A Comparative Analysis of Different Machine Learning Techniques in Intrusion Detection Against Evolving Cyberthreats

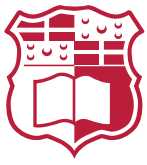
Calvin Azzopardi

Supervisor: Prof. Mark Micallef

Co-Supervisor: Dr Joseph Bugeja

June 2024

Submitted in partial fulfilment of the requirements for the degree of Computing Science.



L-Università ta' Malta
Faculty of Information & Communication Technology

Abstract

The cybersecurity landscape is in constant flux, with new attacks continuously developed by threat actors to circumvent system defences. Consequently, network intrusion detection systems deployed in practice are likely to encounter cyber threats not known to the developer, i.e., unknown attacks. Machine Learning (ML) exhibits great potential to detect such attacks leveraging its capacity to model complex patterns without explicit developer knowledge. Numerous machine learning techniques have been applied in the literature in an effort to achieve this goal, however, many applications of supervised techniques are presented as closed set classification problems neglecting the critical aspect of detecting unknown attacks.

To address this gap, this study assesses and compares the efficacy of several machine learning models in detecting both known and unknown attacks while exploring the underlying relationships between different attack types to facilitate generalisation. The study considers a range of algorithms from three families of ML techniques, namely, traditional supervised techniques, traditional unsupervised techniques and deep unsupervised techniques. Experiments are conducted on the CSE-CIC-IDS2018 dataset, often recognised as the most realistic and up-to-date network intrusion detection dataset currently available. This dataset, characterized by its diversity, scale, and relevance to real-world scenarios, provides a robust foundation for our experiments. This research helps to fill a crucial void in the existing literature by evaluating ML techniques on their ability to detect unknown cyber threats. It also contributes to advancing the understanding of how these techniques can effectively be employed in practical cybersecurity applications.

Our results indicate that supervised models are very effective and generally perform better than unsupervised models on known attacks, however are ineffective against unknown attacks. Unsupervised models, in contrast, are only effective on certain categories of attacks but remain equally effective against both known and unknown attacks.

Acknowledgements

I would like to extend my deepest gratitude to my supervisors Prof. Mark Micallef and Dr Joseph Bugeja, without whom this project would not have been possible. Your insightful feedback, expert guidance and unwavering encouragement have been instrumental to this work and I am truly thankful for your support.

I would also like to express my heartfelt appreciation to the University of Malta for providing me with the knowledge, skills, and resources necessary to carry out this project effectively.

Contents

List of Figures

List of Tables

List of Abbreviations

AIDS Anomaly-based intrusion detection systems.

AUC-ROC Area under the receiver operating characteristic curve.

BiLSTM Bi-directional long short-term memory.

CNN Convolutional neural network.

DBSCAN Density-based spatial clustering of applications with noise.

DL Deep learning.

DoS Denial of service.

DT Decision tree.

FN False negative.

FP False positive.

FPR False positive rate.

FU False unknowns.

GAN Generative adversarial networks.

GB Gradient boosting.

GPU Graphics processing unit.

IDS Intrusion detection systems.

KNN K-nearest neighbours.

LDA Linear discriminant analysis.

ML Machine learning.

NIDS Network intrusion detection systems.

OCSVM One-class support vector machine.

OSS One-side selection.

RAM Random access memory.

RF Random forest.

ROC Receiver operating characteristic.

SAE Shrink auto-encoder.

SIDS Signature-based intrusion detection systems.

SMOTE Synthetic minority oversampling technique.

SSC Sub-space clustering.

SVM Support vector machine.

TN True negative.

TP True positive.

TU True unknowns.

WSL Windows subsystem for linux.

1 Introduction

The ever-evolving landscape of cyberthreats poses a pervasive danger to all facets of society, ranging from commercial enterprises to individuals and government entities. As technology relentlessly advances and more aspects of our lives become digitised, the risks posed by these threats continue to escalate [1].

Cyberthreats refer to malicious acts carried out with the intention of compromising the confidentiality, integrity, or availability of information systems. Cyberthreats can have far-reaching consequences, ranging from the exposure of private information to financial losses to the disruption of critical systems. For instance, the 2015–2016 SWIFT banking hack resulted in the theft of millions of dollars [2, 3] from SWIFT member banks. Another example is the ransomware attack on CommonSpirit Health, which operates 140 hospitals and 2000 patient care sites. The attack led to the disruption of health services and the theft of personal data, and in some cases sensitive health data, of 623,774 patients [4–6]. These examples clearly illustrate the importance of detecting and preventing these cyberthreats.

There are numerous defence options available to help mitigate these threats, prominent among them are Network Intrusion Detection Systems (NIDS). NIDS are a crucial part of any information system's defence that aims to identify and mitigate network threats in real time. Two major classifications exist that characterise the function of intrusion detection systems [7]. Signature-based Intrusion Detection Systems (SIDS) detect attacks by matching sequences of code or commands with those of known attacks. Anomaly-based Intrusion Detection Systems (AIDS) on the other hand, detect threats by recognising deviations from normal, non-malicious traffic [7]. These deviations will be referred to as anomalies and typically represent intrusions, however may sometimes represent false alarms [8].

Designing NIDS suitable for the ever-evolving cyberthreat landscape is no trivial task. NIDS deployed practically are likely to face attacks not known at the time of development, referred to as unknown attacks throughout this study. These could take the form of new deviations of known attacks, new attacks based on known vulnerabilities or zero-day attacks. Zero-day attacks are cyberattacks executed on the basis of an exploit that is not known publicly, and are particularly challenging to detect [9]. Due to their nature, SIDS are ineffective in detecting unknown attacks, as by definition, they cannot exist on any attack signature database [7].

AIDS exhibits promise through its ability to identify unknown attacks [10], particularly with the advent of Machine Learning (ML) techniques. ML techniques, typically considered a branch of AIDS [7], offer enormous potential in anomaly detection as they have the ability to model complex patterns which may not be known

to the developer in advance. These capabilities hold the promise of developing more accurate and effective AIDS.

In the literature, there is no absence of works investigating the use of ML in NIDS [11–16]. This work explores various different approaches including supervised and unsupervised techniques. The works of [11–13, 16] explore supervised techniques and demonstrate remarkable accuracies as high as 100%. These results offer great promise, with near perfect recall and very low false alarm rate, indicating the model detects almost all attacks and seldom flags benign traffic as an attack. Unsupervised techniques have also been researched with results indicating a higher false alarm rate when compared to supervised techniques [17]. ML techniques can also be categorised into traditional techniques and Deep Learning (DL) techniques. The literature explores both branches of ML. Numerous studies focused on supervised techniques have found traditional algorithms to be more effective in the field of NIDS [17, 18].

Some researchers have brought into question the notion that supervised techniques can generalise to unknown attacks [17, 19]. These studies uncover a significant threat to many of the proposed models in the literature, as the evaluation techniques used to measure the efficacy of the model may be misleading. This could lead to a false sense of security that could put information systems at serious risk to unauthorised access and misuse.

Ahmad et al. [20] has noted that many studies focused on supervised learning phrase their works as closed-set classification problems. Closed-set classification problems have all possible classes available in the dataset, whereas open-set classification problems may have other classes not known to the dataset. Network intrusion detection is a dynamic field due to the constant development of new and innovative attacks by threat actors, presenting an open-set classification problem, calling further into question the efficacy of current state-of-the-art models, on unknown attacks.

Unsupervised techniques do not rely on labelled data. It can therefore be hypothesised that they are better suited for open set classification problems as they can detect patterns without prior knowledge of specific classes [21]. This is more akin to true anomaly detection whereby the algorithms focus on detecting deviations form benign traffic as opposed to identifying specific attacks. This notion has been explored previously in the literature by Zoppi et al. [17], with results supporting the hypothesis.

However, unsupervised techniques tend to have significantly higher false alarm rates [17], which hinders their efficacy in practical applications and makes supervised approaches more appealing. Little research exists evaluating the supervised approaches proposed in the literature in the context of an open-set classification problem. Furthermore, the comparative studies found during the literature review do not consider unsupervised DL approaches [17, 19], which may offer an alternative

approach.

1.1 Aims and Objectives

The principal aim of this study will be to investigate the efficacy of supervised and unsupervised techniques proposed in the literature against both known and unknown attacks, comparing the trade-offs between different techniques.

This will take the form of three objectives:

> **Objective 1:** To determine the extent to which current state-of-the-art network intrusion detection models are able to detect both attacks that were present and absent in their training set.

This information is vital to measure the pragmatic performance of these models as NIDS deployed in practical circumstances will face a wide variety of attacks, including unknown attacks.

> **Objective 2:** To investigate the extent to which unsupervised techniques are more or less effective against both known and unknown attacks compared to supervised techniques.

Unsupervised techniques do not rely on labelled attack data in their training. Hence, they are theoretically better adapted to unknown attacks, however may be less accurate on known attacks. This hypothesis is supported by the findings of Zoppi et al. [17] and will be further explored in this study.

> **Objective 3:** To investigate the relationships between specific attacks and attack categories in regard to the generalisation ability of models trained on those attacks.

Certain attacks share similarities that may serve as the basis for models to generalise to these attacks without explicit presence in the training set. These similarities will be explored in this study.

To the best of our knowledge, this study is the first to include DL unsupervised techniques in a comparative analysis considering unknown attacks.

The remainder of this document is divided into four chapters. First, in Chapter 2, we will explore the field of ML-based NIDS, introduce important concepts in the field and discuss similar works in the literature. Secondly, in Chapter 3 we will explain in detail the experiments proposed in this study to evaluate our objectives and the implementation of these experiments. Next, in Chapter 4, we will present the results of our experiments and discuss the interpretations and significance of these results, as well as threats to their validity. Finally, we will summarise and propose future potential research directions in the field.

2 Background

2.1 Network Intrusion Detection

NIDS play a crucial role in the realm of network defence, complementing other techniques such as firewalls and encryption. These sophisticated software systems analyse network traffic to detect malicious behaviour in real time [22]. NIDS can be broadly divided into two categories distinguished by the method of identification employed [7].

SIDS function by identifying specific attack signatures. They typically require access to a signature database, recognising previously documented attacks by correlating sequences of commands or actions to entries on the database. This is highly effective on known threats present in the database, however their efficacy wanes considerably against unknown attacks [7, 22].

AIDS, in contrast, discern attacks by profiling non-malicious behaviour and recognising deviations. This strategy relies on the assumption that malicious traffic inherently differs from benign activity. This approach yields several advantages, primarily, the capability to recognise unknown attacks, absent from any attack signature database and the lack of a need for such a database, which can be laborious to maintain and keep up-to-date. Of course, the approach is not without disadvantages, most notably the high susceptibility to false alarms as any deviation from the expected profile, malicious or not, will be flagged [7, 22].

Numerous methodologies can be employed in the implementation of AIDS, including statistics-based, knowledge-based and ML-based approaches [7]. Statistics-based Intrusion Detection Systems (IDS) rely on statistical models and tests to isolate outliers from normal data, which are then flagged as potential threats. Knowledge-based IDS leverage expert insights or domain-specific rules to flag behaviour that may be malicious. ML-based IDS employ ML models trained to differentiate normal and malicious traffic patterns [7].

Several NIDS implementations exist in practice, for example Snort and Elastic Security are two popular options. Snort is an open-source NIDS that employs signature-based detection to identify threats through predefined rules, while also incorporating some anomaly detection capabilities through preprocessors. Elastic Security is a software tool built on the Elastic Stack, a powerful data analytics and search platform. Elastic Security allows for intrusion detection rules to be defined leveraging data from various sources to identify malicious behaviour. Detection rules offer significant flexibility allowing for a variety of techniques to be employed including signature-based detection, knowledge-based detection and ML-based rules.

2.2 Machine Learning in Network Intrusion Detection

ML exhibits incredible potential in the field owing to its capacity to model complex patterns directly from data without developer knowledge [23]. This could simplify the process of creating NIDS and improve the ability of these systems to detect unknown attacks. The realm of ML encompasses a diverse array of methodologies and algorithms, broadly classified into supervised, unsupervised, and reinforcement learning paradigms [24].

