UNIVERSIDADE D
COIMBRA

Pedro Rafael Barata Ninharelhos Tomás

USING MACHINE LEARNING (ML) FOR ANOMALY DETECTION OVER TRAFFIC PRESENT IN SERVICE MESH ARQUITECTURES

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Prof. Dr. Alberto Cardoso and by Eng. Luis Cordeiro and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September of 2022

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

DEPARTMENT OF INFORMATICS ENGINEERING

Pedro Rafael Barata Ninharelhos Tomás

Using Machine Learning (ML) for anomaly detection over traffic present in Service Mesh arquitectures

Dissertation in the context of the Master in Informatics Engineering, specialization in Intelligent Systems, advised by Prof. Dr. Alberto Cardoso and Eng. Luis Cordeiro, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

September 2022

Acknowledgements

I would like to thank Prof. Dr. Alberto Cardoso for the availability and guidance throughout the period of this dissertation as well as to OneSource advisors Eng. Luis Cordeiro and Dr. Luis Rosa for the constant monitoring and advice. Additionally, I would like to also thank the OneSource team for the collaboration that was crucial for the development of this work.

My appreciation to all that one way or another, performed some role during my education, providing their contribution so I could reach this objective.

A word to my family that always encouraged me to pursue college, thank you!

A final note for a special person that was left without time and attention on countless occasions, during my college endeavors, through the highs and lows, thank you for your endless patience and support!

Abstract

The migration towards cloud-native applications has been increasing in the last years. The new generation of these applications tends to be more distributed, taking advantage of components running on the edge/cloud which brings a number of challenges, including their security. From a network standpoint, the high number of components and their complex communications difficult the process of detecting and mitigating cyber-attacks. To mitigate this problem, several security systems enabled with Artificial Intelligence (AI) components have been recently proposed in the literature.

This work aims to develop an AI component for anomaly detection as part of the design and development of a Holistic Security and Privacy Framework conducted in the context of 5G-EPICENTRE European Project. This framework was designed to automate and intelligently detect and mitigate anomalies in cloud-native applications following a service-mesh architecture.

To achieve this aim, several steps were taken. First, initial research has been conducted on several research topics such as Cloud-Native and micro-service orchestration, service mesh architectures, network security and network anomaly detection using AI techniques. In particular, different approaches were reviewed, namely ML and Deep Learning (DL) ones. Second, based on such an initial related work survey, several possible candidates were chosen for being implemented. Third, the requirements of the proposed approach were elicited and multiple use-cases were defined for use as part of the evaluation of each of the candidate techniques. Fourth, the methodology of the proposed approach was presented in detail, including it's several phases: the design, the implementation, the experimentation and the integration. Fifth, the selected approaches (Support Vector Machine (SVM), Random Forest, Convolutional Neural Network (CNN) and k-means) were implemented, trained and tested. Sixth, the approaches with better performance during the experimentation phase were implemented into the detection module of the security framework. Seventh, the correct behaviour of the implemented approaches was validated, namely through the use of a Graphic User Interface (GUI).

The implemented approaches with better performances and that evolve to the implementation phase were based on a Random Forest classifier and on a CNN, presenting these two approaches performance values (during the experimentation phase) near the maximum values for each metric, for all of the considered datasets.

Keywords

anomaly detection; machine-learning; network-security;

Resumo

A migração para aplicações que retiram vantagem dos serviços em *cloud* tem vindo a aumentar nos últimos anos. Esta nova geração de aplicações tende a ser mais distribuída, utilizando componentes que executam no *edge* o que traz uma série de desafios, nomeadamente, ao nível da segurança. Do ponto de vista de rede, o alto número de componentes e suas complexas comunicações dificultam o processo de detecção e mitigação de ataques cibernéticos. Para mitigar este problema, vários sistemas de segurança tem vindo a ser propostos na literatura, onde a maioria utiliza no processo de deteção. Este trabalho visa o desenvolvimento de um componente de para deteção de anomalias em tráfego de rede como parte do processo de design e desenvolvimento de uma framework de segurança desenvolvida no contexto do projeto Europeu 5G-EPICENTRE. Tal *framework* foi idealizada para automatizar o processo de deteção e mitigação de anomalias de forma inteligente em aplicativos adaptados para serem executados sob micro-serviços.

Para atingir esse objetivo, várias etapas foram percorridas. Primeiro, foi efetuada uma pesquisa inicial sobre vários tópicos, nomeadamente, Cloud-Native e orquestração de micros-serviços, arquiteturas de micros-serviços, segurança em redes e deteção de anomalias de rede usando técnicas de . Em particular, foram analisadas diferentes abordagens, sobretudo focadas em técnicas de *ML* e *DL*. Em segundo lugar, com base na pesquisa inicial, foram escolhidos vários possíveis candidatos para serem implementados. De seguida, de forma conjunta, foram definidos os requisitos e os casos de uso que a framework de segurança deve verificar. Em quarto lugar, foi definida a metodologia do trabalho a realizar, onde foram identificadas quatro fases principais: o planeamento, a implementação, a validação e teste e, por fim, a integração. De seguida, as abordagens selecionadas (SVM, Random Forest, CNN e k-means) foram implementadas e a sua performance foi avaliada. Em sexto lugar, as abordagens com melhor desempenho durante a fase de experimentação transacionaram para o módulo de detecção da *framework* de segurança. Por fim, foi validado o correto comportamento das abordagens implementadas, nomeadamente através da utilização de uma aplicação visual.

As abordagens implementadas com melhor desempenho e que transacionaram para a fase de implementação foram baseadas num classificador Random Forest e numa rede neuronal (CNN), apresentando ambas as abordagens valores de desempenho (durante a fase de experimentação) próximos aos valores máximos para cada métrica, para todos os conjuntos de dados considerados.

Palavras-Chave

detecção de anomalias, machine-learning, segurança em redes

Contents

Acronyms

5G-EPICENTRE ExPerimentation Infrastructure hosting Cloud-nativE Netapps for public proTection and disaster RElief. vii, 1, 3, 4, 27–29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 54, 105, 123

AI Artificial Intelligence. iii, vii, 1–3, 10, 12, 24, 25, 27, 33, 41–43, 45, 47, 48, 54, 59, 60, 105, 123

AICO Analytics Intelligence Control and Orchestration. xv, 46, 53

ANIDS Anomaly Network Intrusion Detection System. 19

ANN Artificial Neural Network. 8

CCC Command Control Center. 32, 33

CIA Confidentiality, Integrity and Availability. 38, 42

CNN Convolutional Neural Network. iii, v, xv, 20, 21, 58, 75, 77, 95–99, 105, 106

DEI Department of Informatics Engineering. 1, 45

DL Deep Learning. iii, v, vii, 11, 12, 15, 19, 20, 24, 25, 58, 60, 61, 105

doOCSVM distributed online One-Class Support Vector Machine. 16

DoS Denial of Service. viii, xvii, 41, 51, 52, 67, 78, 79, 81, 88, 91, 93, 95, 96

