CMP417 - Engineering Resilient Systems 1

Machine Learning

Jack Bowker - 1803838

April 2021

ABSTRACT

This report is a short look at different Machine Learning solutions that could be adopted by a small company, whose IT infrastructure is being targeted by a hacktivist group. The introduction covers how common network-based threats are to businesses, as well as laying out the current situation at the company. The Background section focuses on the fundamentals of Machine Learning and how it has a proven track record of use for security applications. The Implementation section goes into detail on different algorithms the company could implement, in particular K-Nearest Neighbour and K-Means clustering, as well as describing the advantages and disadvantages of each. This section also details the construction of a model and steps that should be carried out when building one. Finally, he Evaluation section covers different ways this Machine Learning model could be tested and assessed to make sure it's performing as intended.

1. Introduction

This paper will provide security recommendations for a small company, who after experiencing threats of a cyber-attack from a hacktivist group are looking to improve their overall security posture. This follows on from a review was done for the company on their development processes, specifically looking into Insecure Communication of their mobile application.

Traffic was monitored from the company's network infrastructure, and according to the technical team some of it looked concerning.

According to PT Security, as shown in Figure 1, the vast majority of businesses using network packet analysis have detected either malware on their network, attempts to brute force or exploit attempts of web vulnerabilities.

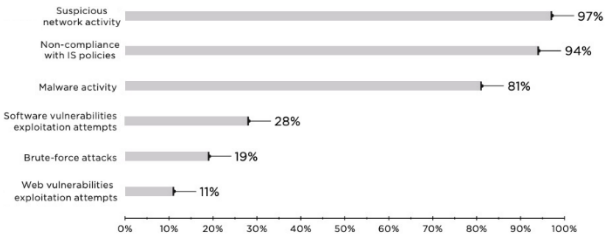

Figure 1 - Percentage of companies experiencing cybersecurity threats (PTSecurity, 2020)

According to the technical team, the traffic recorded is difficult to understand. Manually searching through network traffic is extremely time consuming as the traffic often has a low signal to noise ratio, with the vast majority of requests being legitimate. An effective way of processing this network traffic would be using Machine Learning, especially since the data the technical team provided includes sample traffic with categories attached. Machine Learning is able to detect malicious traffic that hasn't been seen before by detecting patterns used in the past, meaning it can often detect malicious traffic even while it's encrypted (Cisco, 2021).

2. Background

The area of Machine Learning is incredibly wide in scope and can be used for a multitude of purposes. As shown in Figure 2, there are three fundamental areas in the topic of Machine Learning. These areas include Supervised learning, Unsupervised learning, and Reinforcement learning.

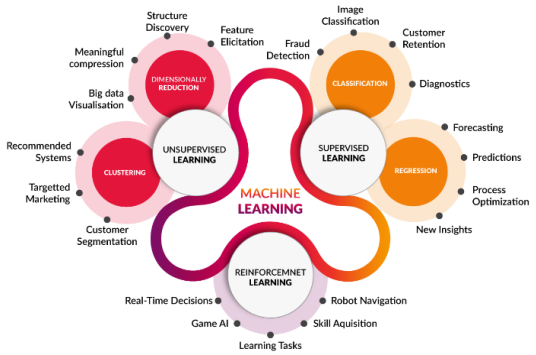

Figure 2 - How different areas of machine learning are used (Cognub, 2019)

Supervised learning uses examples that are given with context, for example data that is labelled into categories

(Waseem, 2019). Supervised Machine Learning algorithms are commonly used for Classification, where after being trained using relevant data the algorithm can determine, for example, detecting whether a photo contains a Dog, Human, or Landscape. As well as this, Supervised algorithms are also often used for Regression tasks, where based on previous values, the next can be predicted - for example predicting the profit of a company, based on previous forecasts.

Unsupervised learning is used when no specific categories exist or if there is no labelled data for the intended purpose. It is commonly used for Clustering, which groups similar data together to find trends, groups and anomalies in unlabelled data (Polyakov, 2019).

