Machine Learning

Adam Board
2005335

1 Introduction

Following the implementation of a network monitoring system to uncover suspicious traffic, Scottish Glen must devise an effective method of analysing the network traffic by the type of attack present. The attack types can be categorised into 10 types: Normal, Fuzzers, Analysis, Backdoors, Dos, Exploits, Generic, Reconnaissance, Shellcode, and Worms. A classifier can be designed using different machine learning algorithms to analyse Scottish Glen's network traffic.

2 Machine Learning

There are a multitude of machine learning algorithms which can be used for classifying network packets. This paper will focus on the following three machine learning algorithms: Logistic Regression, Decision Trees, Random Forest Algorithms.

2.1 Logistic Regression

Logistic regression, commonly known as the Logit model, is a multivariable method of showcasing the relationship between various independent variables and a categorical dependent variable (Park, 2013). Predicting the categorical outcome when there are two or more predictors that may or may not be causes of that outcome is done effectively by utilising logistic regression. By displaying the results of the algorithm on a scale between 0 and 1, a logistical graph can be created to predict the likelihood of an outcome occurring. As seen in Figure 1, the graph typically becomes an "S" shape.

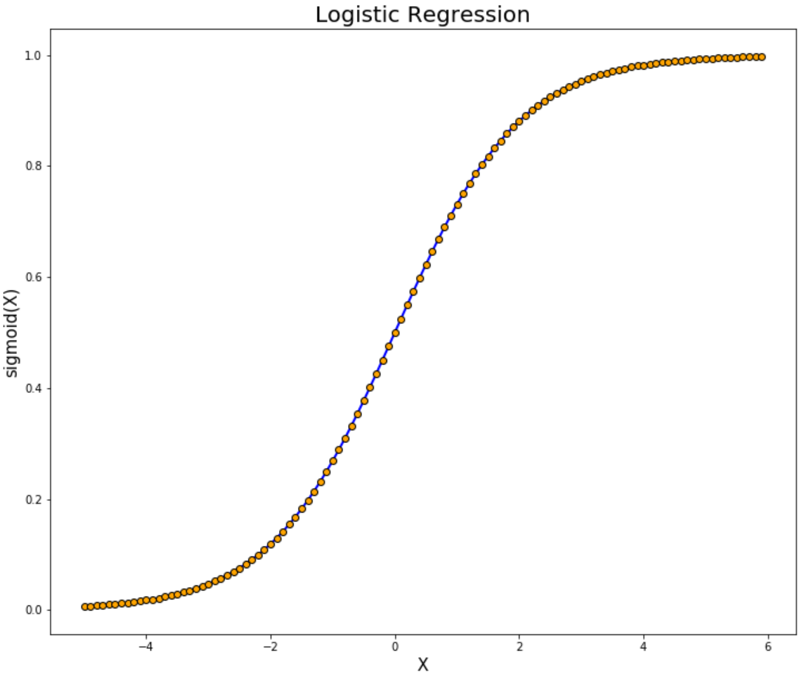


Figure 1 Logistical Curve Graph.

Logistical regression has been applied to many real-world cases. Some cases include predicting lightning strikes at Kennedy Space Centre, self-medication management in older adults, and environmental conservation attitudes in Nepal (Menard, 2010). A case more related to cyber-security is identifying the legitimacy of an email. The algorithm has been used to identify the legitimacy of an email by classifying legitimate emails with a 0 and phishing emails with a 1 on the chart. Using a logical function on the chart can predict whether an email is legitimate or not. However, while logistical regression is suitable for many use cases, it is not suitable for analysing network traffic as it has various attack categories to factor in. Logistic regression is usually suited for binary classification with use of multiple variables and contains limitations in being adapted to work with multiple categorical variables. Therefore, a better suited algorithm should be used.

2.2 Decision Tree

Decision trees are one of the most popular and simplistic learning models and is usually represented in a flowchart-like structure. Decision trees are a sequential model, which logically combine a sequence of simple tests to break down a larger complex decision; each test compares a numeric attribute against a threshold value or a nominal attribute against a set of possible values (Kotsiantis, 2013).

