CURTIN UNIVERSITY

PROJECT 2

# An Investigation on Preventing Cyber Security Attacks using Artificial Intelligence

*Author:*
Joannes Karmel
GANDAHUSADA

*Unit:*
COMP6001

January 11, 2024

# Contents

# Chapter 1

# Executive Summary

Cyber Intrusion Detection is an important consideration for businesses hosting online services or data online, as the compromise of such entities will yield massive financial loss and confidential data loss. This paper explored several feature reduction techniques and machine learning models to explore which were suitable for the task for cyber intrusion detection. It was found that using the Gini importance metric calculated by the Random Forest model that the feature with the largest impact on cyber intrusion detection is the largest IP packet length. Regarding the models, k Nearest Neighbours (kNN) was found to be the most optimal for the topic of cyber intrusion detection due to its consistently high performance and fast training time.

# Chapter 2

# Introduction

In today's age of digital technology and the internet, data has shifted from the physical forms of papers and books to being stored on computers, optical disks, or flash drives, and now being stored on the cloud with the recent developments of cloud technology. With the rise in popularity of such technology, digital data has become integral to business operations. Banks rely on the internet to store confidential customer details, social media companies store lots of user data including interests and online activities, and e-commerce companies store keep track of millions of customer purchases, and can advertise specific products depending on their purchasing history. On top of such important data, these corporations also rely on the internet to host web applications to allow customers to use their services. People are able to access their banking information, communicate with others across the world or purchase items from the convenience of their homes with the internet. According to *Statistica.com*, 11.28 billion devices were detected to be connected to the internet, and the number is forecasted to more than double to 30 billion devices in 2030 (Statistica.com, n.d.). Such statistics show the dependence people have on the internet in order to access online data and services, and it is due to their importance that computer hackers are drawn to compromise the services with malicious intent.

Cyber attacks can be defined as malicious attempts by an individual or group of hackers to bypass the securities of a computer system or network. Companies and organisations can face serious damages when under a cyber attack, not limited to but including money lost due to ransom or company procedures being running and confidential data being leaked. Such attacks are problematic, as they pose many threats to the financial and data security of corporations and are also a challenge to detect and prevent from happening. While most attacks have been targeted towards organisations, some hacker groups can go as far as attack government bodies. Hathaway's paper documents many historical cyber attacks involving government bodies, such as a computer "worm" compromising Iran's nuclear program in 2010, a distributed denial of service (DDoS) attack shutting down all of Burma's internet following an election, or even China's military launching a cyber attack program on a Falun Gong website in 2011 (Hathaway et al., 2012). These attacks had gotten more common and larger in scale, which have sparked up many discussions, laws and policies protecting parties from cyber attacks.

While corporations are quickly learning the methods that hackers use to attack their network, hackers are also rapidly evolving the ways in which their attacks are conducted. Hackers have moved on from simple methods of attacks such as Structured Query Language (SQL) injection attacks to more structured and coordinated attacks that involve much more computing power and complexity.

There are different attacks that hackers have been using in modern times. Cyber attacks include but are not limited to:

- Denial of Service (DoS) attacks shut down an online organisation's online services and data, ensuring that they are inaccessible to its users or own employees(Li and Liu, 2021). DoS attacks generally fall under two categories, those categories being vulnerable attacks - attacks where malformed packets attack a weakness with the organisation's software, causing excessive memory consumption, and flooding attacks, where large and continuous amounts of traffic are sent to a network to bottleneck and overload it (Carl et al., 2006).

- Distributed Denial of Service (DDoS) attacks are a more intensified version of the attack that utilises multiple computers or machines instead of one to flood the network's traffic. The individual or group behind such an attack would send an *execute* message to its network of computers via a control master program, and at the receival of this message, all computers would carry out a DoS dattack at the same time on the victim's network (Lau et al., 2000).

- Brute Force attacks aim to obtain confidential credentials such as usernames and passwords by attempting every single possible credential combination. While such attacks are not as common in today's cyber climate due to organisations implementing lockout systems that prevent subsequent incorrect login attempts, there are still many organisations with low cyber security awareness that fall prey to such attacks. According to research conducted by Kaspersky during the height of the Covid pandemic, the number of brute force attacks skyrocketed. Using an example from the study, USA saw roughly 200,000 brute force attacks per day at the start of March, yet by the end of April, this number would peak at 1.4 million attacks per day (Galov, 2020). Such statistics reinforce the argument of why it is important for organisations to set up systems that can detect even the most simple attacks.

- Infiltration attacks happen when an attacker already has access privileges to a computer in the network through a security exploit. This then opens a backdoor to the victim's network, and lets the attacker conduct powerful attacks onto the network through the compromised computer, such as IP sweeps and full port scans (Rahman, Al-Saggaf, and Zia, 2020).