Reinforcement learning is used in algorithms where a predefined goal is to be achieved. Parameters are given to the algorithm to optimize for, and each run will gradually improve as the algorithm uses information gained in past attempts. A famous use of this is teaching an artificial intelligence model to complete pathfinding in video games in the shortest amount of time (Trivedi, 2020), where on each run the model will learn from the previous runs and reduce the time it takes for it to complete them.

It has been proven that Machine Learning techniques can be effective in detecting malicious activity in Computer software and networks. For the release of Microsoft's Windows 10 operating system, the malware protection system was revamped by moving away from static signature detection which largely relied on input from malware researchers and known malware, to a model that uses Machine Learning behaviour-based prediction to detect malicious activity (*Why Windows Defender Antivirus is the most deployed in the enterprise*, 2018). Shown in Figure 3 is the stack of Machine Learning technologies Microsoft integrates into its Windows Defender software.

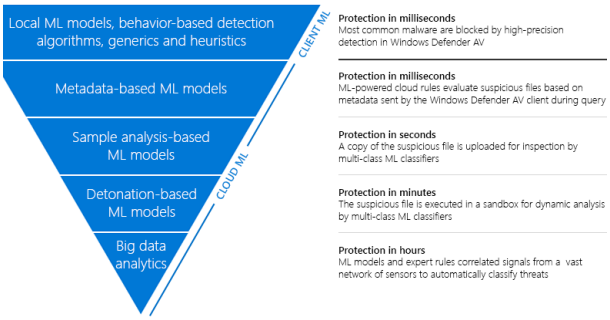

Figure 3 - Microsoft Windows Defender machine learning stack

3. Implementation

In order to effectively implement a Machine Learning model to filter through this data, an appropriate algorithm must be chosen that would be suited to this purpose. Depending on the time constraints/budget the company has to work with, different algorithms could provide more value.

Algorithm 1: K-Nearest Neighbours

The K-Nearest Neighbours (k-NN) algorithm is a popular supervised algorithm, suited for classification and regression tasks. This algorithm requires a labelled training set of past network traffic, and would predict the category of new data based on how the majority of its "Nearest Neighbours" are classified, as demonstrated in Figure 4, where the example $k = 5$ means the category is decided by whichever category the majority of the 5 nearest (most similar) existing results are.

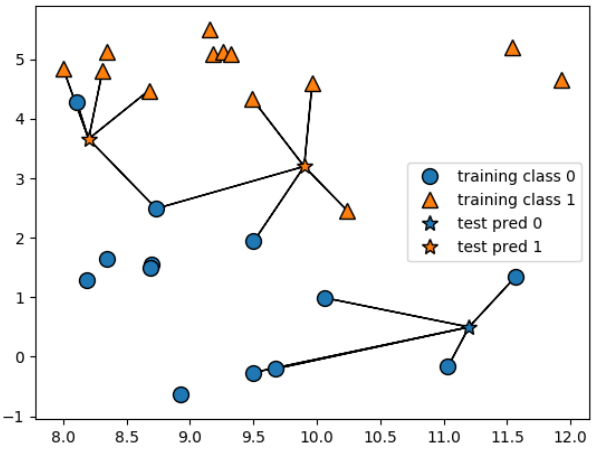

Figure 4 - Demonstration of k-NN in action (Granville, 2020)

An advantage of K-Nearest Neighbours versus other supervised algorithms such as Support Vector Machines (SVMs) or Linear Regression is that it doesn't require a training period, meaning the learning occurs whilst it runs. This also means new data can be added as the algorithm runs so correct classifications can be added to increase the overall accuracy of the model. On the other hand, due to the fact that a lot of the processing is done in real time, with large datasets such as extensive network traffic logs finding the distance between new and existing points can become resource intensive. (Kumar, 2019)

Although supervised algorithms can provide more accurate results, another disadvantage of k-NN is the fact that it is supervised and requires sanitised and correctly labelled data in order to function accurately. If the company is currently being threatened by Hacktivists, it might not be sensible to spend time running through past network activity and creating a dataset the K-Nearest Neighbours algorithm can learn from.

