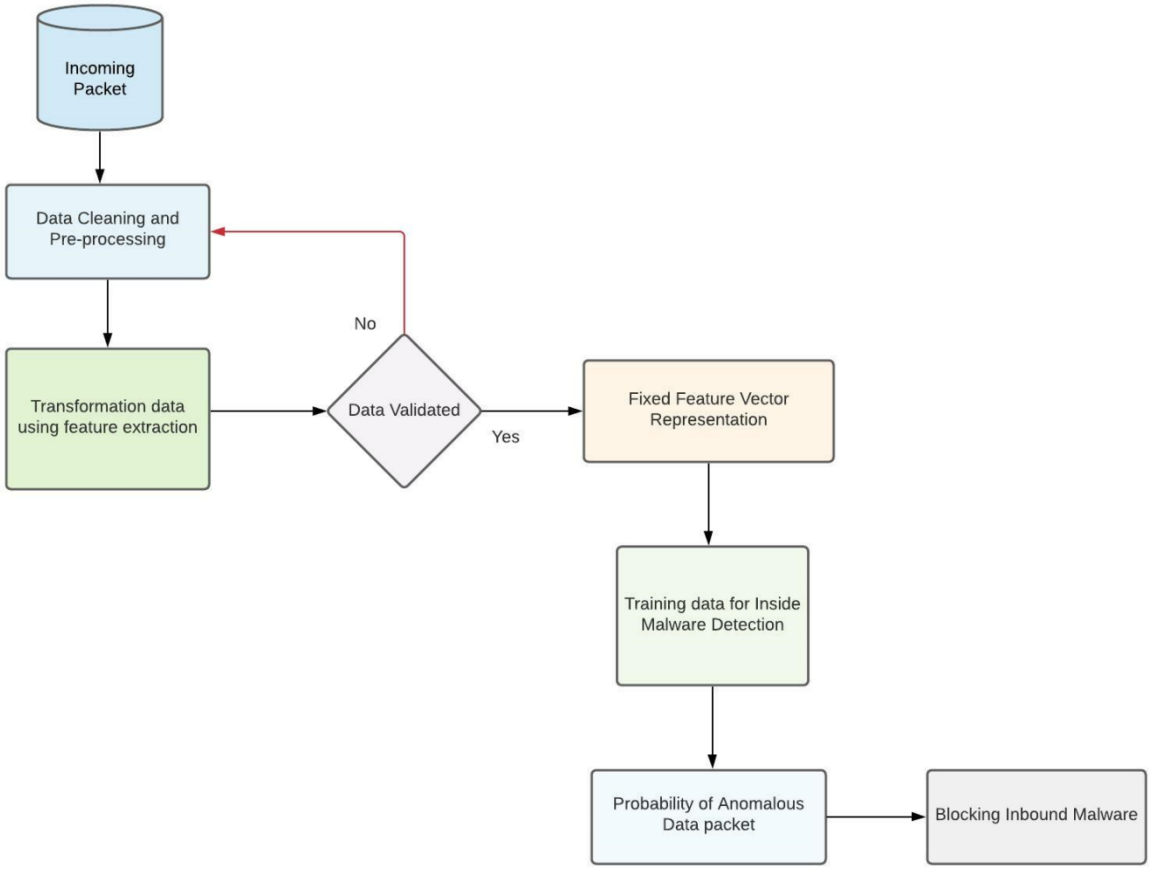

Figure 2: Process flow of preventing malware using machine learning

Our data sample contains 19 quantitative and qualitative research features related to incoming TCP packets observed during network connections. In this section, we present a flow process for creating a feature vector utilizing run-time behaviours and classification machine learning algorithms for analysing malware samples in such a scalable and distributed infrastructure.

This architectural flow process is used to construct a machine learning idea that necessitates the utilization of several techniques for data analysis. The data collecting method may be flow-based or packet-based, and it will be dependent on incoming network traffic from the kaggle public data pool. As a result, for a clearer understanding, the data must first be arranged in a histogram. Then, before doing data validation, data transformation is required. However, in some cases, data imbalance will be detected, resulting in incorrect output results due to the data overfitting. Thus, a co-relation co-efficient diagram between the

data and the result is generated using a feature extraction approach such as a co-relation matrix. Using the co-efficient view may result in the removal of some characteristics in order to avoid data overfitting. Once the data sample is completely fit for analysis, it may be used to train the malware detection algorithm. In this manner, the analysis of anomalous data packets will be performed in real-time incoming traffic. This model is used to create an inbound firewall threshold rule. As a result, every malware pattern matching data packet received from the internet will be blocked before it enters the system.