- Botnet attacks involve hackers taking control of multiple computers, commonly by injecting trojan horses or viruses into computers and seizing control. These computers are labelled as bots or zombies, and can be used to infect other computers to build a bot army. Once enough computers are under control, they can then be used to carry out attacks. Corporations should be extremely wary of this attack, as any computers can be infected as long as they are connected to the internet (Zhang et al., 2011). A survey in 2009 showed that out of all spam emails sent, 83.2% of those emails were sent by botnets.

- Structured Query Language (SQL) Injection is an older form of attack that sees hackers injecting malicious SQL code in the form of web inputs to get access to confidential data. If successful, hackers will be able to edit or delete the sensitive data stored in the databases. Compromises to sensitive data can lead to heavy consequences for the users and the organisation, such as identity loss or loss of confidential information (Halfond, Viegas, Orso, et al., 2006). However as this is an older form of attack, it is less seen in modern times.

Companies have implemented various intrusion detection systems (IDS) to prevent and mitigate such attacks. These systems monitor the activities and events happening on a computer network or system (Liao et al., 2013). Should there be an anomaly of an event, such as a sudden surge of traffic, the IDS may choose to flag the activity as an attack and perform appropriate measures to stop such attack. This paper will look at anomaly-based methods of intrusion detection, which take a look at any activities that are an anomaly compared to the rest. AIDS methods can be broken into 3 categories, namely statistical-based methods which include time-series models, knowledge-based models which include finite state machines, and machine learning models, which will be investigated in this paper (Khraisat et al., 2019).

This paper will train different machine learning models seen in literature on the cyber attacks mentioned above. Additionally, to expand on previous studies identifying the AI methods for cyber intrusion detection, this investigation will additionally focus on the most important variables which can be used for machine learning. To achieve these goals, machine learning methods for feature selection and classification will be used to propose an optimal IDS system.

# Chapter 3

# Literature Review

The literature review will focus on the machine learning methods which have seen the most success in intrusion detection, as well as the feature selection methods and features which have the largest impact on the predictions. A focus on reducing the number of features is important for a study like this, as not only will organisations be aware of the main indicators of a cyber attack, but also that appropriate feature reduction will also result in better model performances in both time and accuracy. In a study from 2005, the term *The Curse of Dimensionality* is described as the unfortunate circumstances that can occur to the behaviour and performances of models when trained on data with high dimensions (Verleysen and François, 2005). Such consequences include needing exponentially more samples of data with each added dimension for a model to successfully develop, which in turn will result in exponentially longer training times for models.

A machine learning model versatile for different data sets is the XGBoost Decision Tree model. This model utilises gradient-tree boosting, a decision tree algorithm which grows each tree sequentially until the model stops improving (James et al., 2021). A study in 2020 used an XGBoost model to classify different intrusions with the help of a principal component analysis (PCA) for dimensionality reduction, and a Firefly algorithm to optimise the training time of the model(Bhattacharya et al., 2020). However for the sake of this paper, we will only be investigating the PCA method, a method which is able to reduce the dimensions of the feature-set by creating a new feature-set, where each feature is a linear combination with maximum variance of the existing variables, meaning that only a few variables from the new feature-set are required to explain most of the variance of the old features Maćkiewicz and Ratajczak, 1993. The paper also compared this model to the k-Nearest Neighbours (kNN) model, Naive Bayes, Random Forest, and Support Vector Machine (SVM) model. Results show that all of these models had little to no improvement when training data was transformed with PCA. However, the benefit of using PCA was seen in the significant reduction of training time, with the Random Forest model seeing the biggest time reduction at 22%. The kNN, Naive Bayes, Random Forest, and SVM models scoring 91.3%, 76.8%, 91.6%, and 84.4%, while the XGBoost model had the highest sensitivity of 93.1%. This paper's extensive research and comparison of metrics make it easy to understand that while different architectures may improve the performance of the models, the training time of the model is just as important to consider. And in such cases, dimension reduction techniques such as PCA are useful.

Another model to be investigated is the Naive Bayes model, a powerful probabilistic algorithm that uses Bayes' rule with the naive assumption that each observation is independent to classify the data (Webb, Keogh, and Miikkulainen, 2010). This model is powerful, as the violation of the naive assumption usually has no effect

on its performance. A paper compared a Naive Bayes model and another using decision trees, with the help of k-Means clustering to select the most optimal features (Bagui et al., 2019). It was found that the Naive-Bayes method saw an improvement when using feature selection. This paper not only sheds light on the applicability of the Naive Bayes model on data but also the method of using k-Means clustering to obtain the most relevant features. The data frame is to be transposed so that each row represents a feature and is clustered into $k$ clusters after being normalised (Ismi, Panchoo, and Murinto, 2016). The most important features were found near the centre of each cluster, while the irrelevant features will be a great distance away from the other features in the same cluster, calculated using the sum squared difference.