This study will consider supervised and unsupervised techniques. Supervised techniques rely on labelled data and generate predictions by learning to map the feature set to the label based on the training data [24, 25]. Unsupervised techniques, on the other hand, operate without the need for labelled data. They instead learn patterns present in the data by measuring similarity and dissimilarity between data points. These identified patterns can subsequently be used to inform predictions [21, 24].

These families can be further divided into traditional techniques and DL techniques. Traditional techniques typically leverage feature engineering and statistical methods to extract meaningful information from datasets enabling predictions or decisions based on the derived information [26, 27]. In contrast, DL techniques consists of numerous layers of interconnected neurons which each take the sum of their inputs, multiply by their weight and apply an activation function to the output. This is then passed either to the neurons of the next layer or to the final output layer [28].

Supervised techniques tend to be affected negatively by imbalanced datasets, where they may exhibit a bias towards predicting the majority class, leading to less accurate predictions for minority classes [29, 30]. One approach to address this issue, which has been employed extensively in the literature, is Synthetic Minority Oversampling Technique (SMOTE) [11, 12, 31, 32]. SMOTE generates synthetic minority class samples to reduce imbalance by interpolating data from other nearby points. These points are selected using the K-Nearest Neighbours (KNN) algorithm [33]. By increasing the number of minority class samples in the dataset, the effects of class imbalance can be alleviated [31].

2.3 Metrics

Various evaluation metrics exist in the field of ML. In the domain of classification problems, some of the most prevalent metrics used to evaluate ML models include accuracy, precision, recall and F-measure values. These are calculated from the counts

of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) predictions made during testing [34].

Accuracy [34] represents the proportion of correct predictions and is the most common metric used. It can, however, be misleading in datasets with significant class imbalance as if the model performs well on the majority class, it will achieve a high accuracy regardless of the performance on the minority class. In tasks like intrusion detection, where detecting minority classes is crucial, this skewed accuracy can be misleading. It is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Recall [34] represents the proportion of samples of a specific class that were correctly identified by the model. In the context of intrusion detection, recall holds particular importance as it directly influences the system's capability to detect attacks effectively. It is defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Recall-Unk [17] represents the recall value when considering only samples of the unknown class. It is proposed by Zoppi et al. [17] to better define the performance of a model on unknown classes and is considered in this study for the same reasons. Zoppi et al. [17] define True Unknowns (TU) and False Unknowns (FU) in order to calculate Recall-Unk. TU represent correctly identified attack instances that belong to an attack excluded from the training set. FU represent attack instances belonging to an excluded attack that are not identified by the model. Recall-Unk is defined as:

$$\text{Recall-Unk} = \frac{\text{TU}}{\text{TU} + \text{FU}}$$

Precision [34] represents the likelihood a positive prediction made by the model is correct, an important metric in intrusion detection as a low precision indicates a high false alarm rate. It is defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

The F1-measure [34] serves as a unified metric that considers both recall and precision. It is less susceptible to the effects of class imbalance when compared to accuracy. It is defined as:

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

The Receiver Operating Characteristic (ROC) curve [34] is also a common tool

used for evaluating models, with the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) serving as a valuable metric. The curve consists of the False Positive Rate (FPR) rate on the x-axis plotted against the recall on the y-axis, where FPR is defined as:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

2.4 Feature Selection

Feature selection is among the most important components of preprocessing. The aim of feature selection is to simplify the set of features in a dataset and increase the efficacy of the model by selecting the most relevant features and discarding unnecessary features [35]. Features may be relevant, redundant, or irrelevant. Relevant features are those that offer valuable information about the target variable and are vital to enable the model to learn patterns present in the dataset. Redundant features refer to features that also offer valuable information to the model however are superseded by other features that contain the same or more information. Hence, their inclusion will increase the model complexity without any significant improvement to model efficacy. Irrelevant features refer to those that do not contain any information about the target variable. These become a source of noise to the model and should be removed to maximise efficacy and reduce excess model complexity.

Feature selection techniques, similarly to machine learning techniques, can be categorised based on their label requirements. Supervised feature selection techniques require labels and are typically employed for classification and regression tasks, whereas, unsupervised techniques do not require labels and are typically employed for clustering tasks [35].

2.5 Literature Review

There exist several works in the literature focused on the field of ML-based network intrusion detection. This section briefly outlines some significant works in the field.

In 2020, Karatas et al. [11] proposed six ML-based IDS, employing the techniques of KNN, Random Forest (RF), Gradient Boosting (GB), AdaBoost, Decision Tree (DT) and Linear Discriminant Analysis (LDA). They evaluated their models on the CSE-CIC-IDS 2018 dataset. Due to the class imbalance present in this dataset, the authors employ SMOTE. The results indicate the proposed models are very effective with accuracies on the sampled data ranging from 91.18% to 99.35%. The RF algorithm had the highest accuracy on the sampled dataset, however the AdaBoost

algorithm had the highest precision and F1 score which may be more significant given the class imbalance present in the dataset.

Jiang et al. (2020) [12] take a DL approach to the problem. They propose a Convolutional Neural Network (CNN) to extract spatial features, which are then processed through a Bi-directional Long Short-Term Memory (BiLSTM) network to extract temporal features. To handle class imbalance, the authors first apply One-Side Selection (OSS) to reduce noisy samples in the majority class, then increase the number of minority samples through SMOTE. The proposed solution is evaluated on both the NSL-KDD dataset and the UNSW-NB15 dataset, achieving accuracies of 83.58% and 77.16% respectively.

Mighan and Kahani (2021) [13] propose a hybrid approach using a stacked auto-encoder network for latent feature extraction followed by a support vector machine classifier. The proposed model was evaluated on the ISCX2012 and CICIDS2017 datasets achieving accuracies of 90.2% and 99.49% respectively. They concluded that DL feature extraction outperformed other methods. Note, class imbalance was not addressed in this study, despite being present in both datasets used.

Kus et al. (2022) [19] argue that the current evaluation standard in ML-based industrial intrusion detection may create a false sense of security. They argue that it does not assess the model's ability to detect unknown attacks, absent from the training set. They propose a new evaluation methodology to assess the efficacy of ML-based industrial IDS in detecting these unknown attacks. This methodology is applied to three industrial intrusion detection models proposed in a work by Lopez Perez et al. [36]. The results indicate an alarmingly low ability to recognise unseen attacks, with detection rates dropping to between 3.2% and 14.7% for some types of attacks. The authors conclude that the models they tested only learn signatures of attacks and are not performing true anomaly-based intrusion detection. The study focuses primarily on industrial intrusion detection and does not attempt to make any generalisations to other domains of intrusion detection.

A literature review carried out in 2023 by Ahmad et al. [20] reviews many more similar works with a particular focus on zero-day attacks. It supports the argument of Kus et al. [19] stating that many researchers in the field are not currently addressing the existence of zero-day attacks and suggests that future works should not assume all existing attack classifications are present in the dataset.

Pu et al. [14] propose an unsupervised approach, combining Sub-Space Clustering (SSC) and One-Class Support Vector Machine (OCSVM). They evaluate their proposal on the NSL-KDD dataset against three other unsupervised models, specifically, a combination of SSC and Evidence Accumulation, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and K-means clustering. The results indicated the proposed model outperformed the other models.

Cao et al. [15] take an unsupervised DL approach, proposing new regularisers to classical and variational auto-encoders to map normal network samples towards the origin. One class classification models can then be used to detect malicious samples which will be mapped further from the origin. They evaluated this approach on 14 datasets, including, CTU1309, CTU1313, NSL-KDD, and UNSW-NB15. The results indicate the approach is effective on high dimensional and sparse network data.

In 2023, Zoppi et al. [17] carried out a comparison of 47 different machine learning algorithms on 11 different network intrusion datasets. The study includes supervised, unsupervised, DL and meta-learning algorithms. The methodology, like that of Kus et al. [19], involves creating variants of each dataset excluding specific categories in order to simulate the occurrence of unknown attacks. They conclude that supervised classifiers yield a higher accuracy, however, this is significantly degraded in the face of unknown attacks. Unsupervised learners on the other hand are less accurate but experience a less dramatic decline in the face of unknown attacks. They also conclude meta-learners outperform their non-meta-learning counterparts.

Other, less conventional approaches have also been employed in the literature, with some novel approaches having been proposed since this literature review was first carried out. For instance, OpenMax is an extension of the softmax activation function commonly used in DL which has seen application to NIDS [37]. Another possibility that has been explored in a different domain is Generative Adversarial Networks (GAN) dataset enhancement which essentially involves using DL to generate realistic attack samples in the dataset without prior knowledge of the attacks [38]. Biologically inspired algorithms may also offer an interesting approach to the field [39].

2.6 Conclusion

From the above works, we observe that the research community has achieved impressive results in the field of intrusion detection. However, as Kus et al. [19] have discovered, these laboratory results may not accurately reflect reality.

The supervised learning approaches explored above tend not to consider the possibility of unknown attacks and hence may be ineffective in practical applications. As indicated by Zoppi et al. [17], unsupervised techniques may be more effective in detecting unknown attacks, however at the cost of reduced efficacy against known attacks.

3 Methodology

In the previous sections, we have defined the aims of this study, introduced the field of ML-based intrusion detection and reviewed some of the significant works in the literature. In Chapter 1 we define our primary aim of investigating both supervised and unsupervised techniques against both known and unknown attacks, dividing this aim into three objectives. This chapter will discuss in detail the methodology we propose to address these objectives.

The works of Kus et al. [19] and Zoppi et al. [17] shine an illuminating light onto the field, taking into consideration the possibility of unknown attacks and revealing the alarmingly low efficacy supervised models tend to demonstrate on these attacks.

Building upon the foundational research conducted by these authors, this study trains and evaluates models on variants of the dataset excluding certain attacks. More specifically, we generate the variants using the approach proposed by Kus et al. [19]. This approach has been selected over that of Zoppi et al. [17] as it is a superset of the approach taken by Zoppi et al. [17]. Additionally, Kus et al. [19] consider individual attacks, which will provide valuable insights to address Objective 3.

Our methodology can be divided into the five following steps:

1. Selection of state-of-the-art models.
2. Selection of a realistic and diverse dataset.
3. Sampling and preprocessing of the dataset.
4. Generation of training variants which intentionally exclude certain attacks.
5. Replication and evaluation of the selected models on the training variants.

The remainder of this chapter further details these steps.

3.1 Selection of Models

The first step to any comparative analysis is the selection of works for comparison. To address Objective 2, we require at least one supervised model and one unsupervised model. Additionally, we consider both traditional and DL-based unsupervised techniques. Hence, we aim to select at least one model from three families. Namely, these families are supervised traditional ML techniques, unsupervised traditional ML techniques and unsupervised DL techniques. It should be noted that supervised DL techniques are not considered as they have been found to be less effective by Zoppi et al. [17].

The criteria considered when selecting works were the results presented by the paper, the number of citations by other works in the literature, the recency of publication and the specification of a clear replicable methodology. The former three criteria are intended to ensure the works selected represent the state-of-the-art whilst the latter criterion is intended to ensure accurate replication of the work is possible.

Additionally, in the case of supervised techniques, works that consider class imbalance were given preference as this has been identified as a significant issue when applying supervised ML to intrusion detection. This is due to the prevalence of normal, non-malicious traffic in realistic scenarios when compared to the volume of malicious traffic [29, 40].

Based on these criteria, the works of Karatas et al. [11], Pu et al. [14] and Cao et al. [15] were selected from the supervised traditional ML, unsupervised traditional ML and unsupervised DL families respectively. Cao et al. [15] propose 14 models, only one of which was selected for use in this study due to time constraints. The Shrink Auto-encoder (SAE)-OCSVM model was selected as it is among the top performing models evaluated by Cao et al. [15] and has a complete implementation available in a Jupyter notebook in the repository provided by the authors. Karatas et al. [11] and Pu et al. [14] only propose seven models in total, and hence, all of these are included in this study.