4.1 TensorFlow Framework

A machine learning framework is needed to execute the proposed approach. Several machine learning frameworks are accessible for development, despite the fact that TensorFlow, an open source machine learning platform, allows execution of code directly through a browser window utilizing Google Collaboratory. Tensor frameworks enable the use of an open source core package to design, test, and train machine learning analyses on data samples. Furthermore, high-level API may be implemented over TensorFlow to determine the possibility of the training. The proposed model is developed using some of the most popular Python packages in this study. Multiple levels, such as the Dense layer, Embedding layer, and Convolutional layer, are linked in such a sequence to possibly handle Tensors.

5 Implementation

This section describes the whole implementation of the designed solution. Further information on the technology utilized and the building of an environment is provided.

5.1 Overview

The term "state-of-the-art" in this case refers to a snapshot of the development stage and methodologies employed in the field of modelling. The goal of the intrusion prevention system's development is to automate the identification of malware signatures using packet headers and characteristics. To do so, a classification machine learning technique is first utilized to locate the malware signature in order to highlight it. If the signature is anticipated, the risk prediction and prioritization are simplified using the recommendation technique. Each level of the development process necessitates the use of computer vision and advanced data analysis techniques. So, using these methodologies as a foundation, they are expanded to the classification algorithm, which needed less manual intervention in the module completion process. Aside from these categorization methodologies, the goal of this research is to look into each incoming datagram. As a result, this study provides data security for end users through this filtration.

5.2 Proposed Approach

The strategy employed in the research design is by using a machine learning algorithm that is designed based on a classification algorithm with a pattern recognition technique.

These classifications are built using the K-nearest neighbour technique and the Support vector machine algorithm. To increase the accuracy of the findings, two separate categorization algorithms are built. Furthermore, the analysis of two algorithms is utilized not only to offer accuracy on results, but also to compare both of these techniques with empirical analysis, and the best model among the two is presented to forecast the malware pattern. So this model can be used as firewall boundaries to prevent from the attack. This implementation is considered as a preventative measure, and it may be used to break the delivery step of the Cyber kill chain by preventing malware transmission into the system.

5.3 Data Gathering

The dataset for this is available at kaggle. [19] The dataset can anticipate five sorts of assaults. As a result, there were two varieties of malware. The information was obtained from an inside organization private research group for the purpose of evaluating an ecommerce website. They conducted a variety of attacks in order to obtain statistical information regarding their internal server firewall system. The dataset included 60938 test results from different attacks and two malware families. The number of outcomes and their details are shown below.

Attack Sample	Record count
Normal record	60593
IPsweep	306
Buffer Overflow	22
Rootkit	13
Worm	2
SQL attack	2

Table 1: Record count of Samples

5.4 Data loading

The experiment that has been conducted out with part of the present work is done using the Collaboratory of Google that also hosts the resources which enhance the further conversion of the samples that have to be estimated and the malware pattern that needs to be predicted once the dataset is collected from Kaggle. Initially, the pandas library is required to read the comma-separated values file. The Pandas library is a highly handy resource for separating values in a dataset file using a comma. Each line in the file represents a sample record. Each sample has one or so more fields with commas separating them. Thus, the data structure and two-dimensional labelled data frames of the data, columns, and index can be visualized using the pandas data analysis library. The pandas object is most commonly used