A comparable light-weight machine-learning model to the Naive Bayes model is the k-Nearest Neighbors (kNN) model, which classifies unlabelled data with the majority class of the k-nearest datapoints (Zhang, 2016). A study in 2020 reviewed the kNN and Naive-Bayes models on their performance on DDoS attack detection, and found them to have great performances, with the kNN model achieving a 96.42% accuracy on textitKDDCup-'99', a data set that included various intrusions in a military environment, and the Naives Bayes model achieving 93.95% on the same data set.

Neural Networks are also a popular method of detecting anomalies in a cyber network. The structure of the Neural Networks can be best compared to the nervous system of a human being. Anderson compares this best, as the cells of a nervous system influence each other by constantly sending information to each other, whereas neural networks pass data through layers of nodes (Anderson, 1995). Neural networks are excellent as they are compatible with complex, non-linear data (Livingstone, Manallack, and Tetko, 1997). While some complications come with training Neural Network models such as overfitting, or the computational and time resources required, these can be avoided by correctly selecting architecture, optimizer, and training parameters. A study conducted by Vigneswaran reviewed Dense Neural Networks (DNN), a form of Neural Network composed of dense layers which receives all the inputs of the neurons from the previous layer. It was found that a DNN with three layers yields the best performance by a thin margin, measured by accuracy, precision, recall, and f1-score. The data used in this investigation is the *KDDCup-'99'* data set, which was created in 1998 for the purpose of studying intrusion detection. It included various intrusions in a military environment.

What can be gotten out of these papers is a myriad of feature-selection techniques and models that seem to classify intrusions well. What can be difficult from reading these papers is gauging which models are better performing than others, as variables such as data set, the study scope and year published differ from paper to paper. While there are comparisons of models in each paper, it can still be confusing, as many papers' results show that their studied models were the most optimal when compared to some selected other models. As such, these methods and models will be examined in this study using the same data source.

*Feature Reduction*

1. PCA - due to its proven effectiveness over many different papers and studies
2. Random Forest Feature Importance

3. k-Means Clustering Feature Selection - due to its light-weight nature

*Models*

1. Random Forest
2. Naive Bayes
3. kNN
4. DNN

# Chapter 4

# Methodology

## 4.1 Data

For this investigation, data manipulation and model creation was done using the *Python* programming language in a *Jupyter* Notebook, with machine learning classes from *sklearn* being imported and used.

The data used is from a network intrusion detection system data set collected by The University of Queensland [1], with each row being a particular activity recorded on a cyber network, whether it be benign or malicious. The data set this paper is using, *NF-CSE-CIC-IDS2018-v2*, has near 19 million rows of data, each row being a recorded activity where packets were transferred.

In this data set, there were a total of 45 columns. However, many of these were categorical variables, such as the ports of the connections and identification strings. After removing these variables in preparation to conduct machine learning, 32 meaningful columns were left in the data set, 1 being the response class variable and the rest being numerical data. Each column was ensured to have a non-zero variance, as a non-zero variance will not contribute to any model.

The myriad of attacks were categorised into 7 different types of attacks, labelled benign (non-attack), bruteforce, Bot, DDoS, DoS, Inflitration, and web attacks. There are roughly 16.6 million rows of benign data (88%) while web attacks only has 3502 rows of data, less than 0.02% of the data. The abundant amount of data as well as the extremely skewed categorical proportions presented issues during investigation. Using all the data for training is impossible due to the computational limits, and as such, downsampling the data would be logical. However, retaining the current proportion of data is also impossible due to the heavy skew in category ratios, making it hard to fit data with scarce class values. As the data set itself is not an accurate representation of the proportion of benign activity and cyber intrusions, it was decided that 3500 samples of each class value were sampled for this investigation, with a 90:10 ratio to divide each class for training to testing data. While not an accurate population representation, works fine for the goal of this investigation - to find optimal machine learning models that can recognise cyber intrusions.