DPI Deep Packet Inspection. 41

DR Detection Rate. 16, 17

FAR False Alarm Rate. 16, 17

FCM Fuzzy-c means. 17

FCTUC Faculty of Sciences and Technology of the University of Coimbra. 1

FPR False Positive Rate. 19

GDR Global Detection Rate. 16

GM Gaussian Model. 16

GUI Graphic User Interface. iii, 125

HMM Hidden Markov Model. 18, 19

HSCBS Hyperspherical Cluster-Based Scheme. 17

HSPF Holistic Security and Privacy Framework. vii, xv, 3, 4, 27, 45–47, 54, 57, 62, 98, 100–102, 106, 107, 125

IDS Intrusion Detection System. 40, 41, 123

K-NN K-Nearest Neighbour. 9, 16

K8s Kubernetes. 33, 34

KPCA Kernel Principal Component Analysis. 17

LDA Linear Discriminant Analysis. 10

LOF Local Outlier Factor. 16

MEI Master in Informatics Engineering. 1

ML Machine Learning. iii, v, vii, 2, 4, 7, 11, 12, 15, 24, 25, 33, 42, 57, 58, 60, 61, 122, 123, 127

MLP Multi layer perceptron. 20

MsC Master of Science (degree). 45

NIDS Network Intrusion Detection System. 19, 123

OPA Open Policy Agent. 37, 53, 54, 100

PCA Principal Component Analysis. 10

PPDR Public Protection and Disaster Relief. 1, 2, 28–30, 106

PS Port Scan. viii, 78, 79, 85, 89, 92, 94, 96, 97

PSO Particle Swarm Optimiser. 18

RBF Radial Basis Function. 20

RDTIDS Rules and Decision Tree-Based Intrusion Detection System. 16

RL Reinforcement Learning. 11, 19

SME Small Medium Enterprise. 28

SMOTE Synthetic Minority Oversampling Technique. 63, 64

SNIDS Signature-based Network Intrusion Detection System. 40, 41, 54

SoA State of Art. 106

SVM Support Vector Machine. iii, v, xviii, 9, 16, 17, 20, 74, 77, 89, 91–93, 99, 105

TNR True Negative Rate. 16, 19

UC Use Case. 27–29

VNF Virtual Network Function. 44

VPN Virtual Private Network. 40

WCSS Within-Cluster Sum of Squares. 18

WSAN Wireless Sensor-Actuator Networks. 19

List of Figures

List of Tables

Chapter 1

Introduction

Conducted under the context of the dissertation of Master in Informatics Engineering (MEI) with the specialisation in Intelligent Systems, by Department of Informatics Engineering (DEI), Faculty of Sciences and Technology of the University of Coimbra (FCTUC), this work is being developed in OneSource, an IT company specialised into data communications, networking, security and systems management [1].

1.1 Context

Reliable communications and bigger field awareness over the field teams are two of the major needs under Public Protection and Disaster Relief (PPDR) activities. These are needed to achieve a higher level of security and efficiency in field operations. Some EU projects have been focusing on this thematic, intending to identify viable candidates PPDR technologies and architectures [2], and now, some are studying how 5G technologies may help improve the handling of situations, as is the case of the ExPerimentation Infrastructure hosting Cloud-nativE Netapps for public proTection and disaster RElief (5G-EPICENTRE) [3] project. Under this project, OneSource is responsible for the development of a network anomaly detection framework, enabled with a Artificial Intelligence (AI) component, able to deal with the high volumes of data associated with cloud-native paradigms.

The low latency, the enhanced capacity and the increased bandwidth are some of the characteristics that will enable reliable and faster communications, so much needed when facing a PPDR situation. The possibility of deploying 5G infrastructures on sight will also contribute to higher coverage, therefore, enabling continuous communication between field agents and controlling points.

The cloud-native approach aims to design, build, and run virtual functions by exploiting the cloud model, in which applications are developed using tools that make the most of its benefits. These benefits include greater agility in development, integration, and deployment, and are enabled by tools such as continuous integration, container engines, and orchestrators.

The large-scale automated management of fully software-based networks, such as 5G networks, makes it impossible to manually configure all the services. In addition to routing, load balancing, etc., automation of security and privacy policy enforcement is now required. The automation of security aspects may involve considering the application of AI techniques, namely, Machine Learning (ML) algorithms, in the identification of anomalies in traffic, due to their capability of processing big amounts of data and to provide accurate and on-time alerts.

In the scope of European research projects, OneSource is involved, along with other partners, in the development of a federated platform of 5G networks for the experimentation of point-to-point scenarios and in an open way, focusing on software solutions that meet the needs of PPDR activities, through the exploration of innovative approaches.

1.2 Motivation

5G is one of the most promising wireless technologies for the closest future. The technical enhancements when compared to 4G or 4G LTE will allow for a significant increase in the number of communications going through a network simultaneously.

5G applications are facing an evolution towards cloud-native systems, which allows these applications to take advantage of cloud-native properties, like flexibility, scalability, and reliability. The 5G core technologies are software-oriented, therefore, it's possible to treat them like common applications able to take advantage of cloud features, such as the possibility of deploying them anywhere, anytime.

Concepts like smart-cities, smart-homes, smart-vehicles and others, all share the same characteristic that is the presence of a huge amount of data being shared by the different involved components. 5G will play a crucial role in these paradigms, providing the network with the needed capabilities to accommodate such a high volume of data.

The concept of edge computing is not exclusive to 5G, but, with the continuous and predicted emergence of connected devices, it is foreseen to become even more common. This concept foresees the possibility of bringing cloud capabilities closer to the end-users, namely computation and data storage. By doing so, it is expected to improve response times and reduce the use of bandwidth [4].

In a paradigm with such big amounts of data, the thematic of security is a serious matter. It will be crucial to detect traffic anomalies, enforce security policies and take actions when those are breached. Mechanisms empowered by AI technologies will be game changing, since the high expected volume of data will no longer allow for a human analyse and detection of malicious traffic.

It's to tackle this issue of security under a cloud-native environment, more in concrete, under a Service-Mesh environment, that this work was conducted. The idealisation, development and integration of, at least, one ML solution to be inte-

grated with a Holistic Security and Privacy Framework is envisioned.

The development and validation of this framework are of extreme importance because of the active research activities that OneSource is engaged with, in peculiar, with a specific security task that OneSource leads as an active contributor to 5G-EPICENTRE. Further details on this may be found under section 4.4.

1.3 Problem Statement

5G technologies are foreseen to speed up the tendency of Big Data, predicted for traffic circulating in networks. To maintain the security of the network systems, automation processes will need to emerge, since such high volumes of data will no longer be compatible with human-based network security systems.

In order to implement an AI approach to detect traffic anomalies, what will it be necessary? Proper research must be conducted in aspects related to network core concepts and technologies, AI approaches and others.

1.4 Objectives

The objectives of this work can be summarised as the following:

- Review the existent literature on the topic of network anomaly detection and select, at least, 3 possible approaches to reproduce;
- Identify the set of features that must be collected from realistic communications within Mobitrust platform (properly described into section 4.4);
- Process a new dataset from realistic communications within Mobitrust platform;
- Apply and evaluate, at least, three approaches for network anomaly detection;
- Integrate, at least, one approach into the HSPF;
- Evaluate the existent detection component of the HSPF and suggest improvements;

The first and the second objectives include a thorough analysis of previous works in the area of traffic anomaly detection using machine learning. In the process, special attention will be given to the datasets that are commonly used for this purpose.