to describe a spread sheet for data analysis. The second element of this section is data pre-processing, which comes after the formulated data frames.

```
[ ]  df
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	udp	private	SF	105	146	1	1	1.0	0.0	0.0	255	254	1.00	0.01	0.00	0.0	0.0	normal.
1	0	udp	private	SF	105	146	1	1	1.0	0.0	0.0	255	254	1.00	0.01	0.00	0.0	0.0	normal.
2	0	udp	private	SF	105	146	1	1	1.0	0.0	0.0	255	254	1.00	0.01	0.00	0.0	0.0	normal.
3	0	udp	domain_u	SF	29	0	2	1	0.5	1.0	0.0	10	3	0.30	0.30	0.30	0.0	0.0	normal.
4	0	udp	private	SF	105	146	1	1	1.0	0.0	0.0	255	253	0.99	0.01	0.00	0.0	0.0	normal.

Figure 3: Viewing dataset using pandas

5.5 Data Pre-Processing

Understanding the available dataset is the very first step in the data pre-processing. To see how the distributions of the attribute function is accessible throughout the dataset gathered, the attribute head is studied and the values are shown from the dataset obtained. When this is noticed, one may be certain that individuals understand what variables are disseminated and how they can be managed in the future. Once these have been displayed and the distribution has been established, the next step will be to resolve the invalid and incomplete data. Along with it all, the following stage in the pre-processing procedure is to resample the records that were acquired as portion of the imbalance in perspective.

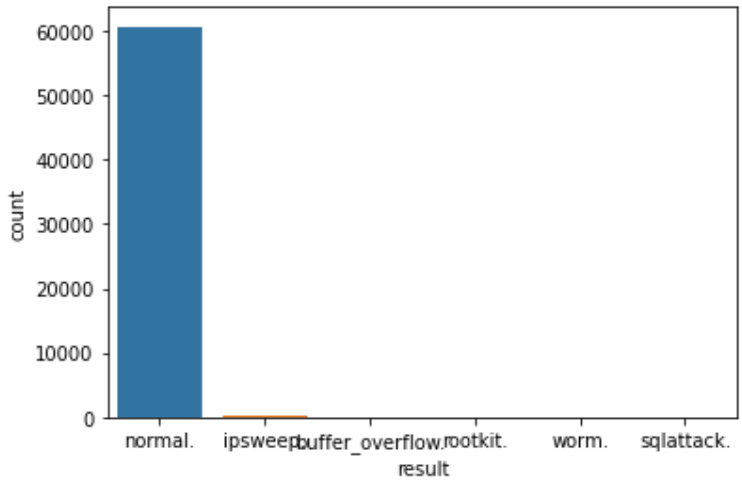

Figure 4: Visualize the imbalance in dataset

Some uniform balancing technique is required in this model to resolve the dataset imbalancing. Following that, these datasets, as well as the dataset's available original data, must be processed and prepared for model validation utilizing the created test and train data. Before transforming the testing and train datasets, all data types must be integers for the other challenge to be handled, such as resampling. The sections that follow describe how this problem was addressed and what functions were necessary for the method.

5.6 Feature Selection:

The data imbalance issue discovered in the dataset was discovered by visual inspection. Imbalance large datasets are one special case of classification problems in which the class distribution is not constant across classes. Normally, there are two classes: the minority class and the majority class. This malware collection has a large number of normal samples. Thus, one resamples strategy for addressing data class imbalance problem is to randomly resample the whole dataset. Oversampling is a popular method for addressing with this situation. Random Oversampling is the process of randomly picking instances from minority class, with replacements, then adding them to that same training sample.

Figure 5: Visualize the Oversample result

The dataset was then manually processed, and a Correlation Coefficient analysis was utilized to establish the link between various variables and to identify the most important ones. Following the discovery of the key characteristics, a portion of the sample is produced based on such features, which is then utilized to create clusters for classification using KNN and SVM. [15]

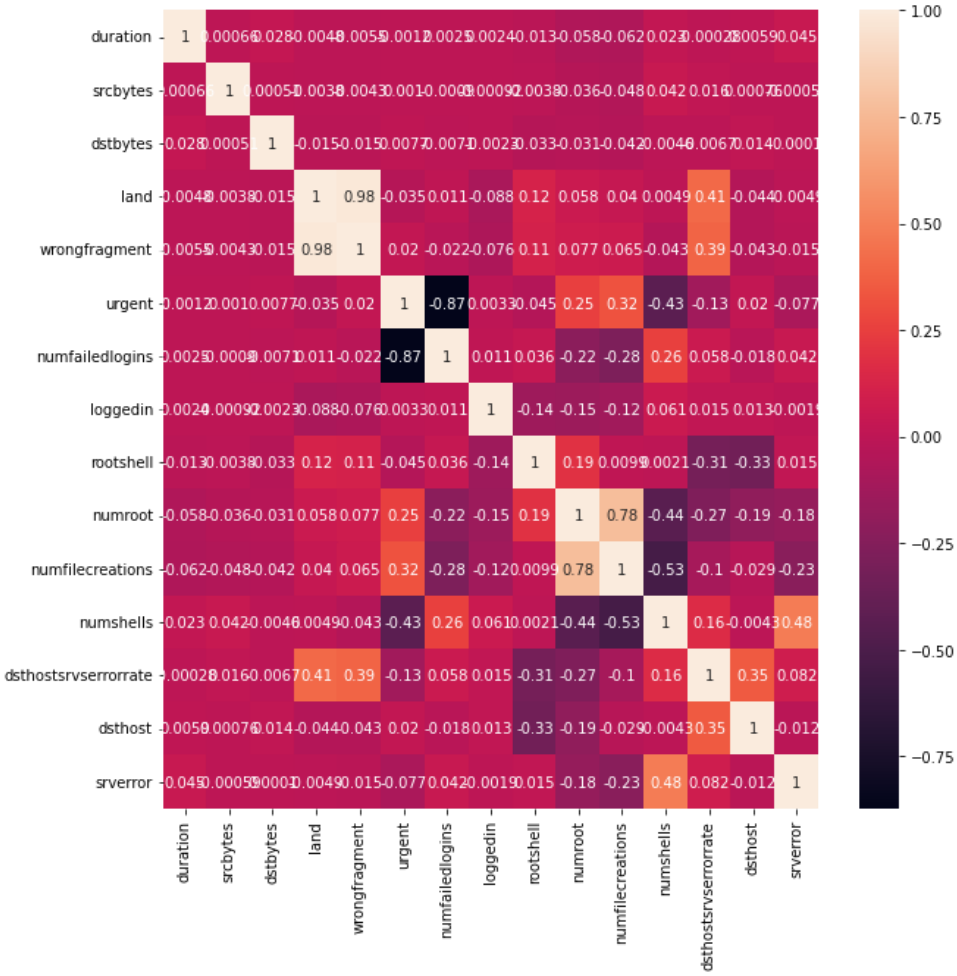

Figure 6: Correlational Test Plot

A correlational testing was carried using Seaborn statistical data visualization to establish the relationship between the different characteristics and also how one feature would enhance the other, and the results are given in Figure 6.

Since the variables included as part of the dataset have already been treated, the pre-processing performed as followed by a second aim of the task consists of exploration and data visualization. The next section discusses the model creation required as part of the suggested strategy.

6 Evaluation

In this section, we addressed the overall analytic result and the findings of the research in a statistical representation. To represent the plotting region, we utilized a Python library and the Tensorflow framework. Then, a study of several authors' research papers is utilized to describe the various ways of evaluating measurements. Applying each of these data as a benchmark, the assessment of this estimation method will be predicated on the criteria listed

