Abertay University

# Machine Learning Report

Isaac Basque-Rice

BSc. (Hons.) Ethical Hacking
Abertay University
Dundee, United Kingdom
1901124@abertay.ac.uk

25th April, 2023
2188 Words

# Contents

# List of Figures



# List of Tables



# List of Acronyms

**AI** Artificial Intelligence
**DT** Decision Tree
**DoS** Denial of Service
**HIDS** Host-based IDS
**IDS** Intrusion Detection System
**ML** Machine Learning
**NIDS** Network-based IDS
**RL** Reinforcement Learning

# 1 Introduction

This investigation aims to identify areas in which to improve the security posture of the organisation through the implementation of Machine Learning (ML) into the Intrusion Detection System (IDS) in use currently. To achieve this aim. It is important first that the optimal ML algorithm is identified. As such, this paper will discuss two possible algorithms in relation to their suitability for categorising the network packet data in the manner required by an IDS. The algorithms will be provided with an extensive dataset, for both training and testing, from several sources (Moustafa and Slay 2015, Moustafa and Slay 2016, Moustafa, Creech, and Slay 2017, Moustafa, Slay, and Creech 2019, Sarhan et al. 2021).

This dataset contains information about individual packets captured over a network, including the protocol, port, duration, and attack category. It is the latter that this report is primarily concerned with. Table 1 shows the ten attack categories displayed in the set separated into normal and abnormal, as the algorithm would be expected to identify them.

| Normal | Abnormal |
|---|---|
| Normal | Fuzzers |
| Analysis | Backdoors |
| Generic | Denial of Service (DoS) |
| | Exploits |
| | Reconnaissance |
| | Shellcode |
| | Worms |

Table 1: A list of attack categories, organised into "Normal" for non-concerning categories, and "Abnormal" for concerning ones

After this, this report will aim to outline the process of implementing the appropriate model. This section will include all of the stages of the so-called 'data pipeline', from data ingestion into the model to analysis at the end of the process.

Finally, an analysis of evaluation metrics will be performed. These are quantifiable, observable qualities of data that allow an analyst to compare the accuracy of two or more models against one another with as much fairness as possible.

# 2 Background

## 2.1 Intrusion Detection Systems

The term IDS refers to several software security tools that perform a variety of functions both on a network and on a host. IDSs' primary function is to detect unauthorised access, with the aim of catching malicious actors before they could do any serious damage, or as part of a digital forensic investigation following an attack.

There are two forms of IDS, Network-based IDS (NIDS) and Host-based IDS (HIDS). The latter is designed to monitor event logs, changes to files, and so on, in order to detect potential intrusions onto a specific host. The former, however, is of most interest in the context of this report, as this concerns the possibility of network-based intrusions. These NIDSs monitor network traffic to detect suspicious patterns, such as outbound traffic to a suspicious IP address, many unauthorised login attempts, known attack signatures, and of course unusual traffic patterns more generally (such as a sudden spike in traffic which may be indicative of a DDoS, or malformed network packets).

## 2.2 Machine Learning

Machine Learning refers to a category of Artificial Intelligence (AI), "defined as the capability of a machine to imitate intelligent human behavior [sic.]" (Brown 2021), in the sense that it 'learns' through iteration. This is to improve task accuracy without human input.

Generally speaking, there are four kinds of ML algorithms, these are supervised, semi-supervised, unsupervised, and reinforcement (Wakefield n.d.), these classifiers refer to the method by which the algorithms are trained.

Supervised learning describes algorithms that use labelled datasets. In this way, the algorithms are able to "weigh accuracy and improve with additional data repetition over time." (Corbo 2023). This contrasts with unsupervised approaches, which interpret the data through iterative processes over unlabelled sets. A visual example of the difference is given in Figure 1.