The third objective foresees the application of methods to process the dataset, namely, but not exclusively, to deal with missing values, finding the most discriminant features, dealing with unbalanced datasets, among others.

The fourth objective includes the development, testing and the first validation phase. The decision of which ML model(s) to be implemented arise from the knowledge obtained during the elaboration of the state of the art. It is supposed to be an informed decision, that should take into account several aspects, such as the availability of previous results obtained, the availability of the datasets used to test such algorithms, among others.

In order to successfully integrate the security solution to be developed with the other components of Mobitrust platform, foreseen in the fifth objective, it will be necessary to proceed to a continuous integration process, divided into two phases. The first, should culminate in the integration with the local deployment located at OneSource facilities, after the development of the solution, validation and tests phases, be concluded. The second embraces a complete integration stage where the security component will be ready to be deployed side-by-side with the remaining Mobitrust components. The evaluation of its performance is of high importance at this stage, since it's the moment when it will be tested in a controlled, but, similar to a real environment.

The final objective, encompasses the study of the current architecture and detection logic of the HSPF, with a special focus on its module of *Intelligence* and on presenting suggestions to improve it.

1.5 Structure of the Document

The structure of the document has been divided into multiple chapters in order to cover all the subjects associated with the planning, development and integration of a ML algorithm for traffic anomaly detection.

Chapter 2 starts by presenting some important core concepts for the realization of this work: the five tribes of ML algorithms, a technique used during preprocessing of datasets, the taxonomy of AI anomaly detection techniques, where the different aspects of the thematic in hands are overviewed and some are further described, and finally the performance metrics that will be used to evaluate the performance of the implemented approaches.

Chapter 3 presents a review of the state of the art around traffic anomaly detection approaches, with a special focus on ML algorithms.

Chapter 4 presents the 5G-EPICENTRE project under which this work is developed and also the OneSource security framework where the developed solution will be included. This is followed by the presentation of some core technologies commonly used with Service Mesh architectures, traditional types of attacks and counter-measures, and also the role of AI in fairly recent network security systems. The 5G-EPICENTRE security task (T2.6) is described, as well as the existent HSPF, developed to provide security to vertical applications deployed as micro-services.

Chapter 5 presents the methodology to be followed during this work, including a description of its major phases: design, implementation, testing and validation

and integration.

Chapter 6 illustrates the work carried out during the implementation phase, namely the phase of pre-processing the datasets and the implementation of the candidate approaches.

Chapter 7 reflects the results attained by the implemented approaches, including its proper analysis and an overall discussion.

Chapter 8 concludes this work, reflecting some conclusions drawn throughout this work.

Chapter 2

Background Knowledge

This chapter aims to present some core concepts and techniques associated to the thematic of ML, as well, to the topic of anomaly detection using intelligent systems. Guidelines were retrieved from the present content, that is further developed in chapter 3 with the presentation of several approaches focused on the problem of network anomaly detection (namely, using ML).

Section 2.1 focuses on presenting the different machine learning tribes defined at [5], section 2.2 presents an overview of a technique used in the pre-processing of datasets, section 2.3 provides an overview of the taxonomy of anomaly detection (based on [6]) and lastly, section 2.4 states the performance metrics used during the evaluation of the implemented approaches.

2.1 Machine Learning Tribes

Five tribes of ML approaches are defined at [5]: Symbolists, Connectionists, Evolutionaries, Bayesians and Analogizers. Some of the types of algorithms mentioned in the next lines are further described in section 3.1.

- Symbolists

  Symbolic AI involves the embedding of previous knowledge (human) and behaviour rules into decision processes. The structure of this algorithms can be described as a series of connections, that can be represented by symbols, which are latter used to explain the rational behind the decisions making process.

  The most effective technique that symbolic AI presents is the use of inverse deduction, which makes possible to understand the needed missing inputs to be able to extrapolate knowledge from a set of known premises and conclusions.

  Decision trees are a common approach of this type.

- Connectionist

  Connectionism, an approach of AI, was developed while trying to understand how the human brain works at the neural nevel and, in specific, how people process and memorizes information. This approach is the one responsible for the impulse that deep learning has been facing over the past years, with the exploration of the Artificial Neural Networks (ANNs) explored.

  Since this approach considers that the knowledge is attained by learning, knowing the results it's possible to adjust the strengths of the connections between the different neurons. To do this, it is necessary to know which connections are responsible for certain under performances and tune them accordingly. Thus, this process is known by reverse engineering in a sense that the process is made backwards, while seeking for the perfect values for the parameters. Due to the nature of this process, the AI models originated from here are often called as black boxes, since it is not easy to understand the different values that the data takes throughout the model.

- Evolutionaries

  Evolutionary AI is based on the behaviour of living beings, specially those that usually leave in group and work towards its greater good (e.g. ants, bees, etc). It involves mechanisms like reproduction, mutation, recombination, ultimately associated with biological evolution.

  A major characteristic of this type of algorithms is the continuous search for the most suitable candidate. This process is dependent on a fitness function that is used to evaluate the fitness of all the candidates that are part of a population. After, a Darwin base approach is followed to select the set of individuals that continue to the next iteration. Thus, these type of approaches don't focus on adjusting algorithm parameters, instead, they focus on the creation of an intelligent structure (similar to a brain) that, when facing a problem, uses a set of individuals to find the optimum solution for it. It is correct to say that the focus of this approaches is on genetic programming, in a sense that the evolution is achieved through the manipulation of individuals and respective components, similar to the manipulation of DNA conducted by scientist while studying a set of organisms.

- Bayesians

  Bayesian based approaches are based on the Bayes' Theorem. Here, the main goal is to estimate a set of distributions that than support the creation of a probabilistic model. Mainly used for classification purposes, this type of approaches uses the likelihood of a sample to belong to a certain class and perform the classification accordingly.

  The calculus of probabilistic inference is used to deal with noisy, incomplete (and even contradictory data) and also to incorporate new knowledge into the model.

- Analogizers

  Analogizers represent a set of AI approaches that try to establish similarities among different samples. Commonly used in situations where the new samples present similarities with previously seen ones, the main challenge on this type of approaches is respectively the process of recognizing the level of similarity between the new samples and the previous seen ones.

  Common approaches of this type are K-Nearest Neighbour (K-NN) and SVMs based approaches.

2.2 Feature Selection and Reduction

Feature selection and/or reduction are often used to deal with datasets containing a high number of features, during the pre-processing stage, before proceeding with the training and assessment of the algorithms.

At [7], the authors focused on investigating the importance of pre-processing the dataset, covering different techniques, namely, feature reduction and normalization, while using the J48 classifier and the NSL-KDD dataset.

Some reasons that support the application of feature selection and/or reduction techniques, are the following [8]:

- It's common that the initial dataset contains several features that are unrelated, redundant, not relevant or even repeated;

- The higher the number of features, the higher the dimensionality of the problem, thus, the higher the amount of data needed for the algorithm to be able to correctly classify unseen samples (and don't fall into *overfiting* situations);

- Allows a better data visualization enhancing its *understandability*;

- Commonly contributes to the reduction of the algorithm's training time;

Feature selection corresponds to selecting a set of features without transformation from the original dataset, while feature reduction is more focused on producing a smaller number of features, from the transformation of the original ones. There are supervised and non-supervised approaches for both these processes, as well as different subtypes of methods.