Algorithm 2: K-Means Clustering

The K-Means Clustering (k-MC) algorithm is a popular unsupervised clustering algorithm. Unlike k-Nearest Neighbour, this algorithm is unsupervised meaning the data used for training doesn't need to be labelled. Instead, "features" are defined by the user and fed to the algorithm to help determine what it should look for. As opposed to classification algorithms which have predefined classes, clustering algorithms such as k-MC group data samples into k-clusters.

In Figure 5, the value $k = 3$, as it refers to the number of "centroids" in the dataset. In the case of the company, there will need to be 10 centroids in order to cover "Normal, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms".

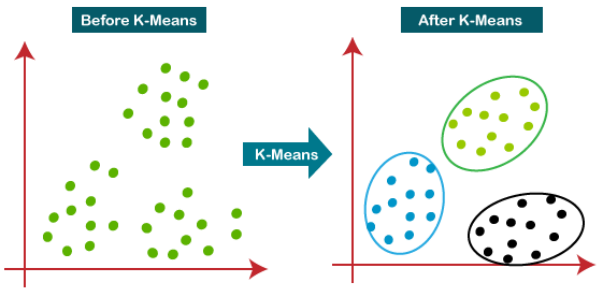

Figure 5 - Visual example of how K-Means clustering groups data (javatpoint, 2018)

An advantage as mentioned previously with K-Means Clustering is that it's unsupervised so won't require the company to filter through existing data to train the dataset. Along with this, the algorithm is more suited to large datasets so would therefore be much less resource intensive to scale to monitoring network traffic. Another disadvantage of this clustering algorithm is that it assumes each cluster will have roughly equal distribution of data, which is unlikely in the case of the company as it's likely the vast majority of the traffic will be normal. This can cause bias and the possibility of malicious traffic being mistaken for normal

Building a model

The technical team at the company provided two Excel spreadsheet files for examination – a training file and a testing file. A sample of this data is shown in Figure 6.

id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload
1	0.121478	tcp	-	FIN	6	4	258	172	74.08749	252	254	14158.94238	8495.365234
2	0.000009	udp	dns	INT	2	0	114	0	111111.1072	254	0	50666664	0
3	0.000008	udp	dns	INT	2	0	114	0	125000.0003	254	0	57000000	0
4	0.000008	udp	dns	INT	2	0	114	0	125000.0003	254	0	57000000	0
5	0.649302	tcp	-	FIN	14	38	734	42014	78.473372	62	252	8395.112305	503571.3125
6	0.000008	br-sat-mon	-	INT	2	0	200	0	125000.0003	254	0	100000000	0
7	0.000006	udp	dns	INT	2	0	114	0	166666.6608	254	0	76000000	0
8	1.681642	tcp	ftp	FIN	12	12	628	770	13.677108	62	252	2740.178955	3358.62207
9	0.000001	udp	dns	INT	2	0	114	0	1000000.003	254	0	456000000	0
10	0.000008	udp	dns	INT	2	0	114	0	125000.0003	254	0	57000000	0
11	0.000001	udp	dns	INT	2	0	114	0	1000000.003	254	0	456000000	0
12	0.000009	udp	dns	INT	2	0	114	0	111111.1072	254	0	50666664	0
13	0.000009	udp	dns	INT	2	0	114	0	111111.1072	254	0	50666664	0
14	1.309505	tcp	http	FIN	60	14	68197	612	55.746256	254	252	403687.625	3476.122803
15	0.000009	udp	dns	INT	2	0	114	0	111111.1072	254	0	50666664	0
16	0.000009	udp	dns	INT	2	0	114	0	111111.1072	254	0	50666664	0
17	0.811134	tcp	http	FIN	10	16	844	10544	30.821049	62	252	7495.679199	97493.14063
18	59.814823	udp	dns	REQ	70	0	12320	0	1.15356	254	0	1624.212769	0
19	0.000003	udp	dns	INT	2	0	114	0	333333.3215	254	0	152000000	0
20	0.000009	udp	dns	INT	2	0	114	0	111111.1072	254	0	50666664	0
21	0.380537	tcp	-	FIN	10	6	534	268	39.41738	254	252	1012.02539	4709.134766
22	0.000003	udp	dns	INT	2	0	114	0	333333.3215	254	0	152000000	0
23	0.000008	udp	-	INT	2	0	332	0	125000.0003	254	0	166000000	0
24	1.916419	tcp	-	FIN	10	8	742	354	8.870711	254	252	2788.53418	1294.0802
25	2.033391	tcp	-	FIN	12	10	1366	836	10.327576	254	252	4923.696289	2962.538818
26	0.000014	mobile	-	INT	1	0	60	0	0	0	0	0	0
27	0.000011	udp	-	INT	2	0	1270	0	90909.0902	254	0	46181376	0
28	0.000009	unas	-	INT	2	0	200	0	111111.1072	254	0	88888888	0
29	0.000005	udp	dns	INT	2	0	114	0	200000.0051	254	0	91200000	0
30	0.00001	unas	-	INT	2	0	200	0	100000.0025	254	0	80000000	0

