Engineering Resilient Systems 1
Machine Learning

Thomas MacKinnon
1704872

ABSTRACT

This report aims to give an effective recommendation for possible Machine Learning algorithms used to protect a small company's network. This is achieved through an overview of different types of Machine learning, with real world examples and the history of the subject. Several different Machine Learning Algorithms are recommended throughout the paper, each suited to the task given, with an objective look at the positives and negatives of the solution, discussing any limitations it might bring. To aid the company in deployment, a comprehensive Implementation section has been provided, detailing each stage of development and construction of a data pipeline. Recommended Evaluation Metrics are also provided, to assure that the algorithm is working correctly and so that testing can be conducted smoothly.

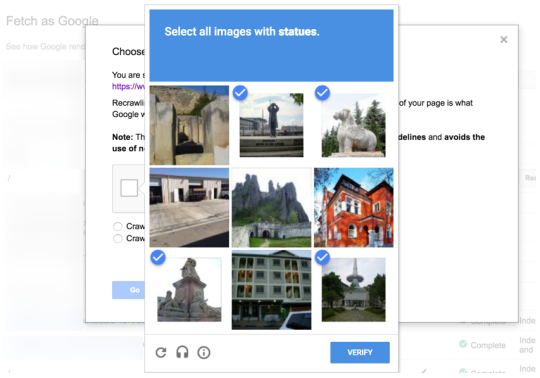

Figure 1: reCaptcha acquiring Training data of statues

1 INTRODUCTION

Machine Learning, as defined by Ellrodt et al. (2018), is the process of recognizing patterns in large volumes of data in order to perform a specific task. These Machine Learning Algorithms power many systems in our everyday life, one prime example being content recommendations in streaming services such as Netflix. Whilst using the platform every piece of data you knowingly give (favourite genres, preferred content length) gets collected and fed to the algorithm to ensure you're happy, entertained, and keep you paying for the service (Hao, 2018).

Machine Learning algorithms work through the feeding of training data with the goal of finding patterns amongst it. However the type of data defines the way the algorithms works, creating three distinct methods in the training of Algorithms.

Supervised learning occurs when training data is labeled, and is the most prevalent training method. The training data is often small, with many similarities to the final dataset, and continues learning even after being deployed. Google's reCaptcha is an excellent example of this, users label images as part of authentication which is fed into the algorithm so that it can find more patterns to distinguish similar items, like statues from buildings as seen in figure 1. Google does this to improve their other services, like Image search or Maps, to better improve their users experience (o'Malley, 2018).

Unsupervised learning unsurprisingly occurs when training data is not labeled, which saves the time of a human supervisor in making data machine-readable. This also means that more data can be used to train the algorithm, and has the benefit of finding "hidden structures" in data that wouldn't have been found by humans. An example of Unsupervised learning is Clustering, where the algorithm finds the structure in uncategorised data and clusters it into specific groups (Potentia, 2021).

Lastly there is Reinforcement Learning, which bases it training of trial and error with a predefined goal, each desired output is encouraged whilst non desired outputs are discouraged. The algorithm repeats until it finds the most effective solution to the task and is reinforced for each favorable output. Typically this is found in Map programs when finding the shortest route between two destinations.

Machine Learning has been around since the early 1950s, originally titled by Arthur Samuel when developing a self learning checkers program (Foote, 2019). It used an early form of Reinforcement learning in order to find the best possible for the current game through a minimax algorithm (minimum loss, maximum gain). Machine learning would evolve over the years, 1957 saw the creation of "Perceptron", an image recognition software that introduced Supervised learning training data, a couple years later a shortest route algorithm was used by travelling salespersons to make their journeys more efficient (Sharma, 2017). Machine learning

quickly became an essential tool for businesses and even everyday life, and will continue growing with our technological needs.

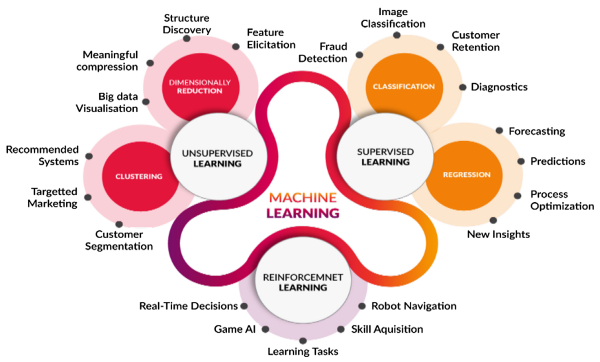

Figure 2: Machine Learning's various uses

Machine learning is excellent in situations where a solution will need continuous improvement after being deployed, such as Chatbots. Instead of hiring a low skilled worker a Chatbot can fill the roll (or even multiple rolls) for a fraction of the cost, and with Machine learning adapt to new slang/vocabulary from customers. Intrusion Detection Systems (IDS) have become crucial in Cybersecurity, as Machine learning aids in finding unknown attacks through predictive patterns (Dua & Kunal, 2019). Machine learning is being utilised more and more to improve our lifes, figure 2 shows many uses for machine learning.