Figure 2.1 further illustrates the difference between feature selection and feature reduction.

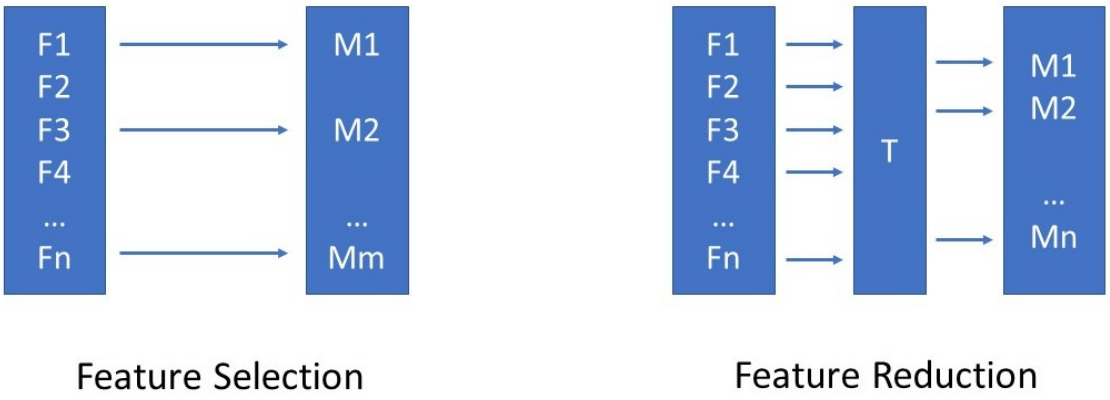


Figure 2.1: Feature Selection Vs Feature Reduction

For the feature selection, there are three main categories: filters, wrappers and embedded. Filters consist of a set of pre-processing methods that are independent of the classifier to be used, whereas wrappers do depend on the classifier. Embedded ones, join the feature selection with the classification.

Some supervised methods for the feature selection are the Kruskall-Wallis H test [9] and AUC analysis, while some non-supervised ones are: Variance analysis and redundancy analysis.

During the feature reduction the physical meaning is lost since the dimension will be altered and this is achieved mainly by two methods: Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) [10], being the first considered a supervised approach, and, the second one a non-supervised approach.

2.3 Taxonomy of AI Anomaly Detection Techniques

The use of AI technologies to detect traffic anomalies has been increasing in the past years. There are several aspects to consider when planning the implementation of a traffic anomaly detection system, being the most relevant: the AI techniques to be used, the nature of the data, the type of anomaly, the algorithm learning mode, the window used for information transmission, the content of the datasets and the model's evaluation criteria.

Figure 2.2 summarises the most common options for each of the mentioned aspects.

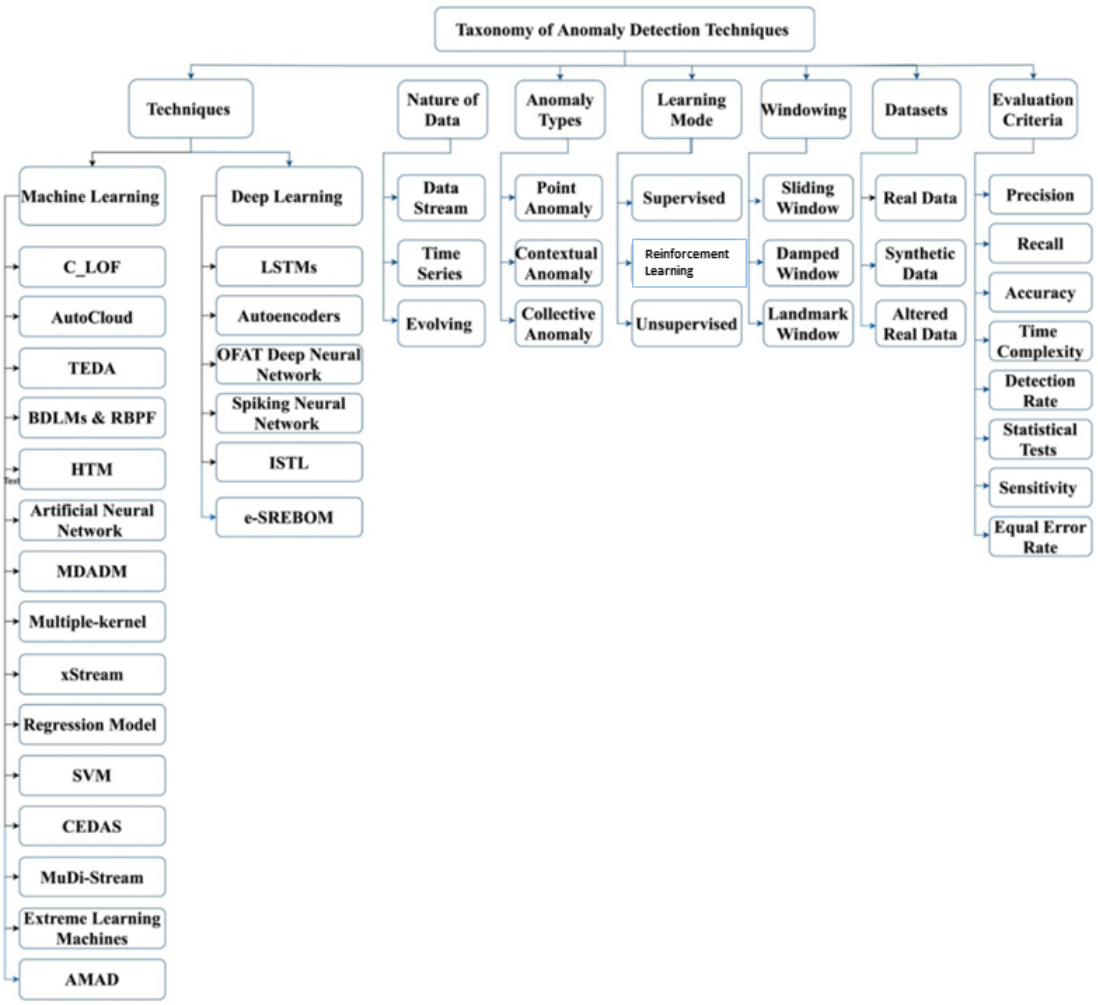


Figure 2.2: Taxonomy of Anomaly Detection Techniques [6] (edited)

There are numerous of different approaches in the literature (some overviewed in section 3.1) that focus on the problem of traffic anomaly detection, however, the algorithms applied into each one, all fit into one of the following categories: generic ML approaches or Deep Learning (DL) approaches.