Figure 6 - Network traffic data given by technical team

In order to build an effective Machine Learning model, implementing a standardized data pipeline is crucial.

This process involves the fetching, sanitisation, and pre-processing of data. Along with this, it's where modelling and analysis will be carried out.

In the case of the training data the company provided in an Excel spreadsheet, it should be manually inspected to ensure both that the data provided gives a good example of what the traffic normally looks like, as well as removing rogue/incorrect outliers in the dataset. If working with text data, it's good practice to sanitise this by removing things such as emojis/non-standard characters. As well as this, the data should be converted to a machine-readable standard format such as CSV.

Once a first model is complete, testing should be done by using similar but not the same data on the algorithm, to ensure output looks as expected. After this, training should be continued using relevant data and parameters and weighting can be changed to improve the accuracy of the model further.

4. Evaluation

There are a multitude of methods to measure the accuracy of machine learning models. Assessing and ensuring the accuracy of Machine Learning models is important especially if the algorithm is going to be used in a scenario such as detecting malicious traffic on a network, where False Positives may be much more tolerable than False Negatives.

Realistic testing data is vital in ensuring the model will act appropriately when dealing with live data and having more of this data will give a more accurate picture of how a model will act.

A reliable method for determining the accuracy of a model is a Confusion Matrix. This matrix can show what a model struggles with the most by visualising the percentage of results that were predicted correctly/incorrectly. (Kulkarni, 2020) The rate can range from 0-1, with 0 representing 100% accuracy and 1 representing 0% accuracy.

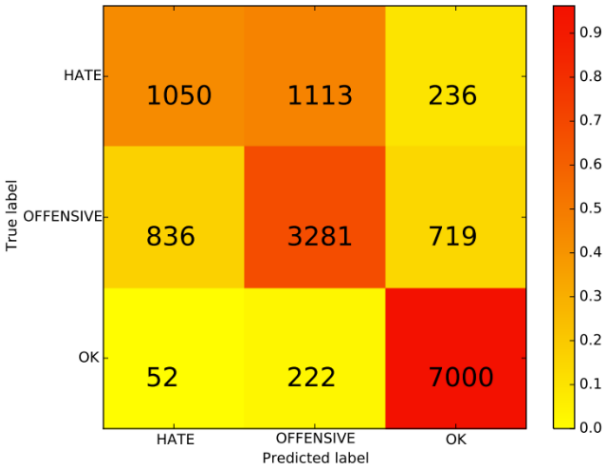

Figure 7 - Confusion matrix from paper measuring model accuracy (Malmasi and Zampieri, 2017)