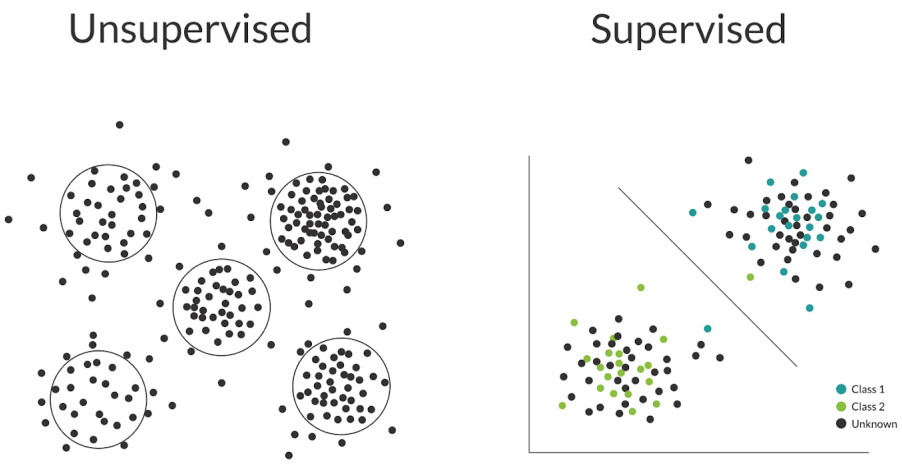


Figure 1: A graphic showing the differences between Unsupervised and Supervised approaches to Machine Learning. Adapted from Wu 2019

.

Semi-supervised models take a hybrid approach, where some data points may be labelled and others may be unlabelled, through which a model can make predictions around new examples (Brownlee 2021).

Reinforcement Learning (RL), however, is different. It departs from the other forms of training in that it does not focus on labelling of data points but rather the behaviour of the algorithm towards these data points. It "[rewards] desired behaviors and/or [punishes] undesired ones" (Carew n.d.).

The investigation will examine the effectiveness of two ML algorithms, one supervised (Random Forest) and one unsupervised (K-means Clustering), with the aim of covering a range of possibilities.

# 3 Machine Learning Algorithms

## 3.1 K-Means Clustering

K-means Clustering is one example of a potentially useful ML model. This algorithm is 'unsupervised', meaning it can work with given data *without the need for human intervention* (IBM n.d.(b)). Classification of data groupings, also known as 'clusters' are done without response variables, that is, without a specific target 'in mind'.

The 'K' in K-means Clustering refers to the number of clusters the algorithm is attempting to identify. This number, chosen by the developer, determines the number of cluster seeds, also known as centroids, available to the algorithm. The locations of these seeds may also be chosen by the developer, at random, or by the algorithm itself dependent on the locations of the data points and the specific implementation (Wohlenberg 2021). After this, it is simply a process of identifying which data points are closest to any given seed, at which point that data point will be assigned to the cluster associated with the seed (Kaloyanova 2021). The standard formula for performing this action was first described by Hartigan and Wong in 1979 and reads as follows:

$$W(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2$$

Where $x_i$ is an individual data point within a cluster $C_k$, and $\mu_k$ is the mean of all data points in that cluster (Kassambara n.d.). The formula subtracts that mean value from each individual data point in what could be described programmatically as a loop, and then squares the result, before summing up all the results from each step.

Find in Figure 2 an example of a graph created from a K-means Clustering process.