ML algorithms can be divided under three types [11]: (i) Supervised; (ii) Unsupervised; (iii) Reinforcement Learning (RL). Algorithms that follow a supervised approach are usually associated with classification problems, in a sense that, from an initially labelled dataset each algorithm is capable to learn the distinction among different classes, using a defined set of attributes. Algorithms that follow an unsupervised approach are usually associated with clustering techniques or indirect classification. Unlike the supervised approaches, this type of algorithms do not require an initially labelled dataset. The distinction is based on clustering techniques supported by several mathematical expressions used to calculate the likeness between data instances. Finally, the reinforcement-learning approaches comprehend the set of algorithms that present a continuous evolution based on a action-reward schema, where the algorithms evolve by seeking to maximize the received reward and, as such, adapting their actions accordingly.

DL algorithms have been lately suggested for the thematic of traffic anomaly de-

tection, and several have been presenting good results. As a concrete sub-area of ML, DL techniques focus only on neural network-based algorithms. Nevertheless, there are still DL techniques that fit into a supervised approach and others that fit into an unsupervised approach. Some approaches are overviewed in section 3.2, due to the promising value revealed in some of the recent works.

The use of DL techniques are often restrained by the heavy computer requirements, needed to process the big amounts of data needed to train DL algorithms. To help in this process, the *transfer learning* [12] technique may help. This method allows applying knowledge obtained from the resolution of a previous problem into the resolution of a new and similar problem. Neural networks can be composed of a set of multiple inputs, outputs and hidden layers, which may allow the models to learn from the processing of input data. Such complexity is often translated into higher performances while dealing with a problem, when compared to non DL approaches [12].

The anomaly detection technique to be used depends on the nature of the data that will be used. There are three major types: data stream, time-series and evolving. In the context of traffic anomaly detection, the most common is a data stream. A data stream is a continuous series of data records, that may be ordered or processed using timestamps [6].

The type of anomalies found in the most diverse anomaly detection systems fit into one of the following three types: point anomaly, contextual anomaly or collective anomaly [6]. Point anomalies can be detected by analysing data occurrences that fall outside of normal patterns, therefore, presenting themselves as *outliers*. Contextual anomalies are characterised by occurring in a specific context and their detection relies on the comprehension of the meaning that its occurrence in that specific context may have (e.g., traffic jams). Collective anomalies are usually detected through the analysis of a set of continuous anomalies that may occur frequently in a specific environment, thus, the analyse of several time periods is needed to identify this type of anomalies (e.g., heart malfunctioning).

The learning modes that an algorithm may follow, are already overviewed above, while is presented a possible division for ML algorithms.

To help the algorithms process data sources, *windowing* techniques may be used. This can be divided into three types: Fading (Damped), Landmark and Sliding. Details on each of these windowing models can be found at [13].

The origins of commonly used datasets are also identified: real data, synthetic data and altered real data. A real data dataset usually contains data collected from real world environments, while an altered dataset usually reflects the result of data transformations from a real data dataset. A synthetic dataset is the result of an artificial process to generate data, usually through the use of programming scripts.

The evaluation of an AI approach is generally done using performance metrics. Some of the identified metrics are further described in section 2.4.

2.4 Performance Metrics

The metrics that will be used to evaluate the algorithm(s) performance are *accuracy*, *precision*, *recall* and *F1* measure. They take into account the notion of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN), and their formula's are:

- Accuracy = $(TP + TN)/(TP + FP + FN + TN)$
- Precision = $TP/(TP + FP)$
- Recall = $TP/(TP + FN)$
- F1 = $(2 * TP)/(2 * TP + FP + FN)$

The confusion matrix present in figure 2.3, illustrates the meaning of each of the elements in the previous formulas.

Model predicts / Ground Truth	**Negative**	**Positive**
Negative	TN	FP
Positive	FN	TP

Figure 2.3: Confusion Matrix

During the experimentation phase and due to the application of a search grid, as described in chapter 6, for one of the algorithms, the value of R^2 [14] was also considered during the assessment of the algorithms performance for some intermedium experiments. Despite this, the best classifier attained for each experiment, was evaluated using the same set of metrics (previously presented).

Furthermore, other performance indicators were also considered, such as the algorithms training time, the algorithms classification time, the features pre-processing time, among others. How these indicators were considered and which impact did they represented over the final considerations, is presented in chapters 6 and 7, referring to the Implementation and to the Result's chapters, respectively.

2.5 Summary

The purpose of this chapter is to highlight some core concepts and techniques relevant for the work in hands. The first section (2.1) starts by identifying the

machine learning tribes: symbolists, connectionist, evolutionaries, bayesians and analogizers, according to [5]. Afterwards, section two (2.2) presents a common technique during the pre-processing of datasets, that is the feature selection and/or reduction.

Section three (2.3) presents an overview of the taxonomy of anomaly detection (based on [6]), where ML algorithms are divided into three categories: supervised, unsupervised and reinforcement learning. In addition, several other aspects are abroad like: the anomaly types (point, connected and collective), nature of the data (data stream, time series and evolving), nature of datasets and evaluation criteria, among others.

The last section of this chapter (section 2.4) presents the performance metrics that were considered during the evaluation of the implemented approaches, which results are presented in chapter 7.

Chapter 3

Literature Review

This chapter aims to complement the information presented on the previous one, throughout the review of different approaches that focused on the thematic of network anomaly detection, with the use of ML and DL. Being the main focus on ML, four approaches are presented for supervised and unsupervised learning approaches and other two for reinforcement learning.

Furthermore, two approaches that use DL are also reviewed, plus some other hybrid ones. An overview of the most commonly used datasets for network anomaly detection presented, as well as a few common steps taken during its pre-processing.

3.1 Network Anomaly Detection using ML

Considering the panoply of existent ML approaches for anomaly detection, this section focus on the presentation of some of the existent approaches. It is structured in order to reflect the division highlighted in section 2.3, where ML approaches are divided into supervised, unsupervised and reinforcement learning.

3.1.1 Supervised

Following are presented some types of the most common supervised algorithms, as well as some previous works where supervised approaches have been followed.

Decision Tree (based)

A Decision tree algorithm is based on attempts to approximate the value of a discrete function. Essentially, the algorithm classifies each sample according to a set of rules. Each sample is tested with a set of different rules, which results in a tree shape classification. Decision Tree-based algorithms have been used

individually to classify some types of attacks [15], but more recently this type of algorithms has been included in hybrid classification systems.

The application of an Extra Tree classifier is conducted in [16]. This algorithm is applied over four well-known datasets: UNSW-NB15, BoT-IoT, ToN-IoT and CSE-CIC-IDS2018, and through the transformation of these four - where the *NetFlow v9* features were extracted - four new datasets were created.

The Rules and Decision Tree-Based Intrusion Detection System (RDTIDS) corresponds to a hybrid classification system proposed at [17]. This system corresponds to a hierarchical model, built over two layers. It considers two classifiers in the first level, where each one considers a different set of features, and a third classifier that besides receiving as input the full set of features, also receives the classification of the two first classifiers, for each sample. One of the classifiers in the first layer corresponds to a tree based approach, in concrete to a REP Tree approach [18].

The datasets used to evaluate the performance of the proposed system were the CICIDS 2017 [19] and the Bot-IoT [20], and the metrics considered were False Alarm Rate (FAR), Global Detection Rate (GDR), accuracy, training and test time. The authors concluded that the proposed system was able to provide the highest True Negative Rate (TNR) and highest Detection Rate (DR) for seven types of attacks, as well in global terms, it presents one of the highest accuracy's and lowest FAR, when compared to different methods proposed in other papers, which results are presented in a table for easier performance comparison.