below. But, in order to compute and comprehend where each of these statistics are built on, we must first establish the data from which they are generated. The preceding serve as the foundation for establishing every statistic required during machine learning techniques. [16]

i) Accuracy - The accuracy is used to assess the entire analysis of proposed approach. One of the classification module's assessment metrics is this. The accuracy rate is the ratio of accurate predictions compared to an overall ratio of input datasets.

ii) F1 Score - This F1 score is just the harmonized average value of recall and precision in the following equation, which takes both measurements into consideration.

F1 = 2 * (precision * recall / precision + recall)

iii) Recall - The model's ability to find all relevant instances within the data set. According to research, recall is a mathematical study of true positive values divided by the entire value of true positive values + the rate of false negative.

iv) Precision - Similarly, precision is a classification model's able to spot relevant data samples. Precision is also demonstrated numerically by dividing the value of true positive by its value of true positive + the value of false positive.

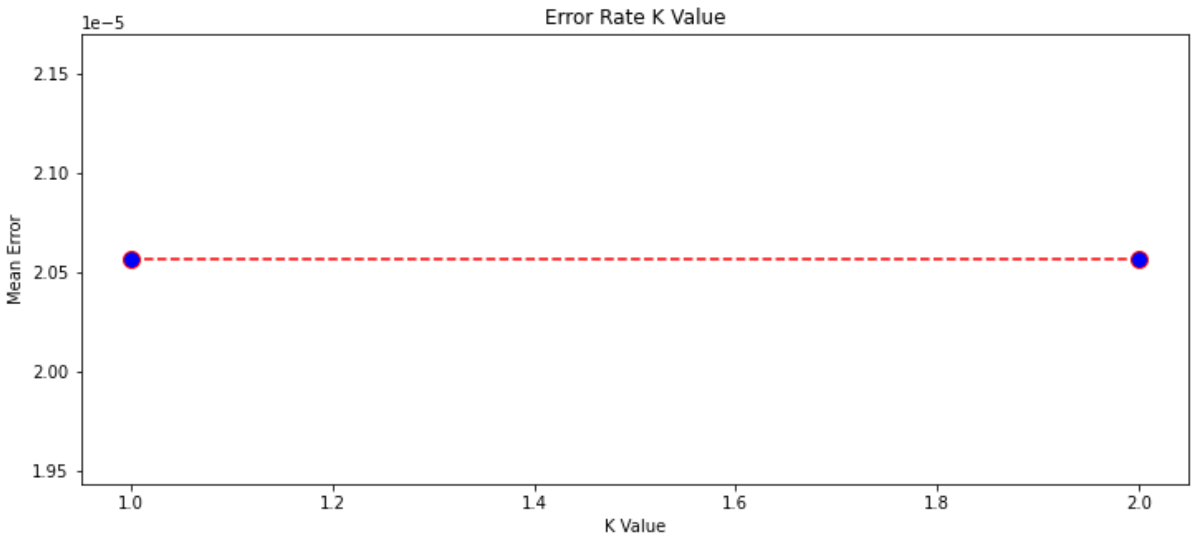

Figure 7: Comparing Error rate with K Value

The above error rate diagram shows the mean error for the predicted number of testing samples for all K values between 1 and 3. As evidenced by this, the data samples clearly overfit the borders. Because the training model's error rate of K = 1 should always begin at zero. However, the error rate here begins at K=2.6, so it may be considered data over fit. [17]

Our K-Nearest Neighbours study seemed to be able to recognize all sample records in the test model with 100% accuracy, indicating an exceptional outcome. However mentioned common metrics of F1 score, Recall, Precision, and Accuracy are sometimes insufficient since they do not provide a whole picture of the analysis behaviour. Therefore, combining error correction metrics and SNS plotting graphical diagram to provide a more informative report on the selected dataset.