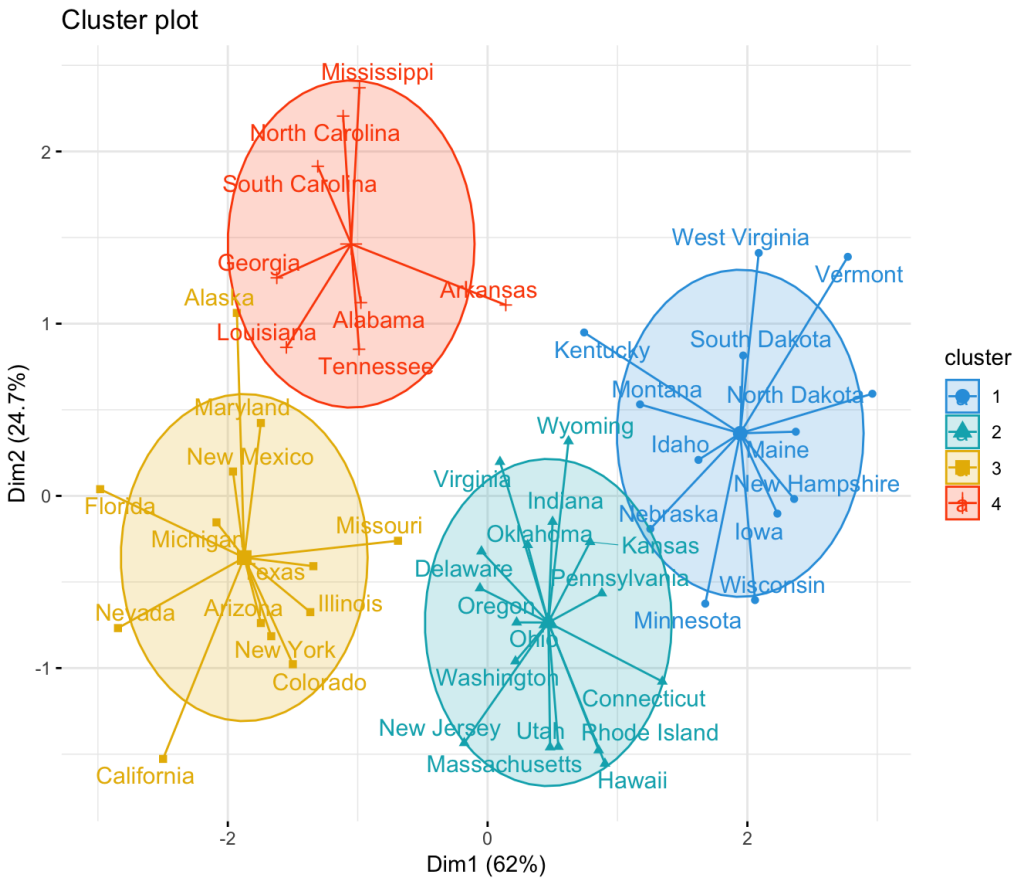


Figure 2: A graph displaying the results of a K-means Cluster algorithm performed on the various states of the USA. Adapted from Kassambara n.d.

.

Regarding the efficacy of the K-means clustering algorithm for the purposes of IDSs, there are a number of positive aspects. One of these is the ease with which services can be scaled up or down at will. As growth is, naturally, an aim of the organisation, a security solution that is able to expand with an organisation is imperative. Additionally, K-means clustering is simple, popular, and easily understood (Education Ecosystem (LEDU) 2018), making it eminently maintainable.

However, there are downsides to this approach that may prohibit the effective use of K-means Clustering. These reasons include the fact that it when implemented fully, gives priority to larger clusters, which may result in false positives or negatives due to certain categories having arbitrarily larger or smaller children. In this sense, it could be described as an inaccurate

algorithm for these purposes.

Most importantly, however, the algorithm sorts the data into its own categories irrespective of what the category is (for examples see Table 1). This means that, if it were to categorise the data by some other characteristic, the utility will be entirely lost. Therefore it may not be an ideal solution for an IDS.

## 3.2 Random Forest

The Random Forest approach uses a supervised algorithm based on 'Decision Trees (DTs)'. These are data structures that facilitate the decision-making process. These structures are similar to flow charts wherein there is a path, beginning at a specific point (the 'Decision Node'), a choice between one of two or more decisions resulting in a 'Chance Node'. This process is repeated until an end state is reached, the 'End Node' (Hillier 2021).

Random Forests are an extension of this concept, proposed in a 2001 paper by Leo Breiman. They employ multiple decision trees to make a 'forest', thereby increasing the accuracy of the model overall by decreasing room for error or biases inherent in using only one structure. Additionally, the algorithm makes use of a technique known as "bagging", also developed by Breiman in 1996. This is the process of sampling data at random by creating several new datasets, training the model on them, and combining the data across all new datasets to make a final prediction. This is (in theory) more accurate (IBM n.d.(a)).

Figure 3 shows this process in simple terms. Here, the Random Forest consists of a start node, X, quantity $b$ trees each built using the bagging method, an output, $CN$ (where $N$ represents the number of trees, and a voter, responsible for making the final decision (Nakahara et al. 2016).