3.2 Datasets

Next, we select a dataset for use in our experiments. The aims of this study are intended to better illustrate the performance of our selected models in a realistic environment where unknown attacks may be encountered. Hence, selecting a realistic, modern dataset with sufficient variety of data is a vital step. In addition to the experimental dataset, we also require datasets used by the original authors of the selected models. These are required so that the replicated models can be evaluated on this data using the same metrics presented by the original authors. These metrics can then be compared to those of the original authors to verify the models were replicated correctly. Similarly, the dataset used by Kus et al. [19] in the original proposal of their methodology will also be used to evaluate the methodology and verify it has been replicated correctly.

Our literature review yielded eight datasets that were considered for use in this study. These include the KDD99 [41], NSL-KDD [42], CAIDA-DDOS2007 [43], ISCX2012 [44], UNSW-NB15 [45], CIC-IDS2017 [46], CSE-CIC-IDS2018 [47] and IoT-botnet2020 [48] datasets.

The CSE-CIC-IDS2018 dataset is, to the best of our knowledge, the most

up-to-date, general purpose network intrusion dataset publicly available today. It offers a large volume of data containing diverse attack scenarios and a realistic network environment. It is for these reasons that it has been identified as the most suitable dataset to evaluate and compare the models of this study.

The CSE-CIC-IDS2018 dataset is a collaborative effort between the Communications Security Establishment and the Canadian Institute of Cybersecurity, and serves as a successor to the CIC-IDS2017 dataset which is a smaller, less diverse counterpart. The aim of the project was to produce an up to date, realistic dataset for the evaluation of NIDS. The dataset was generated on infrastructure consisting of a victim network with 420 machines and 30 servers, and an attack network consisting of 50 machines. The machines were divided into departments mimicking a typical corporate network. A typical network environment was simulated using B-profiles, which describe user behaviour through statistical and ML techniques, allowing it to be accurately replicated. The dataset includes a variety of modern attack categories, namely, Denial of Service (DoS) attacks, brute force attacks, code injection, botnet attacks, and infiltration attacks which involve using an exploit on a host inside the network to attack the network. The resulting dataset contains 80 features and approximately 16,000,000 instances [47, 49].

Karatas et al. [11] evaluate their models on the CSE-CIC-IDS2018 dataset. Hence, this dataset can be used to verify the replication of these models in addition to being used as the experimental dataset. In contrast, the dataset is not considered by Pu et al. [14] and Cao et al. [15]. Therefore, another dataset is required to verify the replication of these models. The work of Pu et al. [14] is evaluated only on the NSL-KDD dataset. This dataset is also one of the datasets on which the models proposed by Cao et al. [15] are evaluated. Hence, this dataset is used to verify the correct replication of both these works.

The NSL-KDD dataset is an improvement over its predecessor, the KDD99 dataset. It was created as a benchmark for intrusion detection systems and offers a more balanced class distribution than its predecessor. The dataset has been used extensively in the field of intrusion detection, however, the underlying network data dates back to 1998 [42]. Due to its age, the results may not generalise well to the modern cyber threat landscapes and hence, results on this dataset cannot be considered when addressing the aims of this study. However, this issue does not apply when comparing our results with values generated from the same dataset, rendering this dataset suitable for comparison with the work of the original authors of the unsupervised models selected. This comparison allows us to verify our replications are correct and accurate.

Finally, we require the Gas Pipeline dataset [50] used by Kus et al. [19] in their original proposal of their methodology. The dataset includes a variety of attacks carried

out on a gas pipeline control system in a lab environment. This will be used to verify the replication of the methodology of Kus et al. [19].

3.3 Preprocessing

The methodology of Kus et al. [19], employed in this study, requires each instance in the dataset to be labelled with both the specific attack name and the attack category. The labels provided in the CSE-CIC-IDS2018 dataset can be described as an attack category, however, some labels are not broad enough to for the purposes of this study. An example of this are the DoS attacks, which are labelled according to the specific tool used. Hence, we append an 'attack name' column to the dataset. The values of this column are filled according to Table 2 on the dataset's website [47]. This table specifies the attack schedule used during dataset generation, detailing the start and end times of each attack as well as the labels attributed to each attack. Hence, we use the 'Timestamp' and 'Label' columns to determine the specific attack each instance in the dataset belongs to.

It should be noted, numerous instances in the dataset occur outside the attack time brackets specified in this table. In most cases, only one attack with a particular label occurred each day. Hence, it is assumed that instances with a matching label, occurring outside the specified attack times, but on the same attack day constitute part of that attack. The exception to this are the 'Bot' attacks carried out on 02/03/2018. Two 'Bot' attacks were carried out on the same day, with malicious instances present between the end time of the first attack and the start time of the second. These instances have been labelled as the 'Unknown' attack type.

Next, an 'attack category' column is generated from this new column. The categorisation adopted for this study is shown in Table 3.1. It should be noted that the previously mentioned ambiguous 'Bot' attacks labelled as 'Unknown' are categorised as 'Bot' attacks.

As noted by Ahmad et al. [20], categorising attacks into broad labels, such as DoS, can create inconsistencies and confusion in the literature due to their subjectivity. Hence, the categorisations used in this paper should be interpreted as a shorthand method of referring to the specific attack types included under the category in this study. This shorthand serves to simplify the process of investigating generalisation across attacks that differ significantly in their nature and attacks that share similarities. Conclusions are not be intended to generalise to attack types not present in the dataset that may fall under the same subjective categorisation.

The dataset contains 83 features, however, the features 'Src IP', 'Dst IP' and 'Flow ID' are only present for a small portion of the data. Additionally, these features

Table 3.1 Attack Categorisation

Category	Original Label
Benign	Benign
Brute Force	FTP-BruteForce FTP-BruteForce SSH-BruteForce Brute Force -Web
DoS	DoS attacks-GoldenEye DoS attacks-Slowloris DoS attacks-SlowHTTPTest DoS attacks-Hulk DDoS attacks-LOIC-HTTP DDOS attack-LOIC-UDP DDOS attack-HOIC
Bot	Bot
Infiltration	Infilteration
Injection	SQL Injection Brute Force -XSS

are specific to the network configuration used during dataset generation and hence, tend to be ignored in the literature. Of the remaining 80 features, the following features represent nominal data: 'Protocol', 'Src Port', 'Dst Port', 'Fwd PSH Flags', 'Bwd PSH Flags', 'Fwd URG Flags' and 'Bwd URG Flags'. The 'Src Port' and 'Dst Port' columns are naturally in numeric form whereas the other nominal features are already encoded numerically by the original authors. All flag columns are encoded as binary values indicating the presence of the flag whilst the protocol column contains three unique values: zero, six and seventeen. It has been assumed these values represent three different protocols as the authors do not specify their encodings. All other features in the dataset represent discrete and continuous data, and hence require no encoding.

The CSE-CIC-IDS2018 dataset contains approximately 16,000,000 instances, an enormous number that introduces complexities as it is not feasible to keep the full dataset in memory during processing on the 32 GB of RAM that were available for this project. Hence, the dataset was sampled to 4,519,553 instances. The sampling strategy adopted was random sampling of 28% of the data.

Once the sampled, fully labelled dataset was generated, individual preprocessing pipelines of each replicated study were applied based on the work of their original authors. Some additional steps were added where necessary, including a common set of steps designed for the CSE-CIC-IDS2018 dataset for the unsupervised

models which were originally evaluated on different datasets. We will now briefly outline the steps taken for each model.

The following steps, based on the methodology of Karatas et al. [11], were carried out prior to all experiments.

1. The CSE-CIC-IDS2018 dataset contains data points dated 1970 with illogical values in certain columns, likely caused by overflow errors during dataset generation. All instances of such data points are removed from the sample. Additionally, instances with an 'Unknown' attack type are also removed.
2. Missing values are replaced with zero
3. Infinity values are replaced with the maximum value in the column
4. During replication, the 'Timestamp' column is separated into date and time columns to eliminate non-numeric values. The column is then removed for the experiments as none of the models replicated analyse time series data. Hence, any information present in this column will be valid only under the specific conditions of the attacks present in the dataset, diminishing the realism of the experiments.
5. The columns 'Src IP', 'Dst IP' and 'Flow ID' similarly contain information specific to the attack configuration employed in the generation of the dataset. Hence, these are also removed.
6. Two columns contain negative values, 'Init Fwd Win Byts' and 'Init Bwd Win Byts'. Two additional categorical columns are added, containing a one for negative values and a zero for positive values in the corresponding column.
7. The 'attack category' and 'attack name' columns are encoded into numeric values. For convenience, the zero value is set to represent the benign class whereas all other classes were numbered according to the order in which they appear.
8. The dataset is split into training and testing sets using an 80/20 split and then shuffled.
9. Lastly, SMOTE is employed on the training set to reduce the class imbalance present in the dataset. Minority classes are increased to at least 5% of the data when considering categories, and at least 1% when considering individual attacks.

Following these preprocessing steps, the unsupervised models have additional preprocessing carried out, based on the work of their original authors. Note, the SMOTE step is excluded when processing for unsupervised models as they do not consider minority samples during training.

The following additional steps are applied to the data processed by the SSC-OCSVM model, based on the methodology of Pu et al. [15].

1. The ‘Protocol’ feature, which is the only categorical feature with more than two levels, is converted to one-hot encoding representation.
2. F-test feature selection is employed, selecting the k best features based on the ANOVA F-value [51] of each feature, where k is a hyperparameter.
3. All features are then normalised using z-score standardisation.
4. Finally, malicious samples are removed from the training set, and the training set was reduced to 200,000 samples. This number was selected by progressively increasing the sample size until the performance did not improve any further.

The following additional steps are applied to the data processed by the SAE-OCSVM model, based on the methodology of Cao et al. [15].

1. The negative one values present in the columns ‘Init Fwd Win Byts’ and ‘Init Bwd Win Byts’ are replaced by -0.5 values. This is so that they scale back to their original negative one value following the log operation.
2. All malicious samples are filtered out of the training set, and the training set was reduced to 6734 random samples. This number was selected as it was the number used by the original authors, and our experimentation with different sample sizes failed to yield better results.
3. Next, columns containing large values (larger than 10,000) have one added to them followed by the logarithm operation base two.
4. Finally, each feature is scaled by its maximum absolute value.

3.4 Training Variants

Following preprocessing, the dataset was split into variants according to the methodology of Kus et al. [19]. This consists of filtering the dataset in four phases.

In the first phase, the dataset has particular attack category excluded from the training set. Hence, the excluded attacks simulate unknown attacks and accurately depict model performance on such unknown attacks should it encounter them at runtime. In the second phase, all categories are filtered out except for one, and of course, the benign category. This provides more insight allowing for a better understanding of how attack categories provide valuable knowledge that may generalise to other categories. In the third and fourth phases, these two steps are

repeated focusing on individual attacks instead of attack categories. This provides more insight into the relationships between different individual attacks allowing for generalisation and allows us to analyse generalisation within a category, which would not be possible when considering only categories.

The CSE-CIC-IDS2018 contains 21 individual attacks, which have been divided into five categories. Hence, 52 dataset variants were generated.

3.5 Implementation and Execution

The implementation of this project's evaluation can be found in the project's repository on GitHub [52].

The evaluation of this study is written in Python 3.11. The DT, RF, KNN and LDA models are implemented using the scikit-learn library [53] as was done by the original authors. The GB model is implemented using the XGBoost library [54] to increase the execution speed beyond what is possible with scikit-learn [53]. The work of Pu et al. [14] is replicated using ThunderSVM [55], a Graphics Processing Unit (GPU)-accelerated library that offers a OCSVM implementation. Cao et al. [15] provide the source code of their implementation on GitHub [56]. This implementation employs Tensorflow [57] for the SAE and scikit-learn for the OCSVM model. The SAE implementation was updated to work with Tensorflow 2 and the OCSVM model was replaced with a ThunderSVM [55] model to reduce execution time.

Experiments are executed on Windows Subsystem for Linux (WSL) Ubuntu 22.04 with an Intel I9–10900F, two NVIDIA GeForce RTX3080 GPUs with 10 GB Random Access Memory (RAM) each and 32 GB system RAM. The ThunderSVM [55] and Tensorflow [57] libraries were configured to leverage GPU acceleration.