SVM

A SVM attains to classify the data through the definition of one or multiple hyperplanes, depending on if the SVM in cause aims to produce a binary or multi-class classification. While computing the equations of the hyper-planes, when the data is linearly separable this calculus is efficiently completed, otherwise, it's necessary to project the data for a higher dimension space where this computation can be done in a simplified way. A thorough investigation over the state of the art in terms of SVM-based algorithms for anomaly detection is conducted over [21]. Two approaches are described in the following paragraphs.

A binary classification model is presented at [22]. Two versions of the distributed online One-Class Support Vector Machine (doOCSVM) are proposed and both share a peculiar aspect: during the training phase, only positive samples are considered. According to the authors, these approaches are able to detect the anomalies recurring to mathematical expressions, more concrete through the definition of hyperplanes that represent the barrier between normal and abnormal data. The doOCSVM approaches were evaluated using created synthetic datasets and a set of datasets belonging to the UCI Machine Learning Repository [23].

In order to evaluate the performance of the proposed algorithms, a set of well known classification algorithms have also been tested with the same datasets, namely, Gaussian Model (GM) [24], K-NN [25], Local Outlier Factor (LOF) [26],

Kernel Principal Component Analysis (KPCA) [27] and Hyperspherical Cluster-Based Scheme (HSCBS) [28]. It was concluded that the algorithms in analysis outperform some of the reviewed algorithms for some datasets, while for other datasets, the attained results are closest to the best. A special analysis has also been given to CPU and memory consumptions, and it has been outlined that both the proposed algorithms presented better performances due to their internal characteristics.

At [29], the authors propose a new intrusion detection system, entitled *DT-EnSVM*, that combines ensemble learning and transformation techniques with SVM. These techniques are described and some theoretical advantages of their use are highlighted. The first layer of the intrusion detection system is composed of a set of SVM's that are fed by heterogeneous training data, prepared by a modified clustering algorithm Fuzzy-c means (FCM) that allows for samples of data to coexist in two or more clusters. The second layer consists of a single SVM that receives the output of the SVMs in the first layer and provides the final classification of the system.

The proposed approach was evaluated using the NSL-KDD, KDD'99 and Kyoto 2006+ datasets. A revision of the performance of similar approaches over the mentioned datasets is conducted, with a special focus on *accuracy*, DR, FAR, and training speed. It is concluded that the approach in analysis is not only able to attain a superior and more robust performance but it also presents a lower training period.

3.1.2 Unsupervised

Following are presented some types of the most common unsupervised algorithms, as well as some previous works where unsupervised approaches have been followed.

k-means

k-means is an iterative algorithm that aims to separate the detaset into K (predefined) number of non-overlapping distinct clusters. Being K a predefined value that constitutes itself an issue, in a sense that this constant needs to be carefully calculated, otherwise, the risk for poor performance is extremely high since the algorithm will start from a wrong assumption. To handle this, several initialisation methods have been proposed [30].

After the initialisation step, each sample is assigned to the nearest cluster recurring to mathematical expressions that evaluate its distance towards the centroid of each cluster. Some of the most common mathematical formulas used for this, are Euclidean Distance, Mahalanobis Distance, Correlation, etc.

At [31], the authors compare the performance of two variants of the *k-means* algorithm: streaming-k-means and batch-k-means. It is stated that conventional analysis methods are not able to process big amounts of data due to resource

limitations. On other hand, batch and stream processing techniques are able to process such amounts of data in short periods of time, especially considering that stream processing techniques are able to process data in real-time.

The problem of 'K' initialisation is handled recurring to Within-Cluster Sum of Squares (WCSS) technique and the experiments are conducted over a MovieRate dataset. Those, reveal a better performance for the Batch approach, when compared with the Stream one, for the same number of train executions. Despite this, it is also concluded that the performance of the streaming approach improves considerably after the initial training executions. Therefore, it is believed that, if trained enough, this approach will be able to present similar results to the ones obtained by the Batch approach, but, with the advantage of being able to process the data in real-time.

The performance of a *k-means* algorithm is compared with the proposed approach that encompasses a *k-means* implementation as a first step and as second, a Particle Swarm Optimiser (PSO) algorithm [32]. PSO is presented as a possible solution for common clustering problems: the clustering accuracy, the predefined number of cluster centres, the presence of many local minimum points that may invoke the algorithm into converging to a set of clusters centroids that may not be the most appropriate ones. A summit of previous applications of PSO algorithm is summarised in a table, indicating for each, the dataset, criteria, parameters values, cost function and major approach contributions.

The performance of both approaches is evaluated recurring to Dunn's index [33], Silhouette index [34], Purity index and Entropy index [35], and the dataset used was the Yahoo! S5 [36]. The authors conclude that the proposed approach, despite not presenting the best values for all the metrics considered, performs better than the algorithm (*k-means*) alone.

Hidden Markov Model (HMM)

HMM models correspond to doubly stochastic finite models that calculate the probability distribution over an enormous amount of possible sequences. These models are marked by the possibility of random state changes, and also by the fact that the transition for the next state is only dictated by the current state [37].

Over the past years several approaches have been presented under several areas, namely, the High-Order HMM, the Hidden-Semi Markov Model, the Layered HMM, among others. Each of these approaches represents an attempt to improve the performance of the existent HMM algorithm to a specific area. Under the application areas of *HMM* models, it´s possible to find speech recognition, human activity recognition, musicology, data processing and, as in minor percentage, network analysis, which will be the focus of the following papers.

A multi-layer HMM model is applied to traffic anomaly detection at [38]. The multi-layer approach arises as a possible solution to one of the biggest challenges that this type of algorithm faces: the course of dimensionality [39]. The entire journey that the data must traverse until be ready to be handled by the algorithm

is described in detail, while also an overview of the common datasets used to evaluate Network Intrusion Detection System (NIDS) is conducted, with the authors pointing several datasets as outdated and finally highlighting the content of the one that will be used, the CIC-IDS2017 dataset [19].

This approach presents values for accuracy, precision, recall, f1_measure all near 1.0, being 0.9793 the lowest value, that corresponds to the precision attained. The authors state that the use of a multi-layer HMM presents several advantages when compared to a single-layer approach: (i) a single-layer needs to be trained on a large number of observation spaces, which might lead to over-fitting situations in cases where enough data is not used, while in a multi-layer approach, each layer may be trained in small-dimensional observation spaces, which don't require a high number of training data and often result into more stable models; (ii) when using multi-layer, each layer is trained independently, opening space for different combinations of HMM models at different layers, thus enabling to aim for a better understanding on the nature of the data; (iii) multi-layer approaches may be expanded in the future, through the addition of more layers, to cover new network traffics.

The efficiency of a HMM model is compared to the efficiency of a cumulative sum (CUSUM) approach [40] in [41]. The authors address a type of a possible vulnerability found within a specific mechanism of LTE signalling for Wireless Sensor-Actuator Networks (WSAN) networks, the *wakeup* packets, used to communicate state changes among mobile network stations and sensors. The data used to evaluate the algorithms was collected from a set of real data, gathered from the system benignum traffic.