[1] Data can be downloaded from - https://staff.itee.uq.edu.au/marius/NIDS_data sets/ - accessed 1 May, 2023

| Column | Description |
|---|---|
| Attack | String category of intrusion type |
| IN_BYTES | Number of incoming bytes |
| IN_PKTS | Number of incoming packets |
| OUT_BY | Number of outgoing bytes |
| OUT_PKTS | Number of outgoing bytes |
| TCP_FLAGS | Total TCP flags |
| CLIENT_TCP_FLAGS | Total client TCP flags |
| SERVER_TCP_FLAGS | Total server TCP flags |
| FLOW_DURATION_MILISECONDS | Flow duration measured by miliseconds |
| DURATION_IN | Client to server stream duration measured by miliseconds |
| DURATION_OUT | Server to client stream duration measured by miliseconds |
| LONGEST_FLOW_PKT | Longest packet of the flow in bytes |
| SHORTEST_FLOW_PKT | Shortest packet of the flow in bytes |
| MIN_IP_PKT_LEN | Smallest flow IP packet |
| MAX_IP_PKT_LEN | Largest flow IP packet |
| SRC_TO_DST_SECOND_BYTES | Source to destination bytes per sec |
| RETRANSMITTED_IN_BYTES | Retransmitted TCP flow bytes from source to destination |
| RETRANSMITTED_IN_PKTS | Retransmitted TCP flow packets from source to destination |
| RETRANSMITTED_OUT_BYTES | Retransmitted TCP flow packets from destination to source |
| RETRANSMITTED_OUT_PKTS | Retransmitted TCP flow packets from source to destination |
| SRC_TO_DST_AVG_THROUGHPUT | Source to destination average throughput |
| DST_TO_SRC_AVG_THROUGHPUT | Destination to source average throughput |
| NUM_PKTS_UP_TO_128_BYTES | Packets with IP size less than 128 bytes |
| NUM_PKTS_256_TO_512_BYTES | Packets with IP size more than 128, less than 256 bytes |
| NUM_PKTS_256_TO_512_BYTES | Packets with IP size more than 256, less than 512 bytes |
| NUM_PKTS_512_TO_1024_BYTES | Packets with IP size more than 512, less than 1024 bytes |
| UM_PKTS_1024_TO_1514_BYTES | Packets with IP size more than 1024, less than 1514 bytes |
| TCP_WIN_MAX_IN | Maximum TCP window source to destination |
| TCP_WIN_MAX_OUT | Maximum TCP window destination to source |

**Figure 4.1:** Brief description of all 32 column

## 4.2 Performance Metrics/Model Evaluation

To decide on the most optimal models, each model generated was compared using the following metrics:

1. Micro F1 Score - This refers to the global average F1 score, which takes into account the count of true positive (TP), false positive (FP) and false negative (FN) predictions. The Equation for Micro F1 Score is as follows:

   $$MicroF1 = \frac{TP}{TP+\frac{1}{2}(FP+FN)}$$

   As each class is balanced in this investigation, the Micro F1 score is essentially the same calculation as accuracy.

2. Training Time - This refers to the number of seconds it takes to train the model. The training time is just as important as the F1 score, as a model that takes very long to train will not be useable for users or other researchers.

   Each model except for DNN has been optimised via a sweep of values for one or two parameters as well as cross-validation. This is due to the large computing power needed to optimise the DNN model, which is not possible to have in this investigation. Therefore, the DNN model is using the supposedly best layout as described in Anderson's paper (Livingstone, Manallack, and Tetko, 1997), with 3 dense layers of neuron sizes 1024, 768 and 524 respectively.

## 4.3 Feature Reduction

The methods of feature reduction were explored via dimension reduction and feature selection. It was a goal in this paper to identify whether feature reduction or dimension reduction would be more suitable for an intrusion detection system. As such, three sets of features were compared in this investigation: the original feature-set containing all 31 numerical features, a feature-set created through PCA, and a feature-set selected with Random Forests and k-Means Clustering.

PCA was applied to the entire feature-set after normalisation. 15 principal components were chosen from the output of PCA that made for 90% of the variance of the feature-set.

The Random Forests model was able to rank variables based on their Gini-importance, which measures the drop in Gini-impurity when the node is removed from the model. The higher the Gini-importance, the higher impact the variable has. The algorithm was able to rank the variables below in figure 4.2.