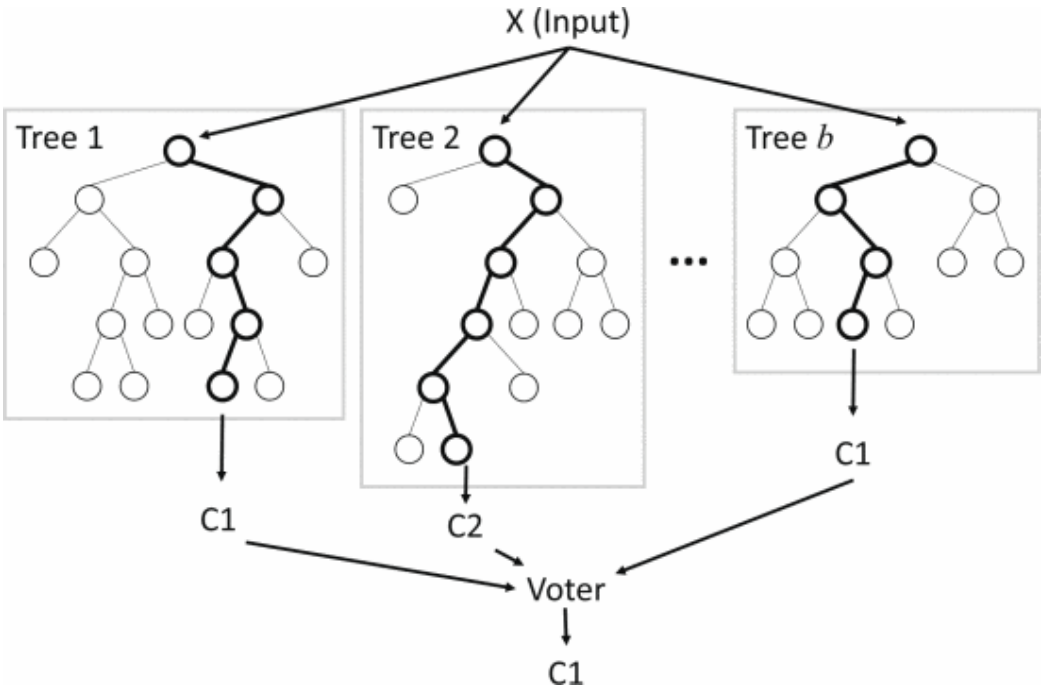


Figure 3: A graphic outlining how a Random Forest algorithm works in relatively simple terms. Adapted from Nakahara et al. 2016

.

In an IDS use case, the Random Forest algorithm has considerable benefits. The foremost of these is its accuracy, as mentioned numerous times, the bagging process provides a significant degree of accuracy in this model. Furthermore, this model reduces the risk of 'overfitting', the phenomenon whereby a model is trained too well on the training data. It begins to internalise noise and random data, resulting in a model too well-suited to the specific conditions it has been fed, and not able to complete general tasks (EliteDataScience 2017). This resistance to overfitting relative to decision trees is because"the averaging of uncorrelated trees lowers the overall variance and prediction error." (IBM n.d.(a)).

Other benefits include the ability to scale up easily, which as mentioned previously will be a great benefit to the organisation as it grows, as well as its speed (allowing it to work well in real-time for the IDS), resilience to outliers (which may be the case for real-world datasets), and flexibility.

There are, unfortunately, downsides to the Random Forest approach. Firstly, the accuracy that Bagging provides comes at the cost of complexity, and therefore interpretability. That is, it is harder to read than a standard

decision tree due to the quantity of data represented and the difficulty determining which variables are important (although this can be made easier programmatically) (Steorts 2015). Additionally, depending on the size of the dataset (which will likely be large in a real-world context), the resources used and the time taken to work will scale up significantly as they need to compute each decision tree's data individually.

## 3.3 Summary Of Algorithms

On balance, therefore, it is clear that the Random Forest approach is the most suitable for this purpose as the limitations of this algorithm do not impede the core intended function of the target implementation (the IDS), and the benefits play well with this target, in that Random Forest will allow for a significantly higher degree of accuracy than K-means Clustering.

K-means Clustering clusters data points based on their proximity to a set of centroids, which can lead to inaccurate classifications. On the other hand, Random Forest, an algorithm based on DTs, utilizes multiple to create a 'forest' and offers higher accuracy due to the decrease in the room for error or biases that come with using only one structure. Additionally, Random Forest offers flexibility in handling large datasets and can be trained to work with various input formats, making it a more versatile and robust algorithm. Therefore, Random Forest is a better choice for applications such as Intrusion Detection Systems, where accuracy is paramount.

# 4 Implementation of the Model

To implement the Random Forest algorithm several steps need to be undertaken. In this section, these steps will be outlined in some detail using a training dataset provided by the organisation as a reference point.

Firstly, the data needs to be 'pre-processed', prepared for use by the algorithm through reformatting. This can be done by removing any extraneous information from the dataset until a smaller set of independent variables remain (e.g. protocol, service, bytes, etc.) alongside the dependent variable, the attack category. Other pre-processing tasks such as splitting the dataset into testing and training have already been done.