As demonstrated in Figure 2, The decision tree consists of nodes which resemble a rooted tree, meaning it is a directed tree with a node called "root" that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is called an internal or test node. All other nodes are called leaves (also known as terminal or decision nodes) (Rokach & Maimon, 2005). Each node is a simplistic decision that continues sequentially until it reaches the leaf nodes. The leaf node is assigned a value that represents the combination of all previously made decisions.

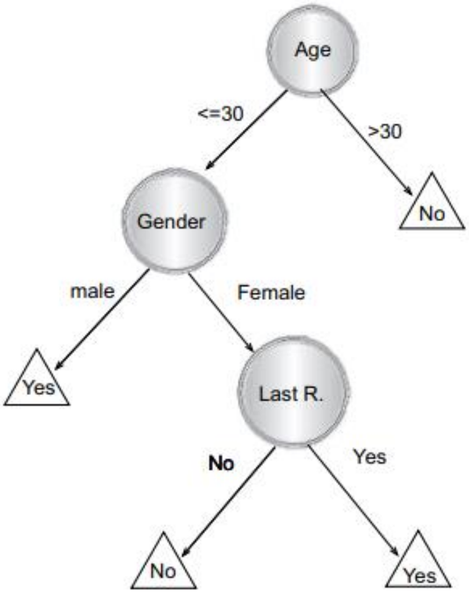


Figure 2 Structure of a decision tree (Rokach & Maimon, 2005).

Whilst decision trees are a popular option as they are highly adaptable and could be used for classification of network packet data, there are better-suited algorithms for accomplishing this task efficiently. To ensure that all possible classification outcomes are considered, a complex and large tree would be required. This would lead to a decrease in efficiency and increases the chance of network packet data being misclassified due to confusion within the algorithm during the long search time. A different implementation of this algorithm would mitigate the mentioned issues.

2.3 Random Forest

Random forest is a popular machine learning algorithm that was trademarked by Leo Breiman and Adele Cutler. Random forest combines the output of multiple decision trees to reach a single result (IBM, 2023). It's popularity stems from its ease of use and flexibility with handling both classification and regression problems. Additionally, the algorithm is fast at learning while being simplistic in design. A common use for random forest is detecting customers in banking that are more likely to repay their debt on time as well as detecting fraud (Meltzer, 2023).

As shown in Figure 3, the algorithm is constructed of multiple decision trees that can repeat and combine to lead to the same result while taking different paths through different decision trees. Using this approach improves the accuracy of the output and increases the efficiency in calculating the output.