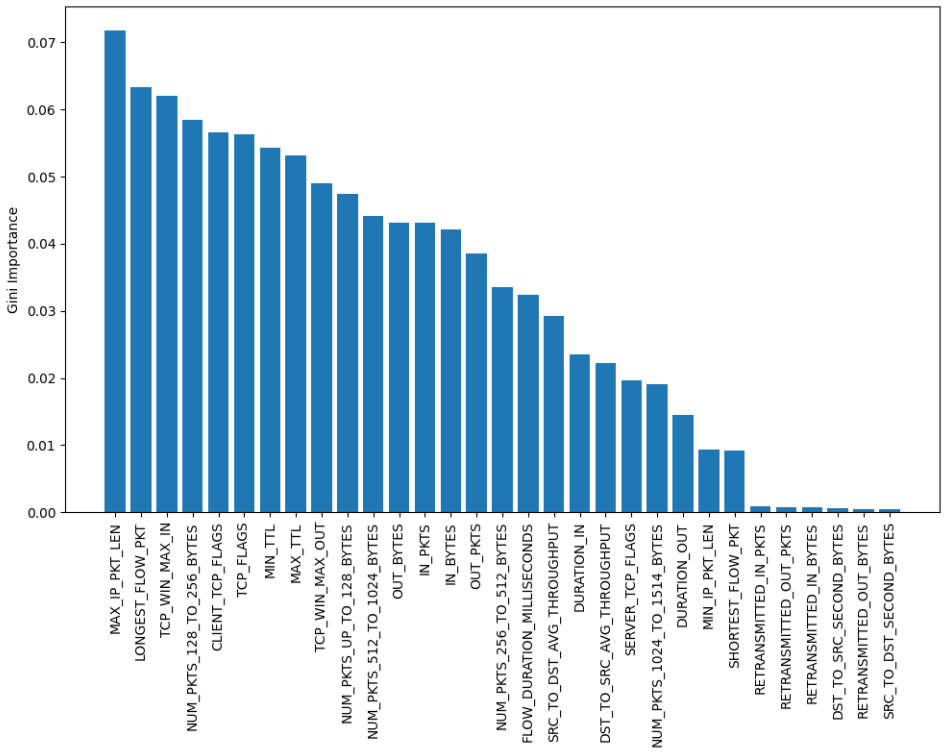


**Figure 4.2:** All 31 feature ranked by Gini importance

Using the k-Means clustering method to select features in Ismi's paper (Ismi, Panchoo, and Murinto, 2016), two clusters were used to group the standardized features. The results were then plotted on two features, seen in figure 4.3.