The authors state that the proposed approach presents lower False Positive Rate (FPR) and higher TNR, when compared to the CUSUM approach, for the three types of attacks considered. It is also concluded that in order to present the same TNR, CUSUM presents a FPRs around 45%.

3.1.3 Reinforcement Learning

The RL paradigm contains a set of key concepts: agent, environment, state, policy, reward and state/action value function [42]. The agent lives within the environment and receives inputs from it, in a discrete way, that later maps to state information. The agent executes actions and receives rewards according the correctness level of these actions. The evolutive process of the agent is mainly divide under two moments: the first, while the agent acts based on states and collects rewards, and the second, where the agent tries to comprehend the changes on the environment and react accordingly, aiming to maximize its reward. The purpose of a DL algorithm is to find the optimal policy that allows to maximizes the value of received rewards.

A multi-mode approach implemented by an Anomaly Network Intrusion Detection System (ANIDS) is proposed at [43]. The authors highlight the possibility of the mechanism to self-updating in run time so the algorithm may also be aware of the most recent patterns, thus, being able to sequentially train and classify. The

datasets used to validated the performance of this approach were the NSL-KDD and the UNSW-NB15 datasets, which are briefly described. The validation of the proposed approach was also conducted with a collected dataset from realistic campus network, whose collection process is presented in detail.

The performance of this approach was compared with the ones attained by Multiplayer perceptron, Random Forest and SVM, for the selected datasets. It is stated that the proposed approach for NSL-KDD dataset outperforms the remaining approaches for accuracy and recall, while for the UNSW-NB15 dataset, the proposed approach achieves similar results.

At [44] the authors present an approach where two reinforcement learning algorithms are used simultaneously. The first corresponds to the algorithm being trained, while the second, upon some time in training, is used to fed the first with the most difficult samples, while trying to provoke a wrong classification. The authors apply a symmetric system of rewards, thus, if the first algorithm correctly classifies a sample it will receive a positive reward, while the second algorithm will receive a negative one.

The performance of this approach is evaluated with the NSL-KDD and AWID datasets and later compared with the performances attained by other approaches based on Radial Basis Function (RBF), SVM, Multi layer perceptron (MLP), Random Forest, CNN, among others. The authors state that the implemented solution outperforms all ML approaches and presents similar results to remaining state-of-the-art classifiers. The innovative character of this approach is highlighted, the short training time of this approach is also underlined, making it suitable for online prediction, as well its capacity to deal with unbalanced datasets, supported by the specific training mechanism.

3.2 Network Anomaly Detection using DL

At [45] an approach based on a feed-forward neural network is followed. The authors' approach encompasses a binary and multi-class classification problem. During multi-class classification, the network can detect denial of service (DoS); distributed denial of service (DDoS), reconnaissance and information theft attacks.

For detecting anomalies in wireless mesh networks, they mention several existing approaches: support vector machine, Bayesian network, principal component analysis and genetic algorithms. The use of a real dataset is also worth mentioning, as well, as the comparison of performance obtained with a benchmark solution based on SVMs.

On [46], a deep analyse is conducted on approaches based on Long Short-Term Memory (LSTM) algorithms. The authors claim that the use of LSTM alternatives, instead of RNN approaches, e.g., enables systems to learn and recognize relations that occur over a long period. It is done an overview focused on LSTM encoder-decoder-based approaches, hybrid approaches, graph-based and trans-

fer learning.

Regarding the mentioned approaches, it is presented a list of remaining open challenges, such as the input data structure over graph-based approaches; the selection of contextual features within the graph and the lack of existent datasets, which makes the comparison between two algorithms almost impossible, due to the missing benchmarks.

At [47] the authors present an approach based on a neural network composed by *Perceptron* based layers. Such neural network is applied to the CSE-CIC-IDS2018 dataset and the obtained results raise questions related to possible overfitting by the system. Despite that, the authors claim that the same network can be used to identify the following type of attacks: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and Infiltration of the network from inside.

A CNN based approach is proposed in [48]. The authors took inspiration on a CNN approach initially used for image recognition and proposed 3 different configurations for their solution-oriented to anomaly detection. The performance of the mentioned approach is validated recurring to the BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020 and IoT-23 intrusion detection datasets.

A thorough description of the neuronal network used is conducted using a set of illustrative images, while the datasets used are also subject to proper description. The authors state that the proposed approach achieves higher performance than existing classification strategies, being the minimum of *accuracy* achieved for the 4 datasets of 99.03%.

3.3 Hybrid Approaches for Network Anomaly Detection

A combination between a *k-means* clustering algorithm and a genetic algorithm has been addressed at [49]. During the initial steps, the KNN is applied to the several features in analyse in order to determine to which class each occurrence of each feature resembles the most. After that, those occurrence values are normalized between [0 to 1] and these new values are used at the chromosomes of each individual in the GA population.

Using this method, for DoS/DDoS attacks, the authors were able to achieve 97.42% of accuracy for known attacks detection and 78% of accuracy for unknown attacks.

Mingzhu Tang *et al* [50] present an approach based on a Humber-Ridge model with a particle swarm optimizer (PSO) algorithm in order to achieve the optimum hyperparameters needed for the mentioned model. A list of the steps followed is present, as well, as an explanation of the mathematical formulas behind the considered model. This work aimed to evaluate the performance of the PSO algorithm when compared with GWO, GA and STA optimisation algorithms, for the same purpose.

A comparison between a ML detection algorithm, *Boosted Decision Trees*, and a deep learning approach based on *Simple Feedforward Neural Networks* is presented at [51]. Using two *datasets*, one simulated and another one created from the collection process of registering the packets flow between PIC and CERN [1].

The performance of both algorithms has been satisfactory. Despite that, the authors have shown a preference for *BST trees* stating that these present a higher processing speed and also considering that they allow a better tunning when compared with the studied feed-forward neural network.

Several methods for anomaly detection, using deep learning are overview at [52]. The authors present an overview of the existent methods. These, are divided into 3 major types: Deep Learning for Feature Extraction, Learning Feature Representations of Normality and End-to-End Anomaly Score Learning.

The first represents a set of algorithms that aims at extracting low-dimensional features from high-dimensional or non-linearly separable features and where the anomaly scoring is completely separated from the feature extraction part. The second englobes methods where the feature learning and the anomaly scoring are coupled in a certain way. More specifically, this group can be sub-divided into two major groups: Generic Normality Feature Learning and Anomaly Measure-Dependent Feature Learning. The first group englobes methods that learn the representation of data by optimizing an objective function since they are compelled to capture key data regularities. The second group covers algorithms that are developed focused on the learning of features representations for recognized anomalies. The biggest difference is the presence of a concrete anomaly that is passed to the objective function, in the second group, while in the first, the anomaly scores are simply based on heuristic processes applied over the learned representations.

Finally, the third approach differentiates itself by not being dependent on existing anomaly measures, since it contains a neural network that directly learns the anomaly classifications. Such neural networks are commonly based on novel loss functions which leverage the learning of the features representations and, simultaneously, of the anomaly scores.