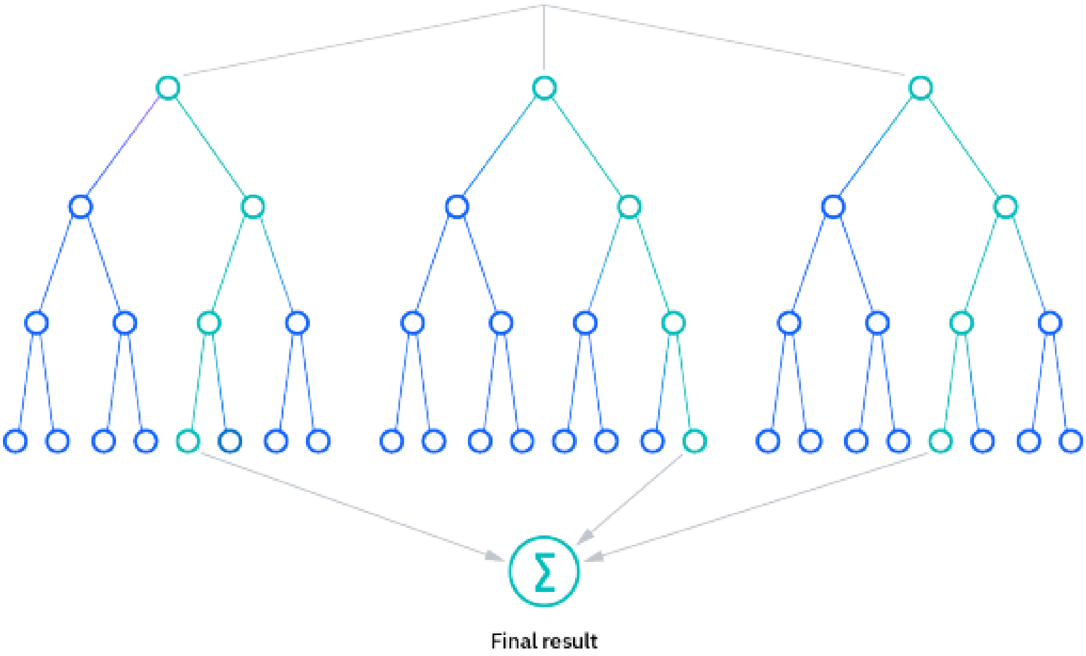


Figure 3 Random Forest Structure (IBM, 2023).

With the algorithm's ability to create multiple diverse trees, with the ability to obtain repeated results with improved accuracy, the random forest algorithm is the best solution for classifying the network packet data.

2.4 Summary of Strengths and Weaknesses

To conclude the research conducted on the various algorithms, the random forest algorithm was chosen as the appropriate algorithm. While decision trees and Logistic regression would be a valid solution to the current problem, random forest can handle large datasets with greater accuracy and efficiency when compared to the other two previously mentioned algorithms. The outcome from a decision tree would be large and complex, which would not meet the requirements of the organisation. Logistic regression is typically suited for binary classification which is not efficient for analysing large datasets with multiple factors to be considered.

3 Classifier Design

As mentioned previously, after reviewing the strengths and weaknesses of various machine learning algorithms, the decision was made to use the random forest algorithm. Several phases are required to build an appropriate model in machine learning. The following must be considered when building a classifier:

- Data Ingestion/Preprocessing
- Modelling
- Analysis

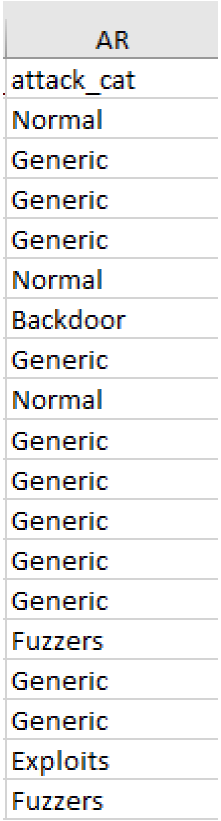
- Communication of Results

Using the random forest algorithm, a classifier can be created to spread the network data over a multitude of decision trees. The use of multiple decision trees allow data to be analysed more efficiently. A research paper titled The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems and written by Nour Moustafa and Jill Slay, highlights nine network packet classifications (Moustafa & Slay, 2015). The following nine categories will be utilised with the random forest classifier to sort network packet data for suspicious traffic.

- Fuzzers
    - An attacker attempts to discover security loopholes in an application, operating system, or a network by providing massive inputs of random data to force a system crash.
- Analysis
    - An attacker tries to breach a web application via ports, email, and web scripts.
- Backdoor
    - A technique designed to aid in stealthily bypassing normal authentication and securing unauthorised remote access to a device.
- DoS
    - Denial of Service attack (DoS attack) occupies an application to prevent authorised and legitimate requests from accessing a device or service.
- Exploit
    - Following instructions to take advantage of a bug, security fault, or vulnerability to perform unintended and malicious tasks on a network or host.
- Generic
    - A technique designed to block ciphers using hash functions by not considering the configuration of the cipher to cause a collision.
- Reconnaissance
    - A task that is typically completed early on to gather information about a device or network to discover a method of evading security controls.
- Shellcode
    - A type of malware used to penetrate and control a device or application, which runs on a shell.
- Worm
    - A common attack method used to spread malicious files through a network by replicating itself on other devices.