The formulas shown in Figure 8 show how the False Positive (FP) and False Negative (FN) rates are found, with TP representing results accurately recorded as true and TN representing correctly recorded negative results.

$$False\ Positive\ rate = \frac{FP}{FP + TN}$$

$$False\ Negative\ rate = \frac{FN}{FN + TP}$$

Figure 8 - Formulas used to calculate False Positive (FP) and False Negative (FN)

Another method of visualising the accuracy of a model is a Receiver Operating Characteristic (ROC) Curve. This metric graphs the performance of models by plotting the True Positive (TP) and False Positive (FP) rates over different decision thresholds as shown in Figure 9 (Google Developers, 2020).

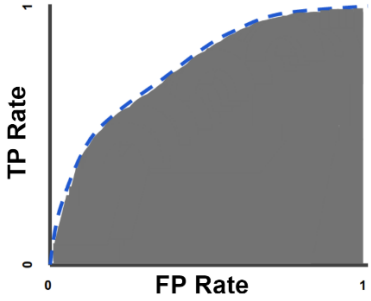

Figure 9 - ROC curve showing lower vs higher classification thresholds

5. References

Cisco (2021) *What Is Machine Learning in Security?*, *Cisco*.Available at: https://www.cisco.com/c/en/us/products/security/machine-learning-security.html (Accessed: 14 April 2021).

Cognub (2019) 'COGNITIVE COMPUTING AND MACHINE LEARNING', *Cognub*. Available at: http://www.cognub.com/index.php/cognitive-platform/ (Accessed: 14 April 2021).

Google Developers (2020) *Classification: ROC Curve and AUC, Google Developers*. Available at: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (Accessed: 16 April 2021).

Granville, V. (2020) *K-Nearest Neighbors (KNN): Solving Classification Problems*. Available at: https://www.datasciencecentral.com/profiles/blogs/k-nearest-neighbors-knn-solving-classification-problems (Accessed: 15 April 2021).

javatpoint (2018) *K-Means Clustering Algorithm - Javatpoint, www.javatpoint.com*. Available at: https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning (Accessed: 16 April 2021).

Kulkarni (2020) *Confusion Matrix - an overview | ScienceDirect Topics*. Available at: https://www.sciencedirect.com/topics/engineering/confusion-matrix (Accessed: 16 April 2021).

Kumar, N. (2019) 'The Professionals Point: Advantages and Disadvantages of KNN Algorithm in Machine Learning', *The Professionals Point*, 23 February. Available at: http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-knn.html (Accessed: 16 April 2021).

Malmasi, S. and Zampieri, M. (2017) 'Detecting Hate Speech in Social Media', *arXiv:1712.06427 [cs]*. Available at: http://arxiv.org/abs/1712.06427 (Accessed: 3 March 2021).

Polyakov, A. (2019) *Machine Learning for Cybersecurity 101, Medium*. Available at: https://towardsdatascience.com/machine-learning-for-cybersecurity-101-7822b802790b (Accessed: 14 April 2021).

PTSecurity (2020) *Top cybersecurity threats on enterprise networks*. Available at: https://www.ptsecurity.com/ww-en/analytics/network-traffic-analysis-2020/ (Accessed: 14 April 2021).

Trivedi, C. (2020) *Game Level Design with Reinforcement Learning, Medium*. Available at: https://towardsdatascience.com/game-level-design-with-reinforcement-learning-fa6eb585eb4e (Accessed: 14 April 2021).

Waseem, M. (2019) 'Classification In Machine Learning | Classification Algorithms', *Edureka*, 4 December. Available at: https://www.edureka.co/blog/classification-in-machine-learning/ (Accessed: 14 April 2021).

Why Windows Defender Antivirus is the most deployed in the enterprise (2018) *Microsoft Security*. Available at: https://www.microsoft.com/security/blog/2018/03/22/why-windows-defender-antivirus-is-the-most-deployed-in-the-enterprise/ (Accessed: 15 April 2021).