Following this, the algorithm needs to be adequately fit to (i.e. train from) the training dataset. To do this we specify first the required number of trees in the Random Forest, and secondly, the 'criterion', which determines the quality of a 'split' in the dataset with the goal of reducing as much randomness as possible.

Next, we make use of the testing dataset to evaluate how successful the model was at predicting malicious intent. After this, various evaluation metrics can be applied to the results to determine how effective this model really was. In-depth information regarding this is in the following section.

Finally, the results of this model must be communicated effectively. In this case, it is likely that these results will be outputted to the IDS that the model has been integrated into, however how this is done specifically is outside of the scope of this paper.

# 5 Analysis of Evaluation Metrics

An Evaluation Metric is the general term for a method of evaluating the performance of a Machine Learning algorithm. In this context the usage of two metrics, an F1 Score and a Confusion Matrix, is appropriate. Note that for the remainder of this section, 'positive' refers to an abnormal value. Additionally, True (real) and False Positives will be referred to as $TP$ and $FP$ respectively, and True and False Negatives will be referred to in turn as $TN$ and $FN$.

A Confusion Matrix is a method of visualising the performance of a model. The four aforementioned outcomes are plotted alongside one another in a table. In practice this matrix would contain the quantity of each of the four metrics the model produces. In an ideal scenario, each of the False metrics would contain 0 entries, however as this is unrealistic, the Matrix can be used to monitor the increase or decrease of $FP$s and $FN$s. Table 2 is a representation of a confusion matrix with stand-in values.

| | | **Actual** | |
|---|---|---|---|
| | | Positive | Negative |
| **Predicted** | Positive | $TP$ | $FP$ |
| | Negative | $FN$ | $TN$ |

Table 2: A so-called 'confusion matrix' which allows us to map whether an outcome is a True/False Positive (T/FP) or a True/False Negative (T/FN)

The F1 Score, alternatively, is a metric intended to be "the harmonic mean of precision and recall" (Korstanje 2021), that is, a weighted average of these two variables. Precision describes the number of values that the model has found to be $TP$, as a percentage of the number of positive values the model found regardless of whether they are true or false. Recall, on the other hand, describes $TP$ as a percentage of all values that are actually positive. The two variables are given through the following formulae.

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

The F1 score grants equal weighting to Precision and Recall, and will increase as either or both of the values increase. This makes it useful in an IDS context as both of these values are crucial to be aware of. The F1 score is given by the following formula.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

# References


Breiman, L. (Aug. 1, 1996). "Bagging Predictors". In: *Machine Learning* 24.2, pp. 123–140. ISSN: 1573-0565. DOI: 10.1007/BF00058655. URL: https://doi.org/10.1007/BF00058655 (visited on Apr. 14, 2023).

— (Oct. 1, 2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324 (visited on Apr. 14, 2023).

Brown, S. (Apr. 21, 2021). *Machine Learning, Explained*. MIT Sloan. URL: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained (visited on Mar. 18, 2023).

Brownlee, J. (Nov. 18, 2016). *What Is a Confusion Matrix in Machine Learning*. Machine Learning Mastery. URL: https://machinelearningmastery.com/confusion-matrix-machine-learning/ (visited on Apr. 22, 2023).

— (Apr. 8, 2021). *What Is Semi-Supervised Learning*. MachineLearningMastery.com. URL: https://machinelearningmastery.com/what-is-semi-supervised-learning/ (visited on Mar. 18, 2023).

Carew, J. M. (n.d.). *What Is Reinforcement Learning? — Definition from TechTarget*. Enterprise AI. URL: https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning (visited on Mar. 19, 2023).

Corbo, A. (Jan. 3, 2023). *What Is Supervised Learning? (Definition, Examples) — Built In*. In collab. with A. Opperman. URL: https://builtin.com/machine-learning/supervised-learning (visited on Mar. 18, 2023).

Education Ecosystem (LEDU) (Sept. 12, 2018). *Understanding K-means Clustering in Machine Learning*. Medium. URL: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1 (visited on Apr. 12, 2023).

EliteDataScience (Sept. 7, 2017). *Overfitting in Machine Learning: What It Is and How to Prevent It*. EliteDataScience. URL: https://elitedatascience.com/overfitting-in-machine-learning (visited on Apr. 15, 2023).