2 CONTEXT

Recently, "Company Redacted" has received threats from a Hacktivist organisation after a blog post by the CEO. The nature of this Cyber-attack is not know, so the whole system is being improved in order to stop this pending attack. The company consists of a small team of developers and technical staff with no dedicated security specialist, this means any changes implemented have to be manageable for the current employees. Previously improvements have been made on the Mobile Application used to access back-end databases, however serious concerns have been raised over the company's network.

Analysis of network traffic revealed several different attack methods being used by outside sources, including DDoS and exploitation attempts. To prevent attacks through the company network a Machine Learning Algorithm will be used, aiming to detect and classify any malicious packets. The retrieved network traffic has already been formatted to be used as training data for this Algorithm. The data has already been labeled by staff, meaning the Algorithm will be using Supervised Learning as its method for improving.

This report aims to give an appropriate and effective recommendations for which Machine Learning Algorithm should be implemented to best combat Cyber-attacks. This includes an objective look at each Algorithm, detailing the strengths, weaknesses and any limitation it may have. Development stages of a Machine Learning Algorithm will also be discussed, followed by an explanation on how the data pipeline works and how to properly evaluate the Algorithm.

3 RECOMMENDATION

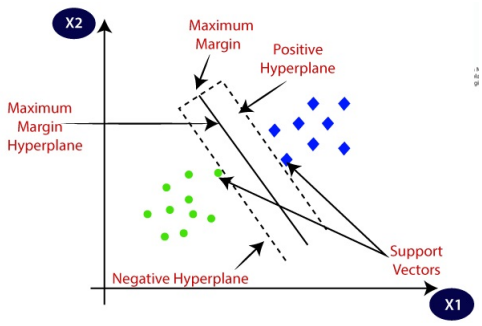

Figure 3: SVM positive/negative hyperplane margins

Support Vector Machines:

The first recommendation for the company is a Support Vector Machine Algorithm (SVM), which is used as a binary classification solution. SVM algorithms works by creating two categories for data to fall into, and divides them using a hyperplane (MonkeyLearn, 2021). Training data is mapped to coordinates as seen in figure 3 (JavaTPoint, 2021) and divided so that margin between the hyperplane and near data points is as large as possible. This is done to avoid data being misidentified in the wrong category, which could cause serious issues depending on the system (Ray, 2017). In the case of "Company Redacted" the two categories would be "safe" (represented by the green circles) and "Malicious" (represented by the blue diamonds), network traffic would be assigned either side of the hyperplane depending on its properties.

Sometimes data is non-linear and a straight hyperplane cannot be mapped without adding a new dimension, as seen on figure 4. This is done through a "Kernel Trick", which is less intensive on computation resources than other methods of transformation (Lessmann, 2004).

Support Vector Machine Algorithms work very effectively for its computational power consumption, and using tools like "Kernel tricks" can map data points accurately even when there is no clear hyperplane. SVM algorithms also work well with smaller data samples, meaning the data supplied

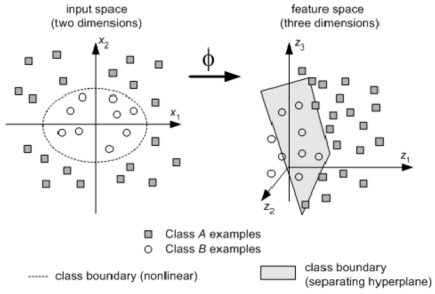

Figure 4: Kernel trick to make a hyperplane

will work very well at training since they are both around hundred lines each (training is 100 lines, testing is 80 lines).

SVMs do not come without its flaws, for one the larger the dataset is the less effective the algorithm will be. This is a considerable issue with "Company Redacted", as although it is a small company there will still be a large amount of network traffic. This could lead to the SVM algorithm becoming less effective over time, especially if the company grows in size. SVMs also struggle when classes overlap, which is often the case with network traffic meaning the algorithm would be more limited when finding malicious packets (Kumar, 2019).