Each model is trained as a binary classification model, classifying each sample as benign or malicious. In order to achieve this without rendering per class metrics impossible to calculate, the labels are stored as multi-class labels and are converted to binary labels immediately before being passed to the models. All correct attack predictions are then set to the appropriate class to ensure per class metrics are calculated correctly.

The results of each original study are replicated by evaluating each model on one of the datasets used by the original authors to verify the models are correct. The methodology is also evaluated on the models and dataset used by Kus et al. [19] to ensure it is replicated correctly.

Once, matching results are achieved, the models are trained on the variants generated to produce the results of this study. The unsupervised models considered expect only normal, or benign, data in their training sets. Hence, these are only trained

Table 3.2 Hyperparameters

Model	Hyperparameters
DT	splitter='best' criterion='Gini' min_samples_split=2 min_samples_leaf=1
RF	n_estimators=100 min_samples_split=2 min_samples_leaf=1 criterion='Gini
GB	loss='log_loss' learning_rate=1 n_estimators=100 max_depth=3 validation_fraction=0.1
KNN	n_neighbours=5 weights='uniform' metric='Minkowski'
LDA	solver='svd'
SSC-OCSVM	k_features=50 nu=0.1
SAE-OCSVM	lambda=10 learning_rate=0.01 n_neurons=(95,49,12) nu=0.5 kernel='rbf' gamma=0.1

once as all variants are rendered identical when malicious samples are omitted.

Hyperparameters are input values required by ML models that are not altered during training. The hyperparameters used in this study are specified in Table 3.2. All of these hyperparameters were set based on the work of the original authors of the models [11, 14, 15].

3.6 Conclusion

In this chapter we have presented the selected state-of-the art models as well as our criteria during the selection process. We then outlined the data used in our replications and experiments as well as our methodology for creating variants of this data to simulate unknown attacks. Finally, we present our preprocessing pipeline and implementation details for all models considered. In the next chapter, we will present the results from all our replications and experiments and discuss the significance of these results.

4 Results and Discussion

In this chapter, we will present the results of all our replications and experiments. The chapter is structured as follows: Section 4.1 presents the results of our replicated models when evaluated on a dataset used by the original authors. These results are presented beside the results of the original work to illustrate the similarities in the figures, confirming successful replication. The remainder of the sections in the chapter present the results of the experiments carried out as part of the methodology proposed in the previous chapter. Section 4.2 presents the metrics discussed in Section 2.3 aggregated over all the variants and classes in the first and second phases specified in Section 3.4, which take attack categories as the target label. Section 4.3 presents the same information, considering instead the third and fourth phases defined in Section 3.4, which take the individual attack name as the target label. More information about the methods of aggregation applied is specified in these sections. Section 4.4 presents heatmaps that indicate the recall values of each variant in the discussed in Section 3.4 on each label class. The predictions of each combination of model and variant are available in pickle format in the 'results' folder on this project's GitHub repository [52].

4.1 Replication

As discussed in the previous chapter, each model considered is evaluated on a dataset used by the original authors to verify it was replicated correctly. More specifically, the supervised models are evaluated on the CSE-CIC-IDS2018 dataset whilst the unsupervised models are evaluated on the NSL-KDD dataset. The evaluation methodology is also applied to two of the models and the dataset originally used by Kus et al. [19], to verify its correct replication. All the models we set out to replicate were successfully replicated with the exception of the AdaBoost model. Our attempts to replicate this model yielded poor results that did not compare to those of Karatas et al. [11].

Tables 4.1 and 4.2 display the results of the replication of the work of Karatas et al. [11]. These are the metrics used by the original authors calculated from the predictions of the models trained on the CSE-CIC-IDS2018 dataset. A stratified sample of the dataset with no omitted attacks was employed during training and testing to match the original work as closely as possible. In the same tables are the results presented in the original work. Analysing these results, we can conclude that the differences are negligible and can be amounted to the stochasticity of the training and testing process. Therefore, we conclude that the five supervised models presented

Table 4.1 Karatas et al. [11] replication aggregate results

Our Results				
Algorithm	**Accuracy**	**Recall**	**Precision**	**F1-Measure**
DT	0.99	0.97	0.94	0.95
RF	0.99	0.94	0.95	0.95
GB	0.99	1.00	0.96	0.98
KNN	0.97	0.91	0.71	0.76
LDA	0.88	0.89	0.52	0.57
Original Authors' Results				
Algorithm	**Accuracy**	**Recall**	**Precision**	**F1-Measure**
DT	0.996	0.996	0.996	0.996
RF	0.994	0.993	0.994	0.996
GB	0.993	0.993	0.993	0.993
KNN	0.981	0.981	0.979	0.980
LDA	0.912	0.912	0.920	0.916

were replicated successfully.

Figure 4.1 shows the results of the replication of the work of Pu et al. [14] in the form of the ROC curve of the replicated model. This model was trained and tested on the complete NSL-KDD dataset preprocessed according to the methodology of the original authors. This curve can be compared to the curve presented by the original authors to verify the replication of the model. The right image in the figure is the image presented by the original authors, with the red line representing the proposed model, SSC-OCSVM. Comparing the two curves, the performance of our replicated model is similar to that of the original authors, and hence we conclude the model was successfully replicated.

Table 4.3 shows the results of the replication of the work of Cao et al. [15]. These figures represent the AUC-ROC per class calculated from the predictions of the model when trained on the complete NSL-KDD dataset. The preprocessing pipeline applied consists of the same steps proposed by Cao et al. [15]. Comparing the AUC-ROC scores of our replicated model with the original work, we observe negligible differences which can be attributed to the stochasticity of the training and testing process. Hence, we conclude the model was successfully replicated.

Figures 4.2 and 4.3 show the results of the replication of the work of Kus et al. [19]. Note, only the RF model results are presented. The Support Vector Machine (SVM) and BiLSTM models were also replicated, however these results are not presented due to space constraints. The methodology of Kus et al. [19] is consistent across all models and hence, these results are sufficient to confirm the validity of the

Table 4.2 Karatas et al. [11] replication accuracy per class

Our Results						
Algorithm	**Benign**	**Bot**	**DoS**	**Brute Force**	**Injection**	**Infiltration**
DT	0.993	1.000	1.000	1.000	0.805	1.000
RF	0.991	1.000	1.000	1.000	0.747	0.917
GB	0.991	1.000	1.000	1.000	0.987	1.000
KNN	0.970	1.000	1.000	1.000	0.490	1.000
LDA	0.840	1.000	1.000	0.961	0.626	0.917
Original Authors' Results						
Algorithm	**Benign**	**Bot**	**DoS**	**Brute Force**	**Injection**	**Infiltration**
DT	0.996	1.000	1.000	1.000	0.962	0.856
RF	0.995	1.000	1.000	1.000	1.000	0.927
GB	0.989	1.000	1.000	0.996	1.000	0.978
KNN	0.981	1.000	1.000	0.999	1.000	0.739
LDA	0.863	0.984	0.999	0.518	0.674	0.978

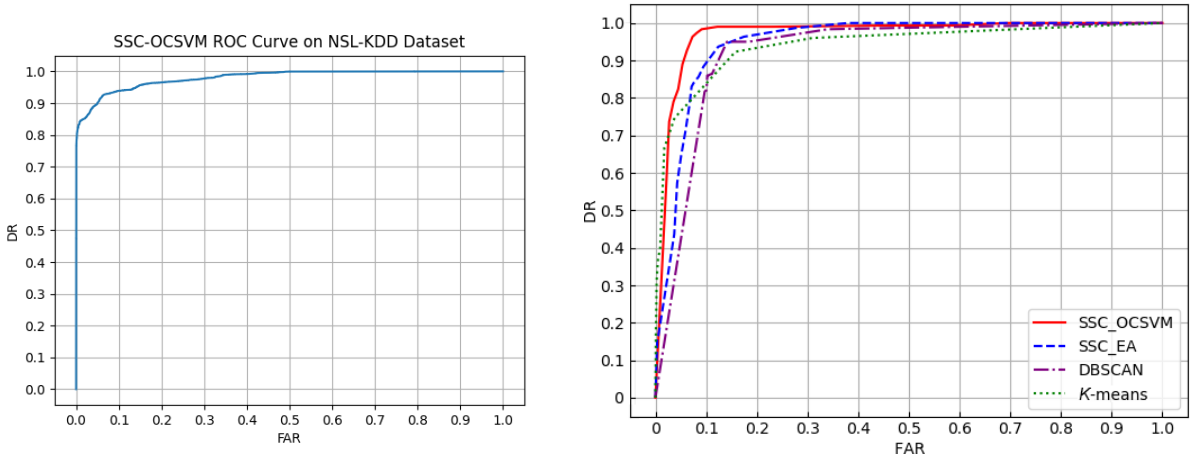


Figure 4.1 The ROC curve of our replicated SSC-OCSVM model (left) compared to that of the original author (right).

Table 4.3 Cao et al. [15] replication AUC-ROC per class

Our Results				
Algorithm	**Probe**	**DoS**	**R2L**	**U2R**
SAE-OCSVM	0.975	0.972	0.926	0.959
Original Authors' Results				
SAE-OCSVM	0.987	0.972	0.923	0.948

replication. These figures represent heatmaps indicating the per class recall values of each training variant. Each row represents the per class recall values of a particular variant whereas each column represents the recall values of each variant on the instances of a particular class. The heatmaps presented are those of our replication. Unlike the previous replications, we cannot present the results of Kus et al. [19] here due to space limitations, however these can be found in the original paper.

Note, the label encodings used by our replication is different to that of the original authors, hence, the columns of the heatmaps appear to be shuffled. The encoding zero represents the benign class in both our replication and the original work. Analysing the heatmaps, we observe near identical patterns in our replication when compared to the original results. Hence, we can conclude our implementation accurately replicates the methodology proposed by Kus et al. [19].

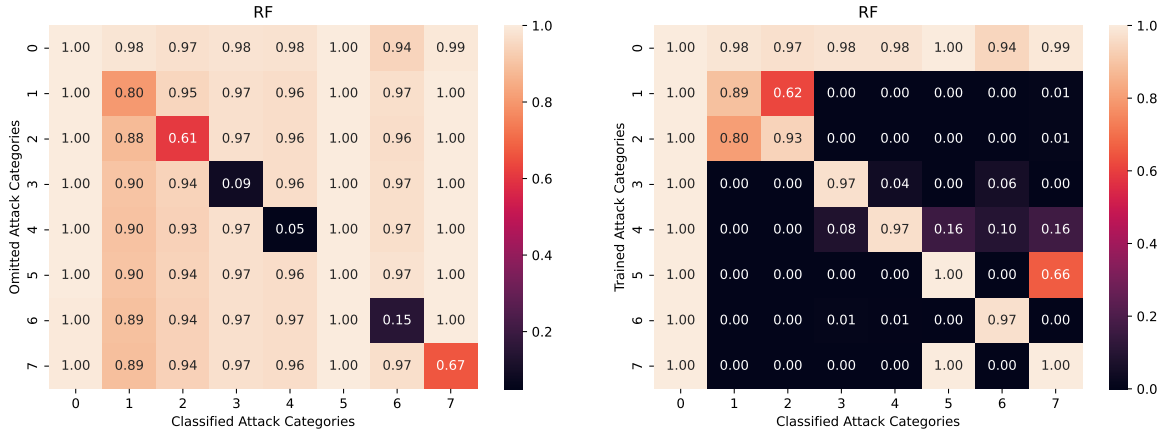


Figure 4.2 Kus et al. [19] replication heatmaps illustrating recall per category on each training variant