Utilising the nine categories highlighted above, a random forest classifier can be developed to analyse network traffic gathered from network monitoring. To differentiate the types of traffic, the random forest classifier integrates different trees for the types of traffic. As Scottish Glen has not set any limitations for hardware, the classifier can be configured to run in parallel with multiple processors to improve the efficiency.

A mock dataset of the network monitoring data was collected by Scottish Glen and provided in an Excel file format. The file contains a large amount of information such as the ID, Protocol, State Service, and Attack Category. The field with the most importance is the attack category field, as it provides an identifier to which of the nine previously mentioned attack categories the network traffic belongs to. A sample of the test data can be seen in Figure 4. The random forest classifier can create trees for each category of attack type so technical staff within Scottish Glen could search specifically for data that may have been flagged as suspicious or malicious.



AR
attack_cat
Normal
Generic
Generic
Generic
Normal
Backdoor
Generic
Normal
Generic
Generic
Generic
Generic
Generic
Fuzzers
Generic
Generic
Exploits
Fuzzers

Figure 4 Network Monitoring Test Dataset.

4 Standard Metrics and Measures

A vital step in developing the classifier with the machine learning algorithms is evaluation metrics. This ensures that the most efficient machine learning algorithm is chosen to complete the task. To evaluate the machine learning algorithms, the following metrics and measures have been considered for evaluation:

- Holdout Method
- Confusion Method
- Receiver Operating Characteristic Curve

4.1 Holdout Method

The holdout method is a simplistic technique, that divides the entire dataset into two sections, training, and testing data. Usually, the training data is greater in size compared to the testing data, as the algorithm only requires a small amount of data for testing. typically, the training to testing data ratio is 70:30 (Devi, 2023). The testing data fed through the classifier will provide an accurate representation of its performance. The original larger dataset is divided into the training and testing datasets that is required to provide accurate information throughout the testing process. The ratio split of test data to training data can be seen in Figure 5

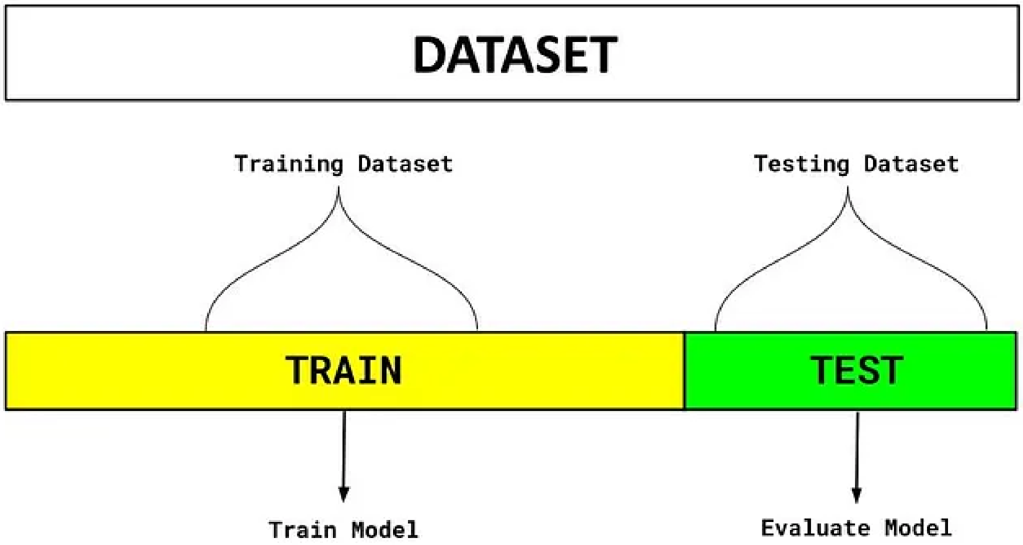


Figure 5 Holdout Method Dataset Split (Devi, 2023).