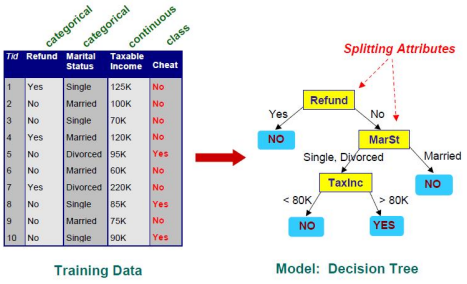

Figure 5: Decision Tree Example

Decision Trees:

Decision trees are another form of binary classification, which filters data depending on certain parameters until it is one of the two categories, known as binary recursive partitioning (Chakure, 2019). Figure 5 (Bakos, 2010) shows a decision tree for the likely hood that a partner will cheat during a relationship, taking several factors from the labeled data and finding the underlying pattern. In the case of "Company Redacted" network data would be partitioned down the appropriate branches of the tree, and depending on the content will either be predicted as "Safe" or "Malicious".

Decision Trees are visual by nature and generate readable rule, meaning staff will have an easy time at understanding this security process. The amount of partitioning also increases with larger training data, meaning passable malicious packets are more likely to be spotted overtime. Training data also needs less preparation from staff members compared to other algorithms, saving additional time when development begins (Scikit Learn, 2021).

Decision trees due suffer from an issue called "Overfitting", which occurs when too many hypotheses are made with the training data. This greatly reduces the error rate in the training data, but when testing data (or unseen data) is introduced the error rate increases, which is undesirable after deployment. Pruning aids in preventing this issue through removing unnecessary nodes to make the tree simpler, this can be done before or after training. Underfitting is also a concern, which occurs when the Decision tree is over pruned, making both the training and testing error rate increase. Without proper pruning maintenance the tree becomes over complicated making it hard for staff to understand.

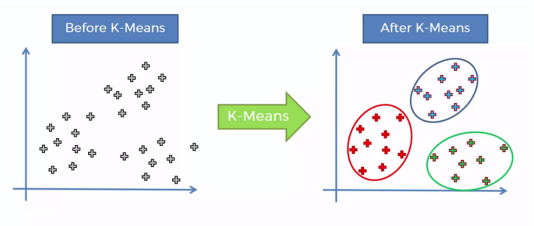

Figure 6: k-Means Clustering on random data points

k-Means Clustering:

Both the previous recommendations have been Supervised Machine Learning algorithms, as the training data has been labeled, however that does not negate the use of an Unsupervised Learning algorithm. k-Means Clustering (k-MC) is a type of unsupervised algorithm that groups similar data points together into a cluster to discover an underlying pattern (Garbade, 2018). Figure 6 shows k-MC clustering the data points into three distinct groups, meaning that *k=3* (Jain, 2021).

This provides are noticeable advantage over binary classifiers, as different attack methods can be clustered together rather than just marked as "Malicious". The network traffic had ten different attack categories, being: Normal, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. If k-Means Clustering was implemented in "Company Redacted" then attack data would be a lot more useful, as the number of each type attacks would be easily readable. If the majority of the packets were clustered under "DoS" then security upgrades could be made to specifically counter that issue.

Due to it using Unsupervised learning methods the algorithm will not require any company time in labeling training data,

and can begin operating almost immediately after development, which is relatively simple compared to other options. k-MC scales very well with large datasets, meaning it is ideal for network traffic. Staff will also find k-Means Clustering simple to understand by providing visually interesting graphs to display data(Henrique, 2019).

However, with k-Mean clustering surveying network traffic there is the issue of an oversized cluster, as k-MC assumes that all clusters will be close to equal in size. The majority of the traffic will be "Normal", leading to a heavily skewed graph, and the risk of malicious packets being misidentified. Outliers are also a cause for concern, as they can skew clusters into overlapping creating confusion in detecting dangerous network traffic (Google Developer, 2021).

4 IMPLEMENTATION

Development of a Machine Learning Algorithm:

To aid "Company Redacted" in implementing a Machine Learning Algorithm a seven staged development process has been provided (Mayo, 2018).

Stage One - Collection of data should be done first, aiming to gather as much quality data to be used in further steps.

Stage Two - Data should then be prepared properly, this is done through removal of duplicates, correcting errors, removal of any outlier data points, and dealing with missing information. Prepared data should then be split, creating the training data and the testing data to be used in evaluation. Training/Testing data should be stored in a machine readable format, such as a CSV file (Data-Driven Science, 2020).

Stage Three - Choosing a Machine Learning Algorithm is next, making sure it is the best fit for the task.

