Part 2. Machine Learning

1. Introduction

Machine learning, a form of AI, describes the process by which data is provided and analysed by a system, typically in the form of an algorithm, to identify patterns from which predictions can be made. This is done typically by a programmer providing instructions and parameters to set up an environment that is then given a large amount of data to respond to, or be "trained" with, and the patterns it identifies are generated rather than explicitly codified beforehand – hence it "learns".

Machine learning is seeing increased usage within business, with 92% of companies planning to increase their AI investments in the next three years (Mayer, et al., 2025). This makes sense when research suggests Machine Learning solutions can produce an average cost reduction of 31% over a similar three years to a business when employed (Polner, et al., 2022). Even though in 2025, machine learning is perhaps being overshadowed by generative AI within the AI market – it still represents a valuable investment for business solutions.

To this end, the assessor was provided with data from the technical staff at a small energy company, ScotGlen, in order to describe how it could be employed in the creation of a machine learning model capable of accurately making predictions increasing security readiness. Two datasets were provided, with the assessor asked to choose between Network Traffic data and System memory data – the assessor chose the Network Traffic data dataset. This assessment involved selecting an appropriate algorithm, a description of the process of constructing the model and the methods through which its success could be evaluated.

2. Algorithm Choice

Prior to selecting an algorithm, the data was observed and understood. Based on the intended function, the following was identified:

- The existing testing data was small, but represents packet capture data – this means that the testing data could grow very large very quickly, meaning that large dataset capability would be ideal.
- There are multiple types of attack category, meaning binary classification will be not very effective and multi-class tasks will be ideal. Knowing IF a packet is malicious is not the task as this is already labelled - identifying the attack category is the relevant aspect.
- The data is likely to be imbalanced, with there being more regular and non-malicious traffic than malicious traffic on average – the provided dataset does somewhat represent this, with 41 instances of "normal" traffic featuring no label that is unmalicious, with the next nearest being "generic" malicious packets at 21. There is *overall* more noteworthy packets than normal packets, but no individual category has more entries than the normal category.

The algorithm chosen had to be capable of classification, given that the desired output is a categorical value as opposed to a numerical value.

The following were identified as potentially valid algorithms:

Decision tree

Decision tree is a machine learning algorithm that employs a set of binary rules repeatably within a tree structure to classify an instance of data – making it an easy-to-interpret and understand ruleset given it resembles a flow chart or other tree structure. It's capable of handling both categorical and numerical data, which is ideal for the packet capture data that must be categorical. The model also has a reduction in need for data preprocessing due to the fact decision tree does not require scaling

or normalization to function successfully, two of the more time consuming areas of data preprocessing desirable to avoid. However, one issue with training against the provided data may be that due to the imbalanced nature, the tree may fail to appropriately classify the minority categories such as reconnaissance with only 2/100 examples. It also tends to have issues with data that has high dimensionality, which is potentially a problem given that the provided dataset and test data do employ a moderately high degree of dimensionality. This makes it ideal for solving classification problems such as spam detection or disease diagnosis. It is also capable of regression tasks, but that was not relevant for the current usage.

Despite these flaws, with appropriate data preprocessing a decision tree would represent a good choice for the presented problem.

Random Forest

Random forest functions as an improvement on Decision Tree algorithm by reducing overfitting, the problem wherein data trains too well on the training data, getting very high accuracy with it to the effect of functioning worse on new and unseen data. Everything said about decision trees as an algorithm applies to Random Forest as it is derivative. When looking at the data, the algorithm's ability to handle missing values is potentially a benefit owing to the fact that the testing data prior to preprocessing fails to identify a service in a lot of cases, simply reading as "-", effectively denoting a null value, meaning the fact it can account for this is potentially valuable. One limitation is that this is a fairly slow algorithm to employ, especially when compared with a decision tree in isolation. Random forest represents the algorithm most likely to be employed by the assessor.