```
[[48673     3     3     0     0     0]
 [    0 48665     0     0     0     0]
 [    0     0 48555     0     0     0]
 [    0     0     0 48647     0     0]
 [    0     0     0     0 48583     0]
 [    0     0     0     0     0 48611]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     48679
           1       1.00      1.00      1.00     48665
           2       1.00      1.00      1.00     48555
           3       1.00      1.00      1.00     48647
           4       1.00      1.00      1.00     48583
           5       1.00      1.00      1.00     48611

    accuracy                           1.00    291740
   macro avg       1.00      1.00      1.00    291740
        weighted avg       1.00      1.00      1.00    291740
```

Figure 8: Classification Model Evaluation

This SNS line graphing nicely visualizes the data that we have chosen, as illustrated in the picture above. For producing statistical graphics charts, we utilized numshells and numfilecreations data frames. The Seaborn plotting representation of exploratory data analysis demonstrated clearly that the data samples were overfitting. Therefore we used random over sampling technique to avoid data imbalance.

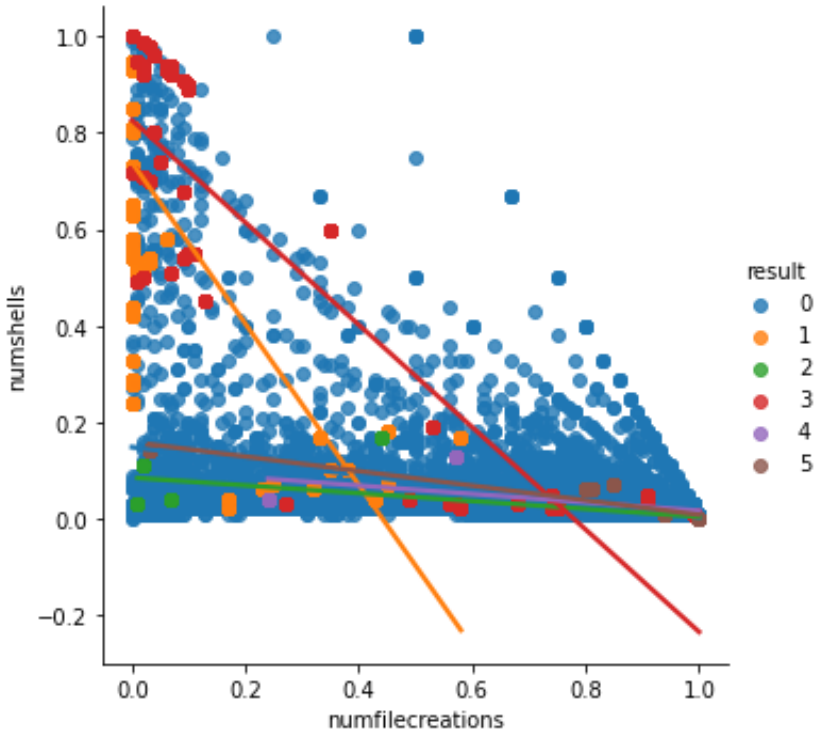

Figure 9: Seaborn Statistical Graphics Line Plot

The below Table 3, that displays the efficiency of a trained machine learning model. The rows indicate the estimated class, while the columns indicate the class's true figures. It really indicates whether or not two classes have been labelled wrongly (confusion). [18]

Predicted	0	1	2	3	4	5	All
True							
0	48634	19	24	2	0	0	48679
1	0	48665	0	0	0	0	48665
2	0	0	48555	0	0	0	48555
3	0	0	0	48647	0	0	48647
4	0	0	0	0	48583	0	48583
5	0	0	0	0	0	48611	48611
All	48634	48684	48579	48649	48583	48611	291740

Table 3: Confusion Matrix output

Because all successful predictions were situated inside the tables diagonally, it's simple to process of revealing the record for prediction mistakes, which will be indicated by numbers outside of the table.

Algorithm applied	Row count	Obtained Accuracy
K Nearest Neighbour	291740	0.999
Support Vector Machine	291740	0.999

Table 2: Accuracy representation for the utilized ML algorithm

The accuracy results of both K Nearest Neighbour and Support Vector Machine studies are presented in the table above. It is apparent that the study results in both algorithms outcomes are within the same range of results. Due to the size of the datasets applied in this research, a sample of 60938 is selected for implementation and efficient performance. The number of samples is raised by 364674 after setting random over sampling.

6.1 Experiment / Case Study 1

In this case, an assumption based on the organization data security is considered. Assume an employee in the organization is surfing on the workplace computer. The employee wishes to obtain a word document from the website. However, the document is infected with a rootkit file, which the user is unaware of. The rootkit-affected document is capable of gaining unauthorized access to a machine via user mode. They are commonly known to as program rootkits since they alter the executable files of popular software such as Office, Paint, PowerPoint, or WordPad. As a result, every time users launch the infected app's.exe files, the attackers get access to a computer system while you continue using the software regularly. The inbound firewalls are hardened, and a traffic capture is performed on the document file that the employee wants to download. In this experiment, rootkit packets are recognized by setting threshold values for srcbytes, service, and numroot.