Stage Four - The training dataset prepared is now used to train the model over a series of iterations, each aiming to make correct predictions more often.

Stage Five - The objective performance of the model is then measured using the unseen Testing data prepared, aiming to see if it can make accurate predictions in a realistic setting.

Stage Six - This stage simply aims to improve the performance of the model by tuning its parameters. This could be changing the maximum depth allowed of the Decision Tree (Jordan, 2017).

Stage Seven - The Final step is to deploy the model and allow it to process real world data, solving whatever task it was built for.

Data Pipeline:

Machine learning algorithms use a Data Pipeline to automate its workflow, by taking raw data and formatting it into valuable data. Developing a Data Pipeline provides many benefits, one being an improvement of predictive performance, and allowing for easier implementation of other algorithms or improvements. Like with development of a Machine learning algorithm there are several stage to how the Data Pipeline works (Gill, 2020).

Stage One - Ingestion is the first stage, which is simply collecting data to be used.

Stage Two - Pre-Processing involves cleaning the data of any errors or unnecessary data, and is prepared for modeling.

Stage Three - Modeling involves finding the right algorithm to accomplish the task at hand and feeding in data for learning.

Stage Four - Evaluation is essentially just testing the model with training data until the predictions are consistently correct.

Stage Five - Results from the algorithm are then communicated to the client, visualising data with priorities as the focus. Then the model is deployed, if it is ready to be used in a real world setting.

5 EVALUATION

To guarantee that the Machine Learning Algorithm is performing well and making accurate predictions an evaluation metric must be put in place. This assessed the accuracy and effectiveness of the model, visualising results to the client. Using realistic testing data is vital in evaluation, as to properly show how the algorithm will handle real life data after deployment.

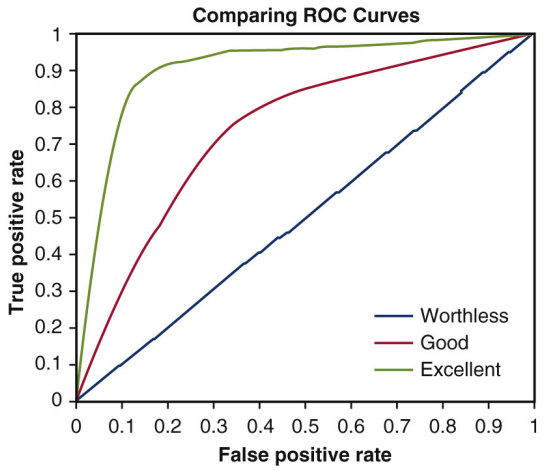

Figure 7: ROC curve comparison

Receiver Operating Characteristic Curves (ROC) provides an excellent evaluation metric for binary classifiers, like the first two algorithms recommended. ROC curves maps out the True Positive rate against the False Positive rate, as seen in figure 7, allowing the user to monitor the trade-off between

sensitivity (true positive rate) and specificity (false positive rate) of results.

Confusion Matrix is a table used to show the performance of classification algorithms, this is done through test data were the true values are already known. The table shows areas of weakness for the algorithm, by displaying which results were predicted correctly or incorrectly through percentages in a matrix, as seen in figure 8. This evaluation metric is very visually informative and can be a great aid in finding under performing areas of the algorithm.

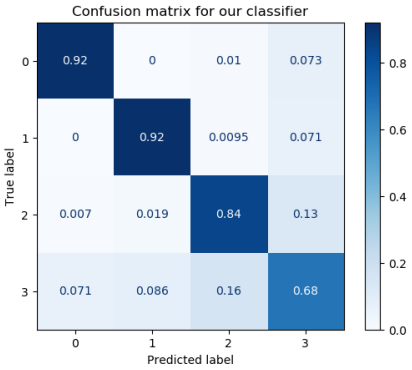

Figure 8: Confusion Matrix Example

Precision recall is another evaluation metric, this time focusing on mapping the positive predictions made from the dataset, aiming for a PR rate of 100%. Precision recall is excellent for training algorithms, but does not hold up after deployment.

6 CONCLUSION

This report effectively gives "Company Redacted" everything needed to understand Machine Learning Algorithms and choose the perfect one to monitor their network traffic. Once selected there should be no problem implementing the Algorithm into the company using the development stages with data pipeline included, and evaluation of the model through the recommended metrics will provide confidence in the solution.

Overall, with these changes put place "Company Redacted" should be assured that their Machine Learning Algorithm is mitigating any risk of Cyber-attacks, keeping their company, reputation and customers safe.