Hartigan, J. A. and Wong, M. A. (Mar. 1, 1979). "A K-Means Clustering Algorithm". In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 28.1, pp. 100–108. ISSN: 0035-9254. DOI: 10.2307/2346830. URL: https://doi.org/10.2307/2346830 (visited on Apr. 13, 2023).

Hillier, W. (July 15, 2021). *What Is a Decision Tree and How Is It Used?* URL: https://careerfoundry.com/en/blog/data-analytics/what-is-a-decision-tree/ (visited on Apr. 14, 2023).

IBM (n.d.[a]). *What Is Random Forest? — IBM*. URL: https://www.ibm.com/topics/random-forest (visited on Apr. 14, 2023).
— (n.d.[b]). *What Is Unsupervised Learning? — IBM*. URL: https://www.ibm.com/topics/unsupervised-learning (visited on Mar. 17, 2023).
Kaloyanova, E. (Oct. 20, 2021). *What Is K-means Clustering?* 365 Data Science. URL: https://365datascience.com/tutorials/python-tutorials/k-means-clustering/ (visited on Mar. 17, 2023).
Kassambara, A. (n.d.). *K-Means Clustering in R: Algorithm and Practical Examples*. Datanovia. URL: https://www.datanovia.com/en/lessons/k-means-clustering-in-r-algorith-and-practical-examples/ (visited on Apr. 13, 2023).
Korstanje, J. (Aug. 31, 2021). *The F1 Score*. Medium. URL: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6 (visited on Apr. 22, 2023).
Moustafa, N., Creech, G., and Slay, J. (2017). "Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models". In: *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*. Ed. by I. Palomares Carrascosa, H. K. Kalutarage, and Y. Huang. Data Analytics. Cham: Springer International Publishing, pp. 127–156. ISBN: 978-3-319-59439-2. DOI: 10.1007/978-3-319-59439-2_5. URL: https://doi.org/10.1007/978-3-319-59439-2_5 (visited on Mar. 14, 2023).
Moustafa, N. and Slay, J. (Nov. 2015). "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)". In: *2015 Military Communications and Information Systems Conference (MilCIS)*. 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6. DOI: 10.1109/MilCIS.2015.7348942.
— (Apr. 4, 2016). "The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Data Set and the Comparison with the KDD99 Data Set". In: *Information Security Journal: A Global Perspective* 25.1-3, pp. 18–31. ISSN: 1939-3555. DOI: 10.1080/19393555.2015.1125974. URL: https://doi.org/10.1080/19393555.2015.1125974 (visited on Mar. 14, 2023).
Moustafa, N., Slay, J., and Creech, G. (Dec. 2019). "Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks". In: *IEEE Transactions on Big Data* 5.4, pp. 481–494. ISSN: 2332-7790. DOI: 10.1109/TBDATA.2017.2715166.
Nakahara, H. et al. (Dec. 2016). "An Acceleration of a Random Forest Classification Using Altera SDK for OpenCL". In: *2016 International Conference on Field-Programmable Technology (FPT)*. 2016 International Con-

ference on Field-Programmable Technology (FPT), pp. 289–292. DOI: 10.1109/FPT.2016.7929555.

Sarhan, M. et al. (2021). "NetFlow Datasets for Machine Learning-based Network Intrusion Detection Systems". In: vol. 371, pp. 117–135. DOI: 10.1007/978-3-030-72802-1_9. arXiv: 2011.09144 [cs]. URL: http://arxiv.org/abs/2011.09144 (visited on Mar. 14, 2023).

Steorts, R. C. (Nov. 2015). "Bagging and Random Forests" (Department of Statistical Science Duke University). URL: https://www2.stat.duke.edu/~rcs46/lectures_2015/random-forest/randomforests.pdf.

Wakefield, K. (n.d.). *A Guide to the Types of Machine Learning Algorithms*. URL: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html (visited on Mar. 18, 2023).

Wohlenberg, J. (June 21, 2021). *3 Versions of K-Means*. Medium. URL: https://towardsdatascience.com/three-versions-of-k-means-cf939b65f4ea (visited on Mar. 17, 2023).

Wu, E. (Nov. 14, 2019). *Supervised vs. Unsupervised ML for Threat Detection — ExtraHop*. URL: https://www.extrahop.com/company/blog/2019/supervised-vs-unsupervised-machine-learning-for-network-threat-detection/ (visited on Mar. 18, 2023).