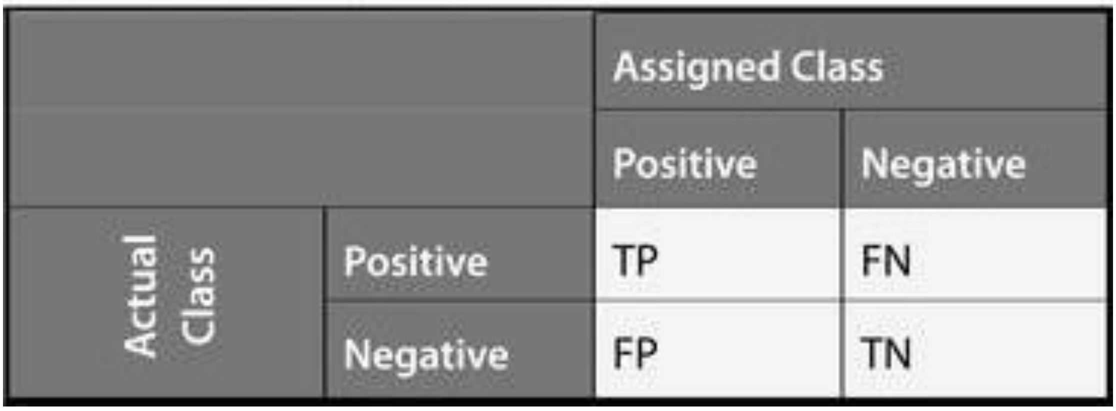
While the holdout method is simple and effective as an evaluation technique, it will not be used for this project. The holdout method relies on one dataset being provided for both training and testing. However, ScottishGlen has provided two different datasets and only utilising one would not maximise the efficiency of evaluating the classifier.

4.2 Confusion Matrix

The confusion matrix was the second method of evaluation to be considered. This matrix describes the performance of the machine learning algorithm and creates an in-depth comparison between the predicted outcome and the real outcome of the classifier. Using the predicted outcome and actual outcome, the following outputs can be determined:

- True Positive (TP)
  - Predicted positive and it is true.
- True Negative (TN)
  - Predicted negative and it is false.
- False Positive (FP)
  - Predicted positive and its false.
- False Negative (FN)
  - Predicted negative and its true.

A table version of the explanations above has been produced and can be seen in Figure 6. An example of a TN is if the outcome was predicted to be negative and the actual outcome was negative then the result equals a TN (Ting, 2011).



		Assigned Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

Figure 6 The outcomes of Classification into Positive and Negative Classes (Ting, 2011).

The overall accuracy of the classifier can be determined by reviewing the number of TP and FN, where FN class shows the inaccurate data classified. Confusion matrices are useful for reviewing the overall accuracy of a classifier.

4.3 Receiver Operating Characteristics Curve

The final method of evaluating the classifier that was considered is the Receiver Operating Characteristics Curve (ROC Curve). Using a ROC Curve allows for a visual representation of the comparisons between true positive and false positive rates. The area underneath the line is used to determine the accuracy of the classifier. Essentially, the closer the curve is to 0, the less accurate the classifier is. Conversely the closer the curve is to 1, the more accurate the data has become. Figure 7 below provides an example of a ROC graph with three different curves, A being the most accurate, and C being the least accurate (Tan, 2009).