```
df['result'] = 'normal'
df.loc[(df['srcbytes'] > 283600) & (df['service'] == 'ftp_data') & (df['numroot'] == 33), 'state'] = 'rootkit'
```

fragment	urgent	numfailedlogins	loggedin	rootshell	numroot	numfilecreations	numshells	dsthostsrvserrorrate	dsthost	srverror	result	state
1	1.0	0.0	0.0	20	33	0.5	0.15	0.5	0.06	0.0	normal	rootkit

Figure 10: Experiment output

6.2 Discussion

The malware was successfully identified in above test and discussion. This recognition of malware Pcap is possible in a live environment. This can be implemented in the incoming firewall system as well. As a result of this research relying on attacks in an organization, it was demonstrated that detecting malware from incoming traffic can be done very quickly using a machine learning technique. That same malware analysis investigation could be

reused that once theory is transformed to actual valid script. To discover from training set, this recommendation learning involves a few iterations. The following are some of my research's constraints.

I. The labelling of data in this research is conducted depending on limitation; this may be a time-consuming operation because it must be redone for each batch of training set.

II. Any firewall system obtained by the program for a specific conditions or event will be restricted to it specific predicted condition.

III. The machine learning analysis approach described above needs massive real-time data samples of training set.

IV. There is a possibility that the data will be prejudiced toward that particular outcome, and reduction may also be required.

7 Conclusion and Future Work

In this study, network packet records were employed to seek for internet-level malware predicated on packet headers such as time, protocol, service, and srcbytes. The study focused on detecting malware in a short time period and using recommendation-based conditions to adapt malware patterns. To optimize the productivity of classification method, traditional methods (K closest neighbours and Support vector machine) were built, and recommendation approaches utilizing those methods were used. The recommendation approaches used give a more proper and effective method for malware investigation.

Future studies might include automated malware pattern prediction and risk prioritization relies on manual inputs. If it is successful, the whole malware detection process should be based on previously acquired knowledge regarding cyber-attacks, with each recommendation serving as a training dataset for computer. Algorithms such as support vector machine and K nearest neighbour can be effective ways of recommending conditions.

Reference list

[1] Zeus Malware: Threat Banking Industry. (2010). [online] Available at: https://botnetlegalnotice.com/citadel/files/Guerrino_Decl_Ex1.pdf.

[2] Wang, J.-G., Neskovic and Cooper (2005). An adaptive nearest neighbour algorithm for classification. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/1527469 [Accessed 8 Dec. 2021].

[3] Ahanger, A.S., Khan, S.M. and Masoodi, F. (2021). An Effective Intrusion Detection System using Supervised Machine Learning Techniques. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/9418291.

[4] Ivey, E. (2021). 26 Cloud Computing Statistics, Facts & Trends for 2021. [online] Cloudwards. Available at: https://www.cloudwards.net/cloud-computing-statistics/.

[5] Morgan, N. (n.d.). Cloud cyber attacks: The latest cloud computing security issues. [online] www.triskelelabs.com. Available at: https://www.triskelelabs.com/blog/cloud-cyber-attacks-the-latest-cloud-computing-security-issues [Accessed 8 Dec. 2021].

[6] Straub, J. (2020). Modeling Attack, Defense and Threat Trees and the Cyber Kill Chain, ATT amp;CK and STRIDE Frameworks as Blackboard Architecture Networks. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/abstract/document/9265953/ [Accessed 16 Aug. 2021].

[7] Xi, B. (2021). Adversarial Machine Learning for Cybersecurity and Computer Vision: Current Developments and Challenges. [online] Researchgate. Available at: https://www.researchgate.net/publication/353067218_Adversarial_Machine_Learning_for_Cybersecurity_and_Computer_Vision_Current_Developments_and_Challenges.

[8] Soodeh, H. (2020). A new machine learning method consisting of GA-LR - ProQuest. [online] www.proquest.com. Available at: https://www.proquest.com/docview/2412657628 [Accessed 8 Dec. 2021].

[9] Suciu, O., Marginean, R., Kaya, Y., Iii, H.D. and Dumitras, T. (2018). When Does Machine Learning {FAIL}? Generalized Transferability for Evasion and Poisoning Attacks. [online] www.usenix.org. Available at: https://www.usenix.org/conference/usenixsecurity18/presentation/suciu [Accessed 8 Dec. 2021].

[10] Hamad, F. (2020). Differential Evolution Wrapper Feature Selection for Intrusion Detection System. Procedia Computer Science, [online] 167, pp.1230–1239. Available at: https://www.sciencedirect.com/science/article/pii/S1877050920309054.

[11] Jagielski, M. (2018). Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning - IEEE Conference Publication. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8418594.

[12] Abrar, I., Ayub, Z., Masoodi, F. and Bamhdi, A.M. (2020). A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset. 2020 International Conference on Smart Electronics and Communication (ICOSEC).

[13] Zhou, Y., Cheng, G., Jiang, S. and Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks, [online] 174, p.107247. Available at: https://www.sciencedirect.com/science/article/pii/S1389128619314203 [Accessed 12 Jul. 2020].

[14] Winter, P., Hermann, E. and Zeilinger, M. (2011). Inductive Intrusion Detection in Flow-Based Network Data Using One-Class Support Vector Machines. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/5720582 [Accessed 8 Dec. 2021].

[15] Raheel Shaikh (2018). Feature Selection Techniques in Machine Learning with Python. [online] Medium. Available at: https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e.

[16] Axman, D. and Yacouby, R. (2020). Probabilistic extension of precision, recall, and F1 score for more thorough evaluation of classification models. [online] Amazon Science. Available at: https://www.amazon.science/publications/probabilistic-extension-of-precision-recall-and-f1-score-for-more-thorough-evaluation-of-classification-models [Accessed 8 Dec. 2021].

[7] Sawla, S. (2018). K-Nearest Neighbors. [online] Medium. Available at: https://medium.com/@srishtisawla/k-nearest-neighbors-f77f6ee6b7f5 [Accessed 8 Dec. 2021].

[18] Jason Brownlee (2020). How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/.

[19] Jabir, P. (n.d.). Comprehensive malware datasets. [online] kaggle.com. Available at: https://www.kaggle.com/paytonjabir/comprehensive-malware-datasets [Accessed 16 Dec. 2021].