K-Nearest Neighbour

K-Nearest Neighbour is a machine learning algorithm that classifies new instances based on a value (the k, in k nearest neighbour) of how many similar features are nearby. For example, if an instance is close to 3 red and 1 blue, it is categorized as a red. Due to this ability to measure against multiple other instances, it's strong in multi-classification problems which makes it a good choice for the provided packet capture data, which can be seen demonstrated below in Figure 1.

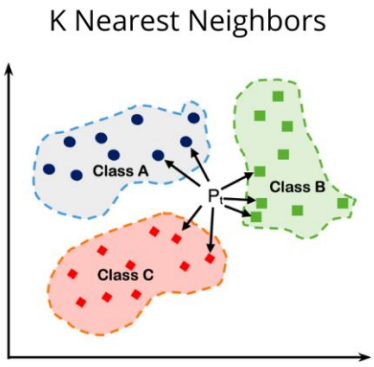

Figure 1 - An example of Multiclass K-Nearest Neighbour visualized (Banerjee, 2023)

The weaknesses of this algorithm are clear, with the primary one being that choosing an appropriate "k" value can prove difficult to identify – it's also quite cost-intensive and requires a lot of memory when used against large datasets making it comparatively resource-intensive, especially on datasets with high dimensionality. K-Nearest Neighbour is primarily used for classification problems, meaning it's good against any problem of that type such as cancer identification or handwriting classification. This would be a good algorithm to choose.

Other Algorithms were considered but were rejected.

Support Vector Machine

Support Vector Machine is an algorithm that places an object above or below a separation line, called a hyperplane, to classify it into one or two binary groups. It's ideal at problems working on smaller datasets with high dimensionality and is relatively memory efficient. Due to the manner in which SVM works, it's binary classifier first and foremost and is weak in multi-classification problems with no easy solution available making this one of its primary limitations. Problems that would be ideal to be solved with SVM include recognising if something is or is not a face or other such image classification and recognizing handwriting.

With regard to the network capture data, Support Vector Machine is unlikely to be significantly useful owing to the fact it fails to account for multi-classification problems. Data within the packet capture could be categorized as malicious and non-malicious but this would not serve to be useful as this has already taken place with the "label" field in the training data, and it would be unable to identify an attack category of which there are more than 2.

3. Building Model

Data Collection

Data collection is the process by which relevant and accurate data is acquired from available sources. This was already undertaken by the staff at Scotglen, therefore step in the typical machine learning pipeline was skipped for the assessor.

Preprocessing

The data provided to a machine learning model must first be processed to ensure it can be ingested appropriately and the system can handle what is being provided. Based on the provided dataset, fairly extensive preprocessing was necessary. To begin with, missing values should be handled – in this case, accounting for a large portion of the "service" field – this can be done by simply replacing these missing values to "0", which is a value representing no service and also encodes it for interpretation. Duplicates should be removed and are already effectively handled through the usage of an ID system ensuring the uniqueness of data. Outlier data should be removed and dealt with, but in this case, would be hard to spot, and the fact that the provided dataset appeared to be labelled it may imply a degree of outlier removal had already taken place, which observation confirmed given almost all values were within reasonable expectations. One value that seems fraudulent is ID 43, an ARP protocoled packet which lists the duration as "0" – which appears impossible for any operation, especially with the degree of significant figures observed in other time values therefore this should likely be removed.

The features present are of type numerical and categorical – for instance protocol (field name "proto") being categorical, with something like "http". This is compared to something like "response_body_length" which is strictly numerical. In order to make the data more usable, the categorical features should be encoded – wherein they are converted to numerical values. An example of doing this with state category could look like something shown below in Table 1.

Table 1 - Classifier conversion from categorical to numerical

Categorical classifier of field "state"	Revised numerical categorization of "state"
INT	0
FIN	1
CON	2

REQ	3

The data was already split into appropriate datasets, with a training and testing dataset provided. A validation dataset was not provided.

Given that the ideal model in use is random forest, a subset of decision tree – then normalization and scaling are not required.