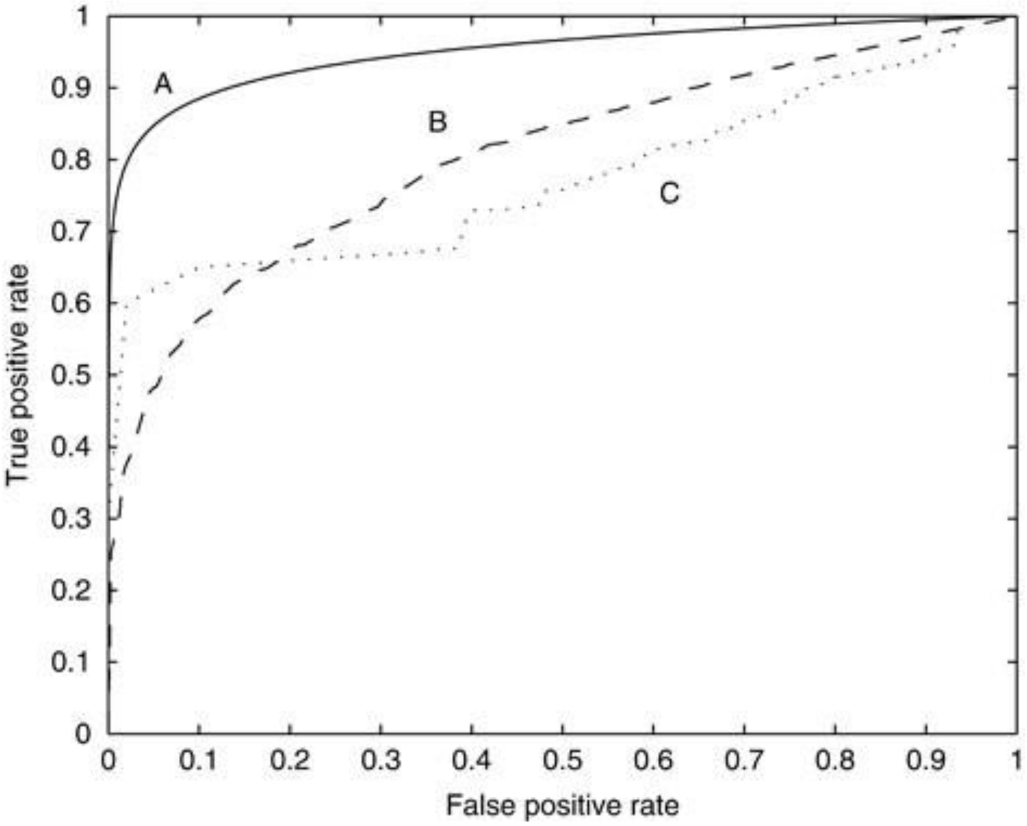


Figure 7 ROC Curve (Tan, 2009).

5 References

Devi, K., 2023. *Understanding Hold-Out Methods for Training Machine Learning Models.* [Online]
Available at: https://www.comet.com/site/blog/understanding-hold-out-methods-for-training-machine-learning-models/#:~:text=The%20hold-out%20method%20involves,both%20model%20evaluation%20and%20selection.
[Accessed 20 March 2024].

IBM, 2023. *What is random forest?.* [Online]
Available at: https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.
[Accessed 19 March 2024].

Kotsiantis, S. B., 2013. *Decision trees: a recent overview,* Patras: Springer Science+Business Media.

Meltzer, R., 2023. *What is Random Forest?.* [Online]
Available at: https://careerfoundry.com/en/blog/data-analytics/what-is-random-forest/#:~:text=3.,%2C%20patient%20history%2C%20and%20safety.
[Accessed 19 March 2024].

Menard, S. W., 2010. *Logistic Regression: From Introductory to Advanced Concepts and Applications.* 1 ed. California: SAGE Publications Inc..

Moustafa, N. & Slay, J., 2015. *The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems,* Kyoto: IEEE.

Park, H.-A., 2013. An introduction to logistic regression: from basic concepts to interpretation with particular attention to nursing domain. *Journal of Korean Academy of Nursing,* 43(2), pp. 154-164.

Rokach, L. & Maimon, O., 2005. Decision Trees. In: L. Rokach & O. Maimon, eds. *Data Mining and Knowledge Discovery Handbook.* Boston: Springer, pp. 165-192.

Tan, P.-N., 2009. Receiver Operating Characteristic. In: L. LIU & M. T. ÖZSU, eds. *Encyclopedia of Database Systems.* Boston: Springer, pp. 2349-2352.

Ting, K. M., 2011. Confusion Matrix. In: C. Sammut & G. I. Webb, eds. *Encyclopedia of Machine Learning.* Boston: Springer, p. 209.