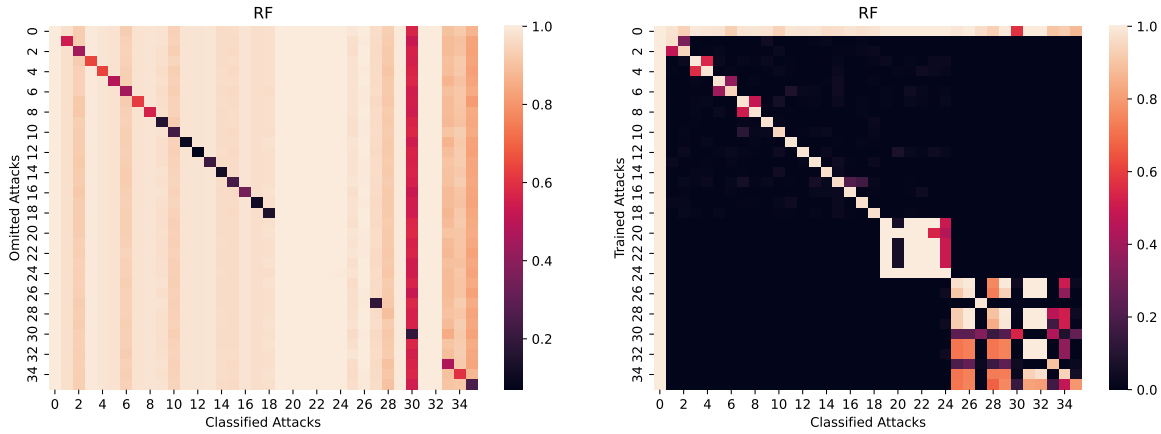


Figure 4.3 Kus et al. [19] replication heatmaps illustrating recall for each individual attack on each training variant

Table 4.4 Aggregate results when excluding categories

Algorithm	Accuracy	Recall	Recall-Unk	Precision	F1-Measure
DT	0.962	0.828	0.134	0.942	0.832
RF	0.964	0.830	0.228	0.910	0.848
GB	0.948	0.804	0.118	0.872	0.800
KNN	0.958	0.800	0.248	0.900	0.796
LDA	0.914	0.766	0.564	0.850	0.718
SSC-OCSVM	0.760	0.590	0.554	0.820	0.670
SAE-OCSVM	0.800	0.640	0.600	0.830	0.650

4.2 Aggregate Results Omitting Categories

The aggregate results of all the experiments are shown in Table 4.4.

The methodology described in the previous chapter yields these metrics for each variant and class. To better interpret the overall performance of each model and address Objective 1 we present the average of all these metrics, considering only variants of Phase 1 discussed in Section 3.4, which exclude attack categories. These variants were selected to generate the aggregate results as they offer the most realistic simulation of a practical scenario whereby an unknown attack is deployed on a model trained on several categories of known attacks.

The unsupervised models were trained on only one variant as all variants are rendered identical following their preprocessing pipelines, which involve removing all malicious samples. Hence, the above discussed averaging does not apply to these models. Both supervised and unsupervised models, however, generate metrics per label class. The macro-average, which is a non-weighted average, is presented when aggregating across classes to reduce the impact of class imbalance.

The results of the supervised models are generally high indicating a good ability to detect known attacks from network data, with the DT and RF models performing slightly better than the other models. However, this degrades significantly in the face of unknown attacks. This is indicated by the low values in the Recall-Unk column for all models except for LDA. This indicates a very low ability to generalise to unknown attack categories in these models. The LDA model stands out from the other supervised models, exhibiting a significantly higher Recall-Unk value, comparable to those of the unsupervised models. The value is still significantly lower than those of known attacks, however is an impressive value nonetheless due to the inherent difficulty in detecting unknown attacks.

The unsupervised models on the other hand exhibit a less significant difference between their recall and Recall-Unk values, however have a lower recall value than the

Table 4.5 Accuracy per class when excluding categories

Algorithm	Benign	Bot	DoS	Brute Force	Injection	Infiltration
DT	0.990	0.971	0.849	0.902	0.938	0.440
RF	0.994	0.802	0.815	0.902	0.862	0.448
GB	0.989	0.741	0.622	0.751	0.754	0.240
KNN	0.985	0.801	0.831	0.920	0.892	0.388
LDA	0.940	0.773	0.805	0.951	0.954	0.162
SSC-OCSVM	0.772	0.006	0.778	0.999	0.667	0.313
SAE-OCSVM	0.818	0.496	0.734	0.999	0.384	0.391

Table 4.6 F1-Measure per class when excluding categories

Algorithm	Benign	Bot	DoS	Brute Force	Injection	Infiltration
DT	0.978	0.734	0.878	0.936	0.964	0.568
RF	0.978	0.670	0.828	0.936	0.894	0.576
GB	0.962	0.508	0.644	0.772	0.774	0.368
KNN	0.974	0.554	0.856	0.950	0.926	0.522
LDA	0.948	0.252	0.892	0.972	0.974	0.276
SSC-OCSVM	0.850	0.000	0.880	1.000	0.800	0.480
SAE-OCSVM	0.870	0.070	0.850	1.000	0.560	0.560

supervised models. The accuracies and F1-scores of the models are also lower indicating worse overall performance when compared to the supervised models. The accuracies stand at 76% and 80% which are notable figures, however, as previously discussed, accuracy may be skewed by a large number of correct benign predictions. Hence, the recalls and F1-scores may be more significant given the class imbalance present in the dataset. Considering the recalls and F1-scores in Table 4.4, the unsupervised models could certainly provide valuable information, however may miss a significant number of attacks.

Comparing the SSC-OCSVM model to the SAE-OCSVM model, the performance figures are quite similar, indicating DL may not suffer the same inefficacy discovered by Zoppi et al. [17] when considering unsupervised models.

Table 4.5 presents the aggregate per class accuracy calculated by averaging the per class accuracies of each variant where an attack category is omitted. Table 4.6 contains the per class F1-scores averaged across the same variants.

Analysing the results of these tables, we can further investigate the behaviour of the models. Firstly, we note that some attack categories appear to be more difficult to predict than others. For instance, the ‘Infiltration’ category exhibits lower accuracy across all models. A potential explanation for this is that ‘Infiltration’ is a broad category

involving the exploitation of software that can take a wide variety of forms. Hence, 'Infiltration' attacks may differ significantly among themselves making it more difficult for the model to learn common patterns. Additionally, the dataset only contains five attacks under this category, with a relatively small number of instances compared to most other categories, potentially aggravating the issue. Another potential explanation may be that 'Infiltration' attacks resemble benign traffic more closely than other categories, rendering themselves difficult to distinguish from normal traffic.

The comparison of the unsupervised model with the supervised models in Table 4.5 and Table 4.6 reveals an interesting phenomenon. Whilst the unsupervised models perform far worse overall, they seem to perform very similarly to the supervised models on certain categories. The models exhibit low efficacy on the 'Bot', 'Injection' and 'Infiltration' categories specifically, weighing down their averages in the overall results. It should be noted, however that the supervised models also perform poorly on the 'Infiltration' category, rendering performance on the 'Bot' and 'Injection' categories the primary difference. This is an interesting observation as this means unsupervised models could serve as very effective tool against both known and unknown attacks of specific categories, against which it is known to be effective. Additionally, this finding brings into question the reasons for this difference in performance. One potential explanation is that these categories resemble benign traffic more closely making it difficult to recognise them as anomalies, however, this hypothesis would require further experimentation to confirm or deny concretely.

Comparing the per class metrics of the SSC-OCSVM model with that of the SAE-OCSVM model also yields an interesting finding. The former outperforms the latter on the 'Injection' category but performs worse on the 'Bot' category, forming a trade-off between the two. The exact nature of this trade-off and the internal mechanics causing it would need further experimentation to uncover.

4.3 Aggregate Results Omitting Specific Attacks

Table 4.7 displays the aggregate metrics from the third phase discussed in Section 3.4, which involves excluding individual attacks to generate variants. Similarly to Table 4.4, the macro-average is taken across classes to avoid overstating the results of the benign class.

The primary point to note from these results is the higher values found in the Recall-Unk column. The supervised models, while still failing to match the generalisation ability of the unsupervised model, perform far better than when the entire category is omitted. This indicates these models are capable of generalising to similar attacks and perform better on unknown attacks when attacks in the same

Table 4.7 Aggregate results when excluding specific attacks

Algorithm	Accuracy	Recall	Recall-Unk	Precision	F1-Measure
DT	0.981	0.870	0.465	0.960	0.884
RF	0.985	0.877	0.506	0.977	0.895
GB	0.980	0.830	0.546	0.966	0.842
KNN	0.975	0.851	0.500	0.970	0.860
LDA	0.918	0.758	0.610	0.958	0.767
SSC-OCSVM	0.760	0.630	0.566	0.900	0.690
SAE-OCSVM	0.800	0.640	0.600	0.830	0.650

category are present in the dataset. However, as previously discussed, this generalisation ability is limited and still fails to supersede that of the unsupervised model.

4.4 Per Variant Results Omitting Categories

In this section, we present and discuss heatmaps showing the recall values on each category for each variant considered. This data allows us to address Objective 3, revealing the relationships each model forms between the presence of attacks or attack categories in the training set, and the predictions of each class.

Figure 4.4 shows the heatmaps of recall values of the supervised models generated from the variants omitting categories. Each row represents the per category recall values generated from one variant. The label of the row is the category that was omitted.

The results in these heatmaps further confirm the observations made earlier on the high efficacy of supervised models on known attacks and reduced efficacy on unknown attacks, indicated by the lower diagonal values. The DT, RF and KNN models seem to behave similarly, performing excellently on known attacks but more poorly on unknown attacks, with DT demonstrating slightly higher generalisation ability compared to the other models. The LDA model differs slightly, exhibiting higher generalisation ability but slightly lower values overall and significantly lower values on the ‘Infiltration’ category specifically. The GB model exhibits strange behaviour as it seems to be dependent on the ‘Infiltration’ category to learn patterns on the other classes, achieving zero recall values when the category is excluded. Paradoxically, the model also displays lower recall on the ‘Infiltration’ category compared to the other models.

This strange behaviour can be further explored in Figure 4.5, which shows the heatmaps of recall values generated from the variants including single categories.

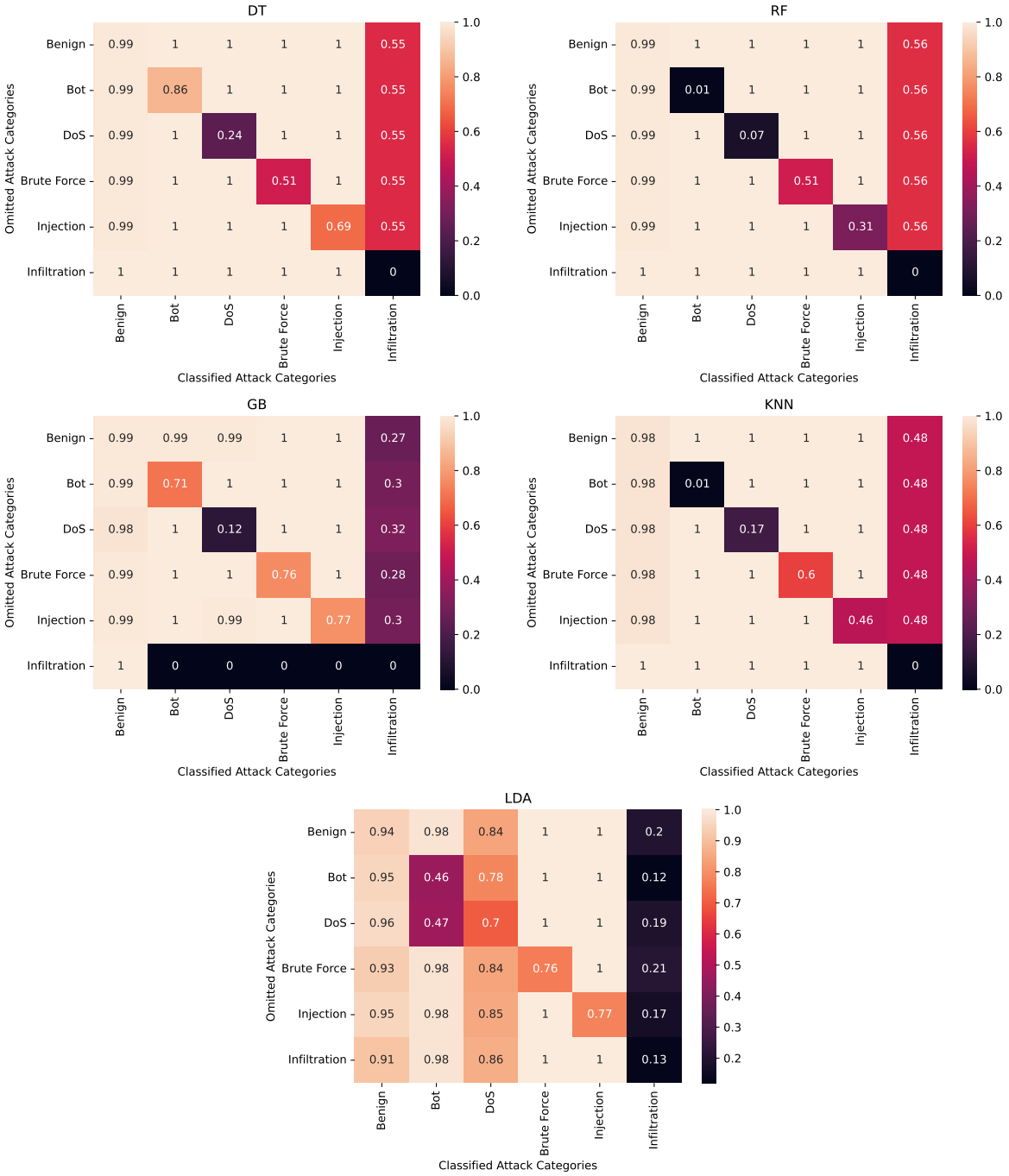


Figure 4.4 Recall values per class when trained on variants omitting a single category.