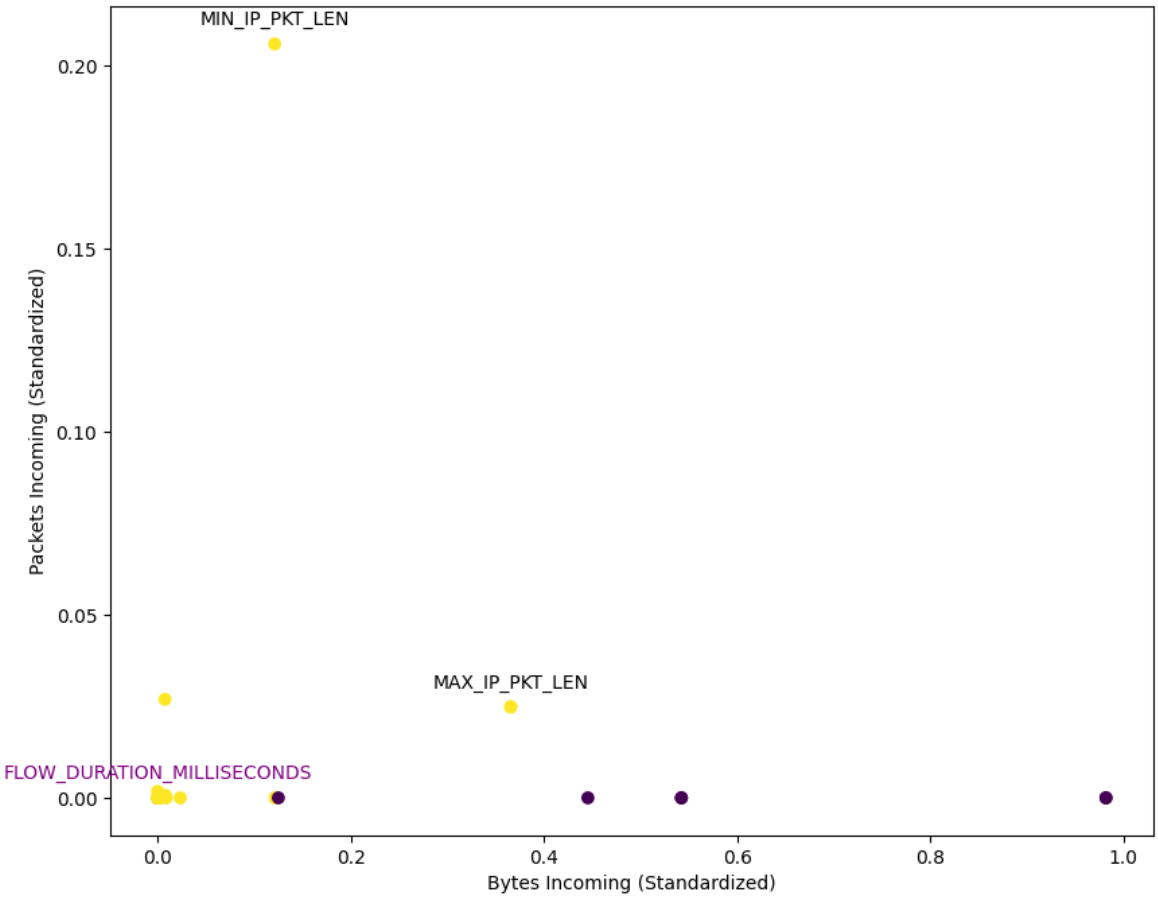


**Figure 4.3:** Features grouped by clusters

It can be clearly seen that in the yellow-labelled cluster, the variable *MIN_IP_PKT_LEN* and *MAX_IP_PKT_LEN* are far away from the other features in the same cluster, meaning they are irrelevant. They each have a sum squared difference of 385.1 and 691.7 respectively across all data points, two of the highest values in the cluster. While not as clear, the variable *FLOW_DURATION_MILLISECONDS* is the furthest away from the other data points in the purple-labelled cluster, having a sum squared difference of 4053.4. These results support the rankings of the random forest model, as these features had a low Gini importance score. Through the combination of a low Gini-importance score and high sum squared value, these variables were dropped from the feature-selected feature set.

1. Flow Duration (ms)
2. Total server TCP Flags
3. Server to client stream duration (ms)
4. Longest flow packet (bytes)
5. Smallest flow IP packet

6. Largest flow IP packet
7. Source to destination bytes per second
8. Retransmission TCP flow bytes from source to destination
9. Source to deestination average throughput
10. Destination to source bytes per second

# Chapter 5

# Benchmarking results

After optimising each model, the results for the F1-Micro metric can be seen in figure 5.1.

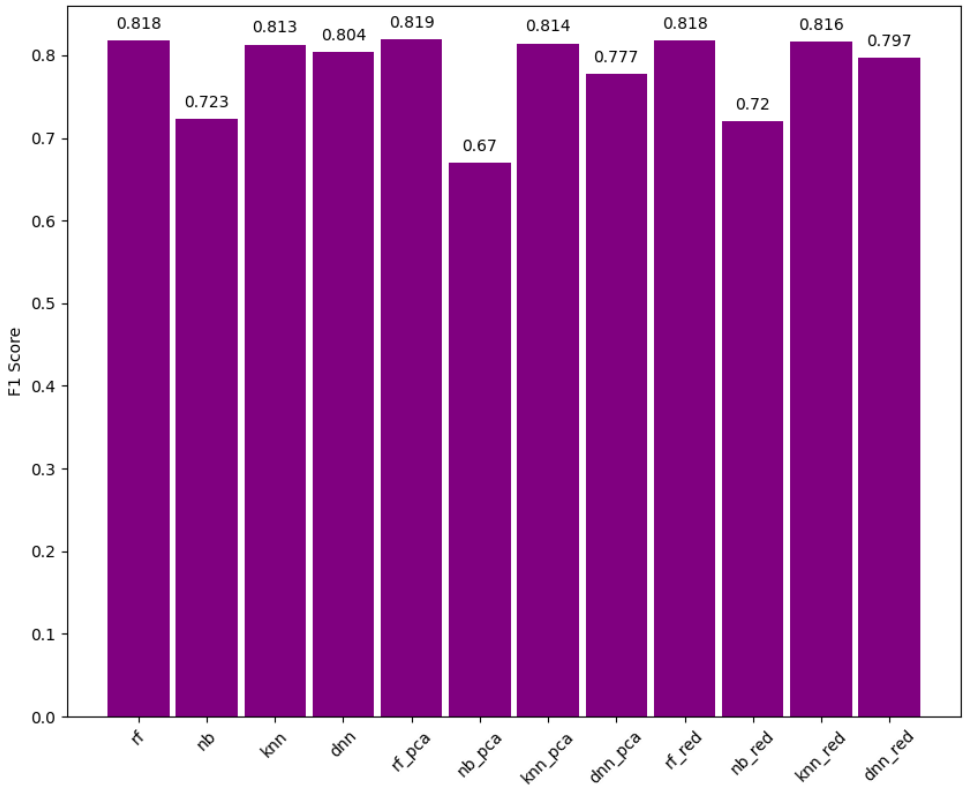


**Figure 5.1:** F1-micro score across each model and feature-set

It can be observed how the Naive Bayes is consistently scoring the lowest, never passing an F1-micro score of 0.73, where as the Random Forest models and kNN models were consistently scoring above 0.8, with the Random Forest models having a small marginal improvement over kNN. The DNN model, despite having more trainable parameters, does not consistently score over 0.8, likely due to the fact that it is not optimised for this dataset. There are no considerable differences in performance accuracy across each feature-set.

Below in figure 5.2 are the times in seconds recorded to fit each model.