Modelling

Modelling denotes the selection of an appropriate model algorithm based on the identified data characteristics and intended outcomes. In this case, an appropriate model would classification based capable of assigning categorical responses. An example would be random forest or K-nearest neighbour, as discussed earlier within the report. This model is provided with the pre-processed training data to learn and identify instances of malicious packets and categorize them appropriately.

Analysis

Analysis denotes the evaluation of a model to ensure its functioning and measure its success on testing data. This is done by assessing how well the model generalizes to new data – primarily through quantitative evaluation metrics, described further down.

Results

Results refer to the communication of model performance through visualization. The most common example of how this can be performed is by graphing the data, with many examples of multiclass classifier data visualization shown below in Figure 2.

Figure 2 - Examples of mutliclass visualization graphs (Looney, 2018)

4. Evaluation Metrics

A number of metrics are significant for machine learning that allow for the measurement of a successful model – the majority of these are based upon the concept of a “confusion matrix” (see Table 2) from which formulas are used to calculate the other metrics.

Table 2 - Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

Precision

Precision measures the accuracy of all positive predictions, so it measures what proportion of positive predictions are true positive: this is primarily concerned with the reduction of false positives. A false positive, while unwanted, is a preferable result to the alternative when it comes to packet identification and categorization. The formula for this is:

Precision = True Positives (TP)/(False Positives (FP)+True Positives (TP))

Recall

Recall measures how often the model correctly identifies positive instances (true positives) among the given positive instances. Essentially: out of all the malicious attacks, how many does the model identify? This is primarily concerned with the reduction of false negatives, where the model misses an instance of a genuinely malicious packet and predicts it to be negative. Reducing false negatives is far more important, as it could be potentially far more costly to NOT identify a malicious attack than it would be to incorrectly identify a safe packet as a malicious one. Data loss is preferable to infection. The formula for this is:

Recall = True Positives (TP)/(True Positives (TP) + False Negatives (FN))

F1 Score

When precision and recall are combined, an F1 score can be derived which is considered an industry-standard metric for the reliability of a model – so these should be considered the two most valuable evaluation metrics for assessment of performance if they must be prioritized. These can be combined in the formula:

F1 = 2 x (Precision x Recall) / (Precision + Recall)

Accuracy

Accuracy is a measure of how accurate a machine learning model is overall. This means the number of true positives weighed against all predictions. This is considered initially to be an important metric, but can fall pray to problems when dealing with imbalanced classes. To explain: if a model were to have 60 packets where 3 of them were malicious, but it predicted all packets to be safe – despite the fact that all the malicious packets were missed, the model would still be deemed as having a 95% accuracy – making this metric, while useful, potentially unreliable hence why it should not be the primary or only metric considered. The formula for this is:

Accuracy = True Positives (TP) + True Negatives (TN) / Total Samples (True Positives (TP) + False Positives (FP) + True Negatives (TN) + True Positives (TP))

References

Banerjee, S., 2023. *Unlocking the Power of K-Nearest Neighbors (KNN) Classifier: Your Guide to Effective Classification.* [Online]
Available at: https://shekhar-banerjee96.medium.com/unlocking-the-power-of-k-nearest-neighbors-knn-classifier-your-guide-to-effective-classification-b50cb74fbfc
[Accessed 31 March 2025].

Looney, O., 2018. *Visualizing Multiclass Classification Results.* [Online]
Available at: https://www.oranlooney.com/post/viz-tsne/
[Accessed 31 May 2025].

Mayer, H., Yee, L., Chui, M. & Roberts, R., 2025. *Superagency in the workplace: Empowering people to unlock AI's full potential,* s.l.: s.n.

Polner, A., Wright, D., Schaefer, G. & Thopalli, K., 2022. *Automation with intelligence.* [Online] Available at: https://www2.deloitte.com/us/en/insights/focus/technology-and-the-future-of-work/intelligent-automation-2022-survey-results.html
[Accessed 30 March 2025].