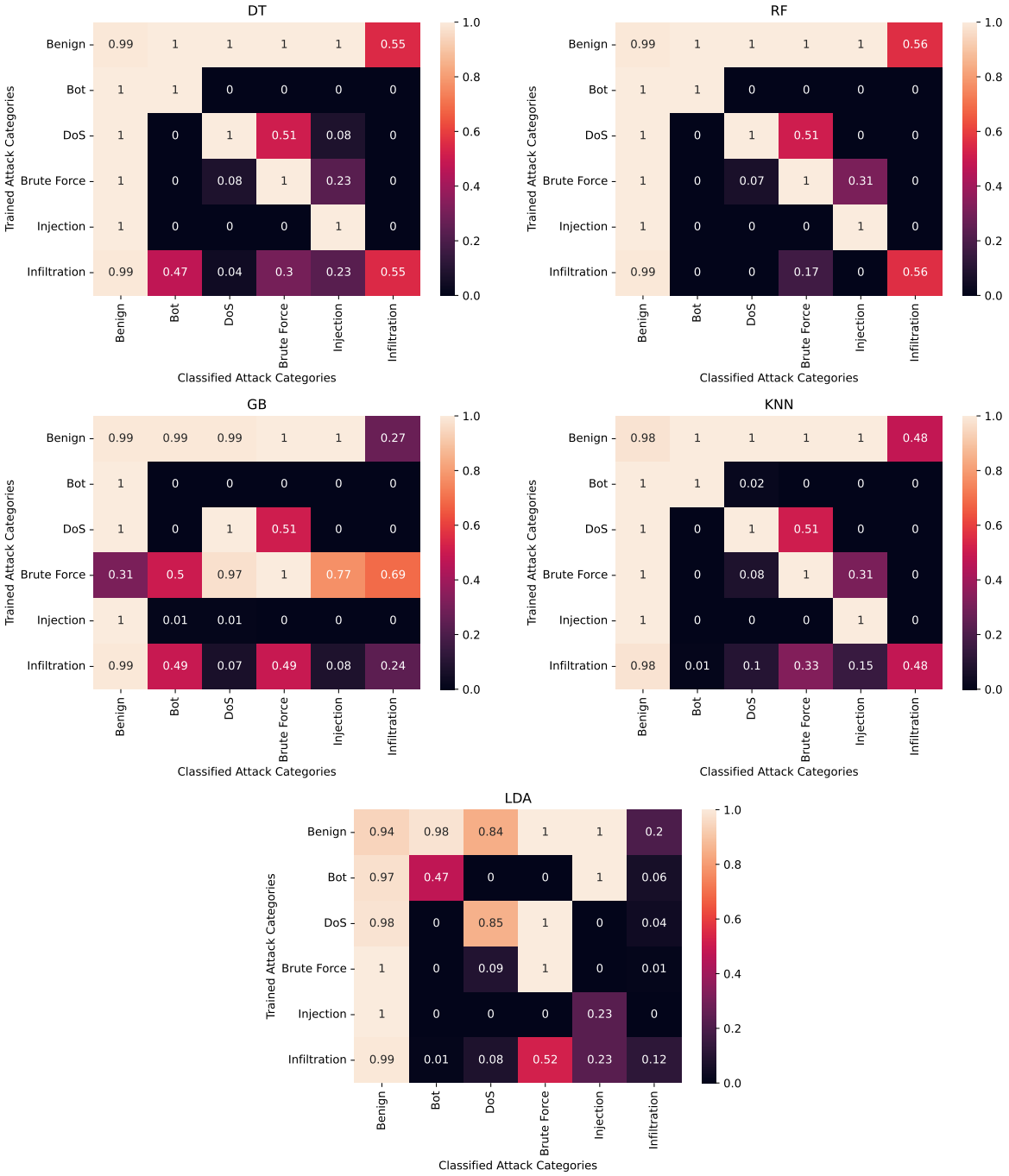


Figure 4.5 Recall values per class when trained on a single category.

Based on the values presented in this heatmap, the GB model seems to be able to learn patterns that allow it to generalise from the ‘Brute Force’ and ‘Infiltration’ categories, but fails to learn anything from the ‘Bot’ and ‘Injection’ categories. This is in contrast with the DT, RF and KNN models, which are able to learn from all categories, indicated by the high recall values in the diagonal, however are not able to generalise when trained on ‘Brute Force’ attacks. The LDA model displays lower values in the diagonal and generalises similarly to the other models excluding the GB model. This indicates lower performance overall when trained on single attacks.

All classifiers except for the RF classifier seem to be able to generalise from the ‘Infiltration’ class. This behaviour, in conjunction with the previous observations on the low accuracy on this class supports the idea that the ‘Infiltration’ category may be too broad and contain more than one sort of attack. This would explain why it is difficult to classify, yet offers information about a wide variety of attacks.

Despite these limited instances of generalisation, the overall generalisation ability of the five models is low. Most instances of generalisation on this set of heatmaps yield low recall values, with the diagonal containing significantly higher values indicating a limited ability to generalise across categories overall.

4.5 Per Variant Results Omitting Specific Attacks

Figure 4.6 shows the heatmaps of recall values of the supervised models generated from the variants omitting individual attacks. Each row represents the per attack recall values generated from one variant. The label of the row is the attack that was omitted.

The primary patterns to observe are the vertical stripes wherever ‘Infiltration’ attacks are present and the diagonal, representing the difficulty of classifying ‘Infiltration’ attacks and generalising to unknown attacks. It is interesting to note, the diagonals have gaps, unlike in the category heatmaps. This indicates intra-category generalisation occurs during training, creating these gaps that are not visible in the category heatmaps. The slightly more noisy heatmaps resulting from the GB model is further evidence this model is more dependent on generalised patterns compared to the other models. The LDA model also exhibits a slightly noisier heatmap however to a much lower degree than the GB model.

Figure 4.7 shows the heatmaps of recall values generated from the variants including single attacks. These results indicate that while generalisation is somewhat scarce, there are numerous attacks in the dataset that influence the model’s performance on other attacks. The relationships between each pair of attacks are indicated in the heatmaps. It should be noted that the algorithm employed seems to have a significant effect on the level of generalisation achieved, with GB, DT and LDA

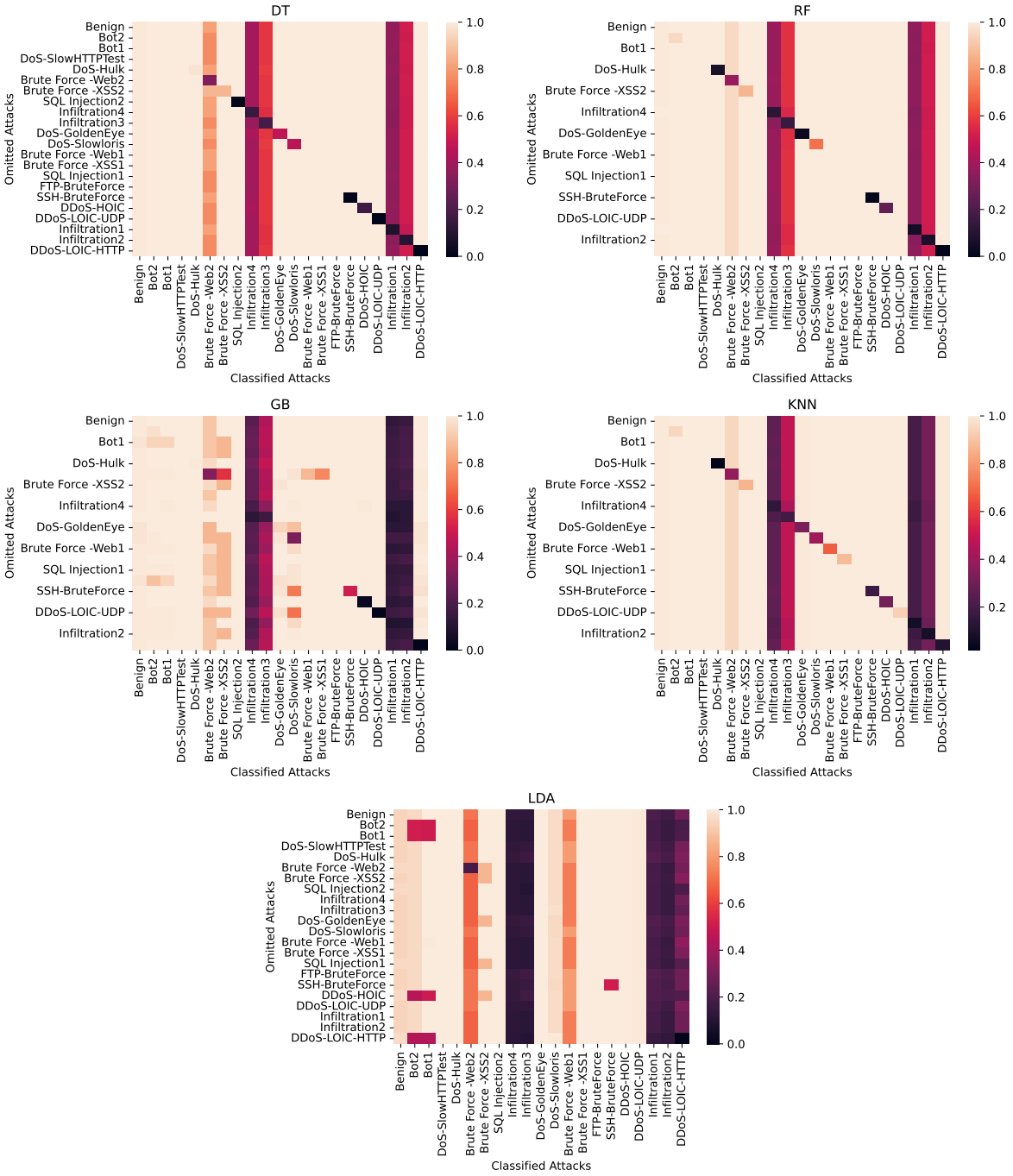


Figure 4.6 Recall values per class when trained on variants omitting a single attack.