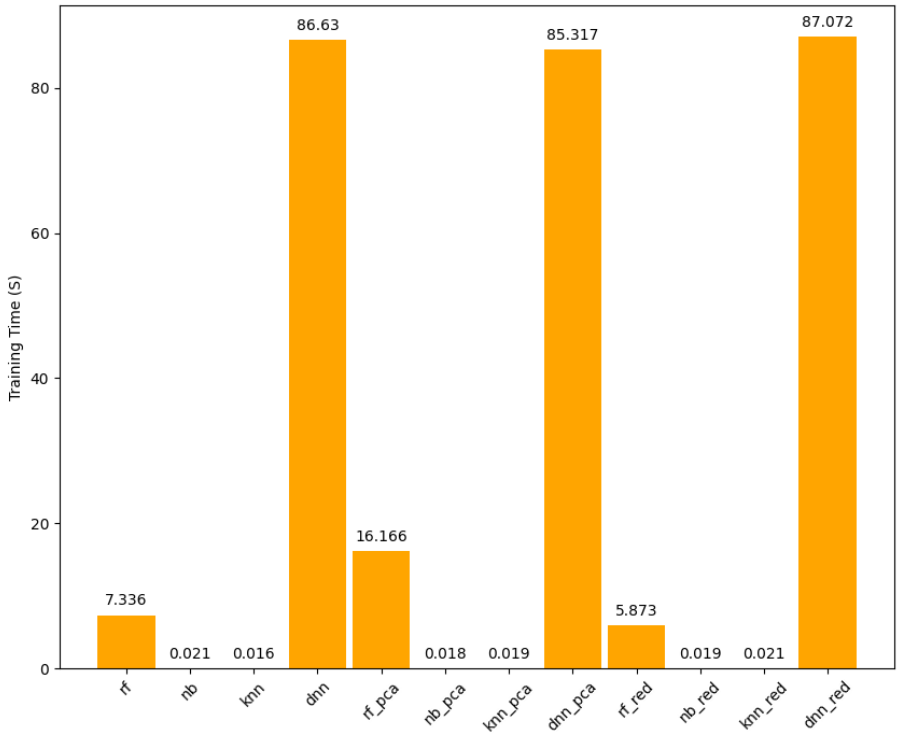


**Figure 5.2:** Training time (s) score across each model and feature-set

DNN models, having the most trainable parameters, take substantially longer than other models to train, which is to be expected. While the performance of a DNN model might be sufficient, the substantial time to train 1 model is enough of a reason for researchers to go with other models, as the usability of a model is almost as important as its accuracy. Random Forests, while nowhere near the time of DNNs, did take a few seconds to train. With a larger dataset, the Random Forest model will struggle more, and may not be usable for researchers with limited computing resources. Naive Bayes and kNN models are the fastest to train, and take almost no time.

# Chapter 6

# AI Ethics

This investigation does not face ethical AI issues. Referring to the 8 main points in AI ethics in Australia, there are only a few situations where AI ethics may potentially be forgotten for the pursuit of better model performance. This section will detail whether the research methods follow the principles of AI ethics or not.

1. **Human, societal, environmental well-being** - this principle is to ensure that the AI systems used are contributing to a good cause for society and are not dangerous. This investigation explores useful methods for the optimizing of intrusion detection models with the necessary data and models, which is helpful for human and societal well-being of in the form of cyber protection and security.

2. **Human-centered values** - this principle refers to the AI systems respecting human rights, culture and life, and not requiring people to abandon humanity's core values to work on the AI. This investigation supports the human-centered values of protection and security, and very much follows this policy.

3. **Fairness** - this principle ensures the AI systems are inclusive for all people and communities. This stays true for this project, there have been no discriminatory motivations in this project.

4. **Reliability and Safety** - This principle ensures the AI systems produce the same responses with the same data and parameters on repeated attempts. This is true for this investigation, which takes advantage of Sklearn's *random_state* parameter to ensure that the experiment is reproduceable for those willing to run the Notebook.

5. **Transparency and Explainability** - This principle ensures users are fully aware of how the AI systems are functioning, and how the output can influence them. As this experiment was done in a Jupyter notebook, users are able to see each step of the way, including how the data is manipulated, how the models are optimised, and how features are deemed useful or not.

6. **Contestability** - This principle ensures users have sufficient time to question the AI's output. This is so that there is enough time to reverse the output if results are considered misleading or wrong. Readers of this report are able to raise questions and concerns about the results on this paper, and depending on the severity, changes can be made to the paper or Jupyter notebook, and republished if needed.

7. **Accountability** - This principle ensures the author's non-anonymity and full responsibility for any mistakes or errors of the AI. The author of this investigation is known and is responsible for any errors and incorrect analysis.