7 REFERENCES

Bakos, Y.J. 2010. Decision Tree Classifier. [online] Colorado School of Mines. Available at: http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/lguo/decisionTree.html [Accessed 16 April 2021]

Chakure, A. 2019. Decision Tree Classification. [online] Medium. Available at: https://medium.com/swlh/decision-tree-classification-de64fc4d5aac [Accessed 16 April 2021]

Data-Driven Science. 2020. 7 Stages of Machine Learning — A Framework. [online] Medium. Available at: https://medium.com/@datadrivenscience/7-stages-of-machine-learning-a-framework-33d39065e2c9 [Accessed 17 April 2021]

Dua, M. & Kunal. Machine Learning Approach to IDS: A Comprehensive Review. *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*. pp. 117-121.

Ellrodt, L.R., Fields, T.L., Freeman, I.C., Haigler, A.J. & Schmeelk, S.E. 2018. What are they Researching? Examining Industry-Based Doctoral Dissertation Research through the Lens of Machine Learning. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. pp. 1338-1340.

Foote, K.D. 2019. A Brief History of Machine Learning. [online] Dataversity. Available at: https://www.dataversity.net/a-brief-history-of-machine-learning/# [Accessed 15 April 2021]

Garbade, M.J. 2018. Understanding K-means Clustering in Machine Learning. [online] towards data science. Availabe at: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1 [Accessed 16 April 2021]

Gill, N.S. 2020. Machine Learning Pipeline Deployment and Architecture. [online] Xenon Stack. Available at: https://www.xenonstack.com/blog/machine-learning-pipeline/ [Accessed 17 April 2021]

Google Developer. 2021. k-Means Advantages and Disadvantages. [online] Google Developer. Availabe at: https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages [Accessed 16 April 2021]

Hao, K. 2018. What is machine learning?. [online] MIT Technology Review. Available at: https://www.technologyreview.com/2018/11/17/103781/what-is-machine-learning-we-drew-you-another-flowchart/ [Accessed 15 April 2021]

Henrique, A. 2019. Clustering with K-means: simple yet powerful. [online] Medium. Available at: https://medium.com/@alexandre.hsd/everything-you-need-to-know-about-clustering-with-k-means-722f743ef1c4 [Accessed 16 April 2021]

Jain, T. 2021. K-Means Clustering. [online] Medium Data Driven Investor. Available at: https://medium.datadriveninvestor.com/k-means-clustering-ac3ff1d3463d [Accessed 16 April 2021]

Java T Point. 2021. Support Vector Machine Algorithm. [online] Java T Point. Available at: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm [Accessed 16 April 2021]

Jordan, J. 2017. Hyperparameter tuning for machine learning models. [online] Blog. Available at: https://www.jeremyjordan.me/hyperparameter-tuning/ [Accessed 17 April 2021]

Kumar, D. 2019. Top 4 advantages and disadvantages of Support Vector Machine or SVM. [online] Medium Blog. Available at: https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107 [Accessed 16 April 2021]

Lessmann, S. 2004. Solving Imbalanced Classification Problems with Support Vector Machines. *Proceedings of the International Conference on Artificial Intelligence, IC-AI '04, June 21-24, 2004, Las Vegas, Nevada, USA, Volume 1.* pp. 214-220.

Mayo, M. 2018. Frameworks for Approaching the Machine Learning Process. [online] KDnuggets. Available at: https://www.kdnuggets.com/2018/05/general-approaches-machine-learning-process.html [Accessed 17 April 2021]

Monkey Learn. 2021. An Introduction to Support Vector Machines (SVM). [online] Monkey Learn Blog. Available at: https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/ [Accessed 16 April 2021]

O'Malley, J. 2018. Captcha if you can: how you've been training AI for years without realising it. [online] techradar. Available at: https://www.techradar.com/uk/news/captcha-if-you-can-how-youve-been-training-ai-for-years-without-realising-it [Accessed 15 April 2021]

Potentia. 2021. What Is Machine Learning: Definition, Types, Applications And Examples. [online] Potentia Analytics. Available at: https://www.potentiaco.com/what-is-machine-learning-definition-types-applications-and-examples/ [Accessed 15 April 2021]

Ray, S. 2017. Understanding Support Vector Machine(SVM) algorithm from examples (along with code). [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ [Accessed 16 April 2021]

Scikit Learn. 2021. Decision Trees. [online] Scikit Learn. Available at: https://scikit-learn.org/stable/modules/tree.html [Accessed 16 April 2021]

Sharma, S. 2017. What the Hell is Perceptron?. [online] towards data science. Available at: https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53 [Accessed 15 April 2021]