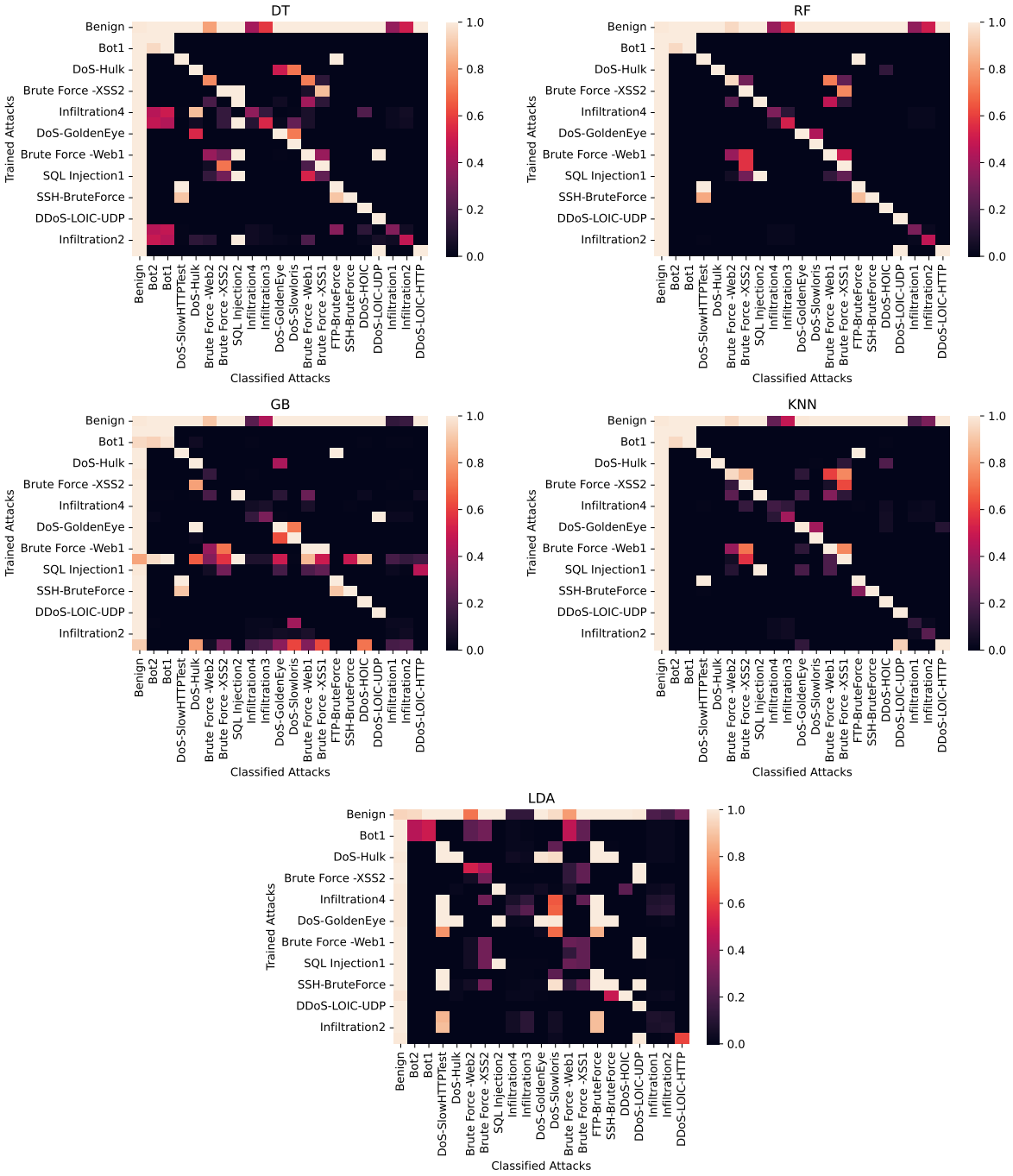


Figure 4.7 Recall values per class when trained on a single attack.

demonstrating the higher generalisation ability than RF and KNN.

4.6 Addressing Objectives

Sections 4.2 and 4.3 contain all the information required to address Objective 1. The state-of-the-art supervised models considered demonstrate impressive efficacy in detecting known network intrusions, however, their efficacy diminishes to an alarming degree when faced with unknown attacks. Hence, their overall efficacy will depend on the types of attacks the system is expected to face, and the dataset available for training. The state-of-the-art unsupervised models considered exhibit far lower efficacy, however, experience negligible decline against unknown attacks. This may offer an invaluable tool in flagging potentially malicious behaviour, however, may not be suitable if a high degree of confidence in predictions is required.

The same sections also detail enough information to address Objective 2, it is clear to see that the supervised models considered are more effective overall, performing better in all metrics except for Recall-Unk. However, unsupervised models are significantly more effective in detecting unknown attacks.

The heatmaps presented in Figures 4.5 and 4.7 address Objective 3, clearly illustrating the relationships between attack categories and individual attacks that allow the different ML models to generalise across classes.

4.7 Threats to Validity

It should be noted from the results, that the unsupervised models perform much better on the NSL-KDD dataset than the CSE-CIC-IDS2018 dataset. This highlights the issue of generalisation across datasets. The unsupervised models considered were selected as they were the most highly cited and best performing models within the domain that provide a clear methodology from the models found during the literature review. This implicitly assumes that the high efficacy demonstrated on the NSL-KDD dataset and the other datasets they were evaluated on, generalises well to the CSE-CIC-IDS2018 dataset, which may not always be the case. Hence, it is possible these models are not the best performing models on the dataset considered for this study, resulting in an unfair comparison with the supervised models.

Furthermore, the issue of cross-dataset generalisation also brings into question how well these results will truly generalise outside the lab environment. Whilst, to the best of our knowledge, the CSE-CIC-IDS2018 dataset is the most up-to-date and realistic dataset currently available, it may still not generalise well to other datasets and practical environments.

As mentioned in Section 3.3 the individual attack labels were generated based on the timestamp and label information in Table 2 of the CSE-CIC-IDS2018 website [47]. It was also mentioned that some attack instances are dated outside the range of any attack listed on the table, and it has therefore been assumed that instances occurring on a date with only one attack of that label, belong to that attack. This assumption, whilst likely the most logical approach, may lead to the mislabelling of certain samples. Measuring the likelihood of this or mitigating the issues is difficult without more information on the source of the ambiguous samples, which does not appear to be available in the resources documenting the dataset.

Finally, all pragmatic cyberthreat defence systems present vulnerabilities of their own which should be kept in mind when considering results. The CSE-CIC-IDS2018 dataset was not designed with the NIDS components employed by this study in mind. Hence, the attacks it contains were performed without any knowledge of the defence system we have employed. In practice, attackers could learn implementation details of any defence system and orchestrate targeted attacks intended to circumvent that specific defence system. A prime example of such an attack targeted at ML-based NIDS are adversarial attacks, which are attacks intentionally designed to evade ML-based classifiers. [58] and [59] propose examples of such attack tools.

4.8 Conclusion

In this chapter, we have presented the results of our replicated models both on datasets used by the original authors and on the variants of the CSE-CIC-IDS2018 dataset generated from the methodology discussed in the previous chapter. These results address all three objectives originally defined in Chapter 1 and provide valuable insights into the strengths and weaknesses of different machine learning techniques in the face of both known and unknown attacks. In the next chapter, we conclude the document, summarising are key points and proposing future research directions.

5 Conclusion

In conclusion, cybersecurity is an ever-evolving landscape with new threats constantly being developed. NIDS are a crucial component to the defence of any information system, analysing data and identifying attacks in real time. Unknown attacks are a vital consideration when developing NIDS as most systems will encounter these attacks in practice and failure to detect such attacks could lead to a security breach. ML techniques offer tremendous potential in this field due to their unique ability to identify complex patterns unknown to the developer. This offers the potential to learn patterns allowing for the detection of unknown attacks before these attacks are discovered or perhaps even developed.

This potential has not gone unnoticed in the research community with numerous authors proposing and exploring ML techniques for NIDS. Some of these authors have achieved remarkable results demonstrating the efficacy of ML in this field. However, most of these works do not consider the impact of unknown attacks when evaluating their models, creating a false sense of security that may leave information systems vulnerable. In this study, we have replicated seven ML models from three state-of-the-art works and evaluated them using the methodology proposed by the work of Kus et al. [19], which has also been replicated. More specifically, we have replicated and evaluated the DT, RF, GB, KNN, LDA, SSC-OCSVM and SAE-OCSVM models. The best performing algorithm from these in the domain of NIDS depends on the attacks the system is expected to face. DT demonstrates the best performance on known attacks, however, SAE-OCSVM demonstrates the best performance on unknown attacks. The LDA model offers a balance between the two and may prove an effective option in the right circumstances.

This analysis has addressed our three primary objectives. We have analysed the efficacy of current state-of-the-art techniques and discovered that they are extremely effective on known attacks and on certain categories of unknown attacks, however, some categories require prior knowledge to be classified effectively. We have also compared the efficacies of supervised and unsupervised techniques and discovered that supervised techniques are more effective on known attacks overall as they do not struggle with any categories. Unsupervised techniques, in contrast, are ineffective against certain categories, for example the ‘Bot’ category, for which both unsupervised models demonstrated low metrics. Unsupervised techniques, however, are unaffected by the prior knowledge, or lack thereof, available on attacks. Supervised techniques, in contrast, suffer a considerable decrease in efficacy without prior knowledge of the attack. Finally, we have explored the relationships present between individual attacks and attack categories in regard to generalisation. This analysis has revealed an

alarmingly low level of generalisation in supervised models, even within attack categories in certain cases. However, it should be noted that some level of generalisation does take place, potentially offering an advantage over SIDS. Furthermore, the specific algorithm employed has a significant effect on the level of generalisation achieved.

Overall, these results should help to shed light on the efficacy and behaviour of a variety of ML techniques in the ever-evolving cyberthreat landscape we face.

5.1 Future Work

Moving forward, work on integrating the benefits of both supervised and unsupervised techniques could reveal new opportunities within the field. A variety of techniques could be explored to combine the predictions generated from both supervised and unsupervised classifiers which may be able to reap the benefits of both approaches.

Additionally, other less conventional and novel techniques could be explored. The models introduced at the end of Section 2.5, for instance, would serve as excellent candidates for a comparative analysis similar to this one. These were not considered in this work due to time limitations and due to the fact some of them were not published at the time of model selection.

All the models considered in this work treat samples individually, without considering neighbouring samples. Models that consider time series data may be able to learn patterns present across several network flows originating from the same attack, which may improve model performance. Future work could consider such time series models in comparison to single sample models such as those analysed in this work.

Finally, an evaluation of these techniques in conjunction with current practical NIDS tools employed in the industry, such as Elastic Security, would help to further evaluate the efficacy of these techniques in practical applications. Due to time constraints, this was not possible in this study. As discussed in Section 2.1, Elastic Security combines data from various sources including ML pipelines. Hence, each of these ML models could be added as a custom ML detection rule. A comparison could then be made between the performance of the system with and without the custom rule to determine whether these techniques could improve pragmatic NIDS performance when incorporated with current state-of-the-art tools.

References

[1] European Union Agency for Cybersecurity (ENISA), "Enisa threat landscape 2023," European Union Agency for Cybersecurity (ENISA), Annual Threat Report 2023-12-11, 2023. [Online]. Available: https://www.enisa.europa.eu/topics/cyber-threats/threats-and-trends/?tab=details.

[2] M. Corkery, "Once again, thieves enter swift financial network and steal," *The New York Times*, May 2016. [Online]. Available: https://www.nytimes.com/2016/05/13/business/dealbook/swift-global-bank-network-attack.html (visited on 05/20/2024).

[3] T. Bergin and N. Layne, *Special report: Cyber thieves exploit banks' faith in swift transfer network*, Reuters, May 2016. [Online]. Available: https://www.reuters.com/article/us-cyber-heist-swift-specialreport-idUSKCN0YB0DD/.

[4] S. Liss, *Commonspirit health confirms it was hit by ransomware attack*, Healthcare Dive, 2022. [Online]. Available: https://www.healthcaredive.com/news/commonspirit-health-ransomware-cyberattack/634011/.