# Chapter 7

# Conclusion

Little change is seen in the performance of the models across the baseline, dimension-reduced and feature-selected data sets. This is in line with the expectations of this investigation, which was that selecting or creating only the necessary variables for a classification task can lead to the same, or improved accuracy as using the full feature-set. Little change was also seen in the training time taken for each model across all data sets. This is likely due to the training data being relatively small because of computer resource limitations for training. Additionally, the results also highlight that having the more complex model is often at times not beneficial. Theoretically while they should perform better when tuned properly, this takes a lot of computing and time resources. The Dense Neural Networks took much longer to train than the other models, and could not be optimised with cross-validation due to computing resource constraints. For researchers with low computing power, DNNs are not recommended. The most optimal model seen in this experiment was the kNN, which consistently performed well across all feature-sets, and took the least amount of time to train. In addition, it is also found that the largest flow of an IP packet is the most crucial feature of determining a cyber attack, which was determined by the Gini importance index by the Random Forest model.

# Bibliography


Anderson, James A (1995). *An introduction to neural networks*. MIT press.

Bagui, Sikha et al. (2019). "Using machine learning techniques to identify rare cyber-attacks on the UNSW-NB15 dataset". In: *Security and Privacy* 2.6, e91.

Bhattacharya, Sweta et al. (2020). "A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU". In: *Electronics* 9.2. ISSN: 2079-9292. DOI: 10.3390/electronics9020219. URL: https://www.mdpi.com/2079-9292/9/2/219.

Carl, G. et al. (2006). "Denial-of-service attack-detection techniques". In: *IEEE Internet Computing* 10.1, pp. 82–89. DOI: 10.1109/MIC.2006.5.

Galov, Dmitry (2020). "Remote spring: the rise of RDP bruteforce attacks". In: Accessed 12 May, 2023. URL: https://securelist.com/remote-spring-the-rise-of-rdp-bruteforce-attacks/96820/.

Halfond, William G, Jeremy Viegas, Alessandro Orso, et al. (2006). "A classification of SQL-injection attacks and countermeasures". In: *Proceedings of the IEEE international symposium on secure software engineering*. Vol. 1. IEEE, pp. 13–15.

Hathaway, Oona A. et al. (2012). "The Law of Cyber-Attack". In: *California Law Review* 100.4, pp. 817–885. ISSN: 00081221. URL: http://www.jstor.org/stable/23249823 (visited on 05/10/2023).

Ismi, Dewi Pramudi, Shireen Panchoo, and Murinto Murinto (2016). "K-means clustering based filter feature selection on high dimensional data". In: *International Journal of Advances in Intelligent Informatics* 2.1, pp. 38–45.

James, Gareth et al. (2021). *An Introduction to Statistical Learning*. Vol. 2. Springer, p. 347.

Khraisat, Ansam et al. (2019). "Survey of intrusion detection systems: techniques, datasets and challenges". In: *Cybersecurity* 2.1, pp. 1–22.

Lau, F. et al. (2000). "Distributed denial of service attacks". In: *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0*. Vol. 3, 2275–2280 vol.3. DOI: 10.1109/ICSMC.2000.886455.

Li, Yuchong and Qinghui Liu (2021). "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments". In: *Energy Reports* 7, pp. 8176–8186.

Liao, Hung-Jen et al. (2013). "Intrusion detection system: A comprehensive review". In: *Journal of Network and Computer Applications* 36.1, pp. 16–24.

Livingstone, David J, David T Manallack, and Igor V Tetko (1997). "Data modelling with neural networks: Advantages and limitations". In: *Journal of computer-aided molecular design* 11, pp. 135–142.

Maćkiewicz, Andrzej and Waldemar Ratajczak (1993). "Principal components analysis (PCA)". In: *Computers  Geosciences* 19.3, pp. 303–342. ISSN: 0098-3004. DOI: https://doi.org/10.1016/0098-3004(93)90090-R. URL: https://www.sciencedirect.com/science/article/pii/009830049390090R.

Rahman, Md Anisur, Yeslam Al-Saggaf, and Tanveer Zia (2020). "A Data Mining Framework to Predict Cyber Attack for Cyber Security". In: *2020 15th IEEE*

*Conference on Industrial Electronics and Applications (ICIEA)*, pp. 207–212. DOI: 10.1109/ICIEA48937.2020.9248225.

Statistica.com (n.d.).

Verleysen, Michel and Damien François (2005). "The curse of dimensionality in data mining and time series prediction". In: *Computational Intelligence and Bioinspired Systems: 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10, 2005. Proceedings 8*. Springer, pp. 758–770.

Webb, Geoffrey I, Eamonn Keogh, and Risto Miikkulainen (2010). "Naïve Bayes." In: *Encyclopedia of machine learning* 15, pp. 713–714.

Zhang, Lei et al. (2011). "A survey on latest botnet attack and defense". In: *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, pp. 53–60.

Zhang, Zhongheng (2016). "Introduction to machine learning: k-nearest neighbors". In: *Annals of translational medicine* 4.11.