[5] H. Journal, *Commonspirit health issues update confirming 164 facilities affected by ransomware attack*, HIPAA Journal, Apr. 2023. [Online]. Available: https://www.hipaajournal.com/commonspirit-health-issues-update-confirming-164-facilities-affected-by-ransomware-attack/.

[6] C. Page, *Commonspirit health says patient data was stolen during ransomware attack*, TechCrunch, Dec. 2022. [Online]. Available: https://techcrunch.com/2022/12/09/commonspirit-health-ransomware-attack-exposed-patient-data/.

[7] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, Jul. 2019, ISSN: 2523-3246. DOI: 10.1186/s42400-019-0038-7. [Online]. Available: http://dx.doi.org/10.1186/s42400-019-0038-7.

[8] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, Mar. 2021, ISSN: 2523-3246. DOI: 10.1186/s42400-021-00077-7. [Online]. Available: http://dx.doi.org/10.1186/s42400-021-00077-7.

[9] L. Bilge and T. Dumitraş, “Before we knew it: An empirical study of zero-day attacks in the real world,” in *Proceedings of the 2012 ACM conference on Computer and communications security*, ser. CCS’12, ACM, Oct. 2012. DOI: 10.1145/2382196.2382284. [Online]. Available: http://dx.doi.org/10.1145/2382196.2382284.

[10] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges,” *Computers & Security*, vol. 28, no. 1, pp. 18–28, 2009, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2008.08.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404808000692.

[11] G. Karatas, O. Demir, and O. K. Sahingoz, “Increasing the performance of machine learning-based idss on an imbalanced and up-to-date dataset,” *IEEE Access*, vol. 8, pp. 32 150–32 162, 2020. DOI: 10.1109/ACCESS.2020.2973219.

[12] K. Jiang, W. Wang, A. Wang, and H. Wu, “Network intrusion detection combined hybrid sampling with deep hierarchical network,” *IEEE Access*, vol. 8, pp. 32 464–32 476, 2020. DOI: 10.1109/ACCESS.2020.2973730.

[13] S. N. Mighan and M. Kahani, “A novel scalable intrusion detection system based on deep learning,” *International Journal of Information Security*, vol. 20, no. 3, pp. 387–403, Jun. 2020, ISSN: 1615-5270. DOI: 10.1007/s10207-020-00508-5. [Online]. Available: http://dx.doi.org/10.1007/s10207-020-00508-5.

[14] G. Pu, L. Wang, J. Shen, and F. Dong, “A hybrid unsupervised clustering-based anomaly detection method,” *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 146–153, 2021. DOI: 10.26599/TST.2019.9010051.

[15] V. L. Cao, M. Nicolau, and J. McDermott, “Learning neural representations for network anomaly detection,” *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 3074–3087, Aug. 2019, ISSN: 2168-2275. DOI: 10.1109/tcyb.2018.2838668. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2018.2838668.

[16] R. Atefinia and M. Ahmadi, “Network intrusion detection using multi-architectural modular deep neural network,” *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3571–3593, Aug. 2020, ISSN: 1573-0484. DOI: 10.1007/s11227-020-03410-y. [Online]. Available: http://dx.doi.org/10.1007/s11227-020-03410-y.

[17] T. Zoppi, A. Ceccarelli, T. Puccetti, and A. Bondavalli, “Which algorithm can detect unknown attacks? comparison of supervised, unsupervised and meta-learning algorithms for intrusion detection,” *Computers & Security*, vol. 127, p. 103 107, 2023, ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2023.103107. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823000172.

[18] L. Liu, P. Wang, J. Lin, and L. Liu, “Intrusion detection of imbalanced network traffic based on machine learning and deep learning,” *IEEE Access*, vol. 9, pp. 7550–7563, 2021. DOI: 10.1109/ACCESS.2020.3048198.

[19] D. Kus *et al.*, “A false sense of security?: Revisiting the state of machine learning-based industrial intrusion detection,” in *Proceedings of the 8th ACM on Cyber-Physical System Security Workshop*, ser. ASIA CCS ’22, ACM, May 2022. DOI: 10.1145/3494107.3522773. [Online]. Available: http://dx.doi.org/10.1145/3494107.3522773.

[20] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, “Zero-day attack detection: A systematic literature review,” *Artificial Intelligence Review*, vol. 56, no. 10, pp. 10 733–10 811, Feb. 2023, ISSN: 1573-7462. DOI: 10.1007/s10462-023-10437-z. [Online]. Available: http://dx.doi.org/10.1007/s10462-023-10437-z.

[21] Z. Ghahramani, “Unsupervised learning,” in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 72–112, ISBN: 9783540286509. DOI: 10.1007/978-3-540-28650-9_5. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-28650-9_5.

[22] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion detection system: A comprehensive review,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, Jan. 2013, ISSN: 1084-8045. DOI: 10.1016/j.jnca.2012.09.004. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2012.09.004.

[23] I. El Naqa and M. J. Murphy, “What is machine learning?” In *Machine Learning in Radiation Oncology*. Springer International Publishing, 2015, pp. 3–11, ISBN: 9783319183053. DOI: 10.1007/978-3-319-18305-3_1. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-18305-3_1.

[24] F. Emmert-Streib and M. Dehmer, “Taxonomy of machine learning paradigms: A data-centric perspective,” *WIREs Data Mining and Knowledge Discovery*, vol. 12, no. 5, Jun. 2022, ISSN: 1942-4795. DOI: 10.1002/widm.1470. [Online]. Available: http://dx.doi.org/10.1002/widm.1470.

[25] V. Nasteski, "An overview of the supervised machine learning methods," *HORIZONS.B*, vol. 4, pp. 51–62, Dec. 2017, ISSN: 1857-9892. DOI: 10.20544/horizons.b.04.1.17.p05. [Online]. Available: http://dx.doi.org/10.20544/HORIZONS.B.04.1.17.P05.

[26] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, Feb. 2015, ISSN: 2196-1115. DOI: 10.1186/s40537-014-0007-7. [Online]. Available: http://dx.doi.org/10.1186/s40537-014-0007-7.

[27] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012, ISSN: 1557-7317. DOI: 10.1145/2347736.2347755. [Online]. Available: http://dx.doi.org/10.1145/2347736.2347755.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, ISSN: 1476-4687. DOI: 10.1038/nature14539. [Online]. Available: http://dx.doi.org/10.1038/nature14539.

[29] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study1," *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–449, Nov. 2002, ISSN: 1088-467X. DOI: 10.3233/ida-2002-6504. [Online]. Available: http://dx.doi.org/10.3233/IDA-2002-6504.

[30] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data," *Journal of Big Data*, vol. 7, no. 1, Nov. 2020, ISSN: 2196-1115. DOI: 10.1186/s40537-020-00382-x. [Online]. Available: http://dx.doi.org/10.1186/s40537-020-00382-x.

[31] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, ISSN: 1076-9757. DOI: 10.1613/jair.953. [Online]. Available: http://dx.doi.org/10.1613/jair.953.

[32] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, Apr. 2018, ISSN: 1076-9757. DOI: 10.1613/jair.1.11192. [Online]. Available: http://dx.doi.org/10.1613/jair.1.11192.

[33] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers - a tutorial," *ACM Comput. Surv.*, vol. 54, no. 6, Jul. 2021, ISSN: 0360-0300. DOI: 10.1145/3459665. [Online]. Available: https://doi.org/10.1145/3459665.

[34] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.

[35] J. Li *et al.*, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, Dec. 2017, ISSN: 0360-0300. DOI: 10.1145/3136625. [Online]. Available: https://doi.org/10.1145/3136625.

[36] R. L. Perez, F. Adamsky, R. Soua, and T. Engel, "Machine learning for reliable network attack detection in scada systems," in *2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE)*, IEEE, 2018, pp. 633–638.

[37] Z. Qiu, D. Zhou, Y. Zhai, B. Liu, L. He, and J. Cao, *Vaemax: Open-set intrusion detection based on openmax and variational autoencoder*, 2024. arXiv: 2403.04193 [cs.CR].

[38] E. Seo, H. M. Song, and H. K. Kim, "Gids: Gan based intrusion detection system for in-vehicle network," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, 2018, pp. 1–6. DOI: 10.1109/PST.2018.8514157.

[39] H. Huang, T. Li, Y. Ding, B. Li, and A. Liu, "An artificial immunity based intrusion detection system for unknown cyberattacks," *Applied Soft Computing*, vol. 148, p. 110 875, 2023, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2023.110875. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494623008931.

[40] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009. DOI: 10.1109/TKDE.2008.239.

[41] S. D. Bay, "The uci kdd archive," *http://kdd. ics. uci. edu*, 1999.

[42] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6. DOI: 10.1109/CISDA.2009.5356528.

[43] *DDoS 2007 attack*, https://catalog.caida.org/dataset/ddos_attack_2007, Dates used: <date(s) used>. Accessed: <date accessed>. DOI: https://catalog.caida.org/dataset/ddos_attack_2007.

[44] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[45] N. Moustafa and J. Slay, "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)*, IEEE, 2015, pp. 1–6.

[46] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, pp. 108–116, 2018.

[47] *IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB — unb.ca*, https://www.unb.ca/cic/datasets/ids-2018.html, [Accessed 01-12-2023].

[48] I. Ullah and Q. H. Mahmoud, "A technique for generating a botnet dataset for anomalous activity detection in iot networks," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 134–140. DOI: 10.1109/SMC42975.2020.9283220.

[49] *A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018) - Registry of Open Data on AWS — registry.opendata.aws*, https://registry.opendata.aws/cse-cic-ids2018/, [Accessed 01-12-2023].

[50] T. H. Morris, Z. Thornton, and I. Turnipseed, "Industrial control system simulation and data logging for intrusion detection system research," *7th annual southeastern cyber security summit*, pp. 3–4, 2015.

[51] N. O. F. Elssied, O. Ibrahim, and A. H. Osman, "A novel feature selection based on one-way anova f-test for e-mail spam classification," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 7, no. 3, pp. 625–638, 2014.

[52] C. Azzopardi, *Source code for "a comparative analysis of different machine learning techniques in intrusion detection against evolving cyberthreats"*, 2024. [Online]. Available: https://github.com/calvinA21/FYP.

[53] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[54] *GitHub - dmlc/xgboost at 82d846bbeb83c652a0b1dff0e3519e67569c4a3d — github.com*, https://github.com/dmlc/xgboost/tree/82d846bbeb83c652a0b1dff0e3519e67569c4a3d, [Accessed 06-05-2024].

[55] Z. Wen, J. Shi, Q. Li, B. He, and J. Chen, "ThunderSVM: A fast SVM library on GPUs and CPUs," *Journal of Machine Learning Research*, vol. 19, pp. 797–801, 2018.

[56] *GitHub - vanloicao/SAEDVAE: This implementation for the paper: Learning Neural Representations for Network Anomaly Detection. If you have questions, please do not hesitate to contact me via email: Loi.cao@ucdconnect.ie OR loi.cao@lqdtu.edu.vn — github.com*, `github.com/vanloicao/SAEDVAE/tree/master`, [Accessed 20-04-2024].

[57] Martín Abadi *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: `https://www.tensorflow.org/`.

[58] F. Ceschin, M. Botacin, H. M. Gomes, L. S. Oliveira, and A. Grégio, "Shallow security: On the creation of adversarial variants to evade machine learning-based malware detectors," in *Proceedings of the 3rd Reversing and Offensive-Oriented Trends Symposium*, ser. ROOTS'19, Vienna, Austria: Association for Computing Machinery, 2020, ISBN: 9781450377751. DOI: `10.1145/3375894.3375898`. [Online]. Available: `https://doi.org/10.1145/3375894.3375898`.

[59] Z. Lin, Y. Shi, and Z. Xue, "Idsgan: Generative adversarial networks for attack generation against intrusion detection," in *Advances in Knowledge Discovery and Data Mining*, J. Gama, T. Li, Y. Yu, E. Chen, Y. Zheng, and F. Teng, Eds., Cham: Springer International Publishing, 2022, pp. 79–91, ISBN: 978-3-031-05981-0.