In [6] a graph is presented with the taxonomy of anomaly detection, containing: techniques, nature of data, anomaly types, learning mode, windowing, datasets and even evaluation criteria. Regarding techniques, two are presented: Machine Learning and Deep Learning. In terms of data nature, the most common types of sources are listed: Data Stream, Time Series and Evolving. After, the several types of anomalies are described in-depth, recurring to graphics analysis in order to explain the difference between point anomaly, contextual anomaly and collective anomaly. Some approaches of learning techniques are further described, such as (i) supervised, (ii) semi-supervised and (iii) unsupervised.

An interesting approach of Window Models is presented as long with the description of three major types: Fading, Landmark and Sliding. The authors also mention some real-world datasets and a detailed report of the experiments con-

[1] CERN is a Tier-0 site in Geneva, Switzerland, PIC is a Spanish Tier-1 centre.

ducted is also present, from which it is possible to highlight a table containing information about the performance of each tested algorithm, in terms of Noisy Data Handling, Time Limits, Memory Limits, Evolving Data Handling, among others.

3.4 Existing Datasets and Feature Engineering

This section aims to provide an overview of the most common datasets used for the validation of traffic anomaly detection systems and to make a succinct description of the entire process associated with the traffic processing and respective analysis.

3.4.1 Anomaly Detection Datasets

Datasets are frequently used by the authors when proposing a new approach. From the literature review, the most commonly used are CSE-CIC-IDS2018, BoT-IoT and CIC-IDS2017. CSE-CIC-IDS2018 is the result of a partnership between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC), therefore can be seen as an evolution of the CIC-IDS2017, considering that it was elaborated by the same authors that actively gather representative network traffic of the different type of attacks present in these datasets.

At [38] the authors conduct an exhaustive assessment of the most used datasets for anomaly detection and select CIC-IDS2017 as the best option, due to the types of attacks considered, but also, due to quality of the data (no repeated samples or redundancy registered among the samples, among other factors). Taking this into account, CIC-IDS2017 will be used to validate the approach that will be implemented. CSE-CIC-IDS2018 could also be selected for this effect, but, more literature reviews described CIC-IDS2017 in detail and used it to validate the proposed approaches, therefore, in order to have a bigger comparison baseline, 2017' version is envisioned to be used for the mentioned purpose.

Several authors, when validating their approaches, not only execute experiments with the chosen datasets, but also, compare the metrics values attained with similar values achieved by other approaches [30], [53], [32], [17], [29], [22].

An important observation over the validation of algorithms is stated at [16]. Here, the authors defend the need for a well defined set of features for each known publicly available dataset. Such a set of features would be crucial to evaluate the performance of different algorithms. It is also claimed that the existence of such a set of features, would help fill the gap between academic research and real market solutions, given the fact that it would allow a concrete evaluation and comparison between different algorithms, which, nowadays are compared in a non-accurate away, since, in each experience, different authors use different features, which makes the comparison not possible or of very low importance.

3.4.2 Common steps pre-processing datasets

This section aims to describes several phases inherit to the process of data collection and processing when dealing with network traffic analysis in order to detect anomalies, according to [54]. Must be noted that not all the approaches include the phases described neither by the presented order. Despite this, such phases are present in the most of the approaches.

Dealing with packet processing is usually one of the first steps, namely by extracting the relevant feature values from each packet, dealing with missing values, dealing with *outliers*, proceed to data normalisation, among others.

Next, proceed with feature selection and/or reduction (previously overviewed in section 2.2). Since traffic packets are composed of an enormous amount of fields, it is necessary to select a set of these fields to use as the features for the dataset. Furthermore, some algorithms can achieve better performances if the *dimensionality* of the problem is relatively small. Therefore, it is very common to evaluate the amount of information that each feature provides and then generate a new data object containing only the most relevant features.

Afterwards is the classification itself. In this phase, an AI algorithm is usually used to classify the input data. A set of approaches is overviewed in the previous sub-sections.

Then, there is the creation of the dataset object, by registering into a database the several features values of each sample plus the respective classification (label).

When a classification system is already executing for some time and a database with previous knowledge is available, it is also common to compare the current input data with previous data occurrences. Naturally, this comparison aims to allow the algorithm to make a more accurate decision, with the help of previous knowledge.

It is worth mentioning that the dataset that will be generated from realistic communications will partially follow this workflow, being the full process described in section 6.1.

3.5 Summary

Research activities were conducted to find approaches that have been proposed for the problem of anomaly detection. The relevant approaches found were summarised and organised taking into account the ML algorithms division (supervised, unsupervised and reinforcement learning). It is inferred that the newest approaches tend to contain more than one algorithm and often the involved algorithms are organised hierarchically, thus, the output of some algorithms is used as input for others. It was also noticed that the most recent approaches are focused on unsupervised approaches instead of supervised ones, and that there is an increase of approaches that start to explore the use of DL techniques to solve the problem in hands.

The most frequently used datasets to validate AI models are inferred from the reviewed literature and a generic process that the data needs to be subjected to, is briefly described (in a common traffic anomaly detection system).

For the development of the AI component, a supervised and an unsupervised ML approach will be selected, as well, as a DL approach. The approaches envisioned to be used as reference during the implementation phase are: [32], [29] and [48]. Due to the lack of a concrete away to compare the several approaches, the conducted selection was made having in mind the mindset for keeping the biggest diversity possible. The missing of a comparison form to compare different algorithms is addressed into one of the reviewed articles, where the authors claim that since there is no common set of features, for each dataset - used by all the authors when evaluating their approaches - it's not possible to precisely compare the performance of such approaches.

Chapter 4

5G-EPICENTRE

This chapter aims to present the context where this work is being developed, as well, as some progress already achieved.

This work is conducted under a European Project (5G-EPICENTRE [3]) and meets the needs of one of its tasks related to security aspects, more in concrete, with the development of an AI component to integrate the detection module of the HSPF responsible for detecting traffic anomalies, present in traffic of service-mesh based applications. As a Use Case (UC) in this project, Mobitrust will be used to test the framework to be developed.

To better understand the context of network security, research has been conducted and the attained knowledge is presented in section 4.5. Service-mesh architecture related concepts and technologies are presented in sub-section 4.5.1. Traditionally network solutions that are commonly applied in network security systems are overviewed in sub-section 4.5, while the most common attacks are listed in sub-section 4.5.2 and further insights on traditional countermeasures are described at 4.5.3. Sub-section 4.5.4 provides an overview of the role that AI may have into network security systems.

The purpose of this work is highly related to the security task within the 5g-EPICENTRE project and an introduction to it is provided on section 4.6, followed by the description of the existent HSPF developed to attend the needs of such task on section 4.7.

4.1 Project Description

5G ExPerimentation Infrastructure hosting Cloud-nativE Netapps for public proTection and disaster RElief (5G-EPICENTRE), is a European project funded by the European Union [3], being its *logo* presented in Figure 4.1. This work will meet the needs of a specific task of this project, that OneSource is responsible for, Task 2.6: Attack surface decrease and network edge access control, namely due to the need of the development of an AI component to integrate the already idealised framework. This framework is presented on section 4.7, while the overview of

the mentioned task is conducted under 4.6.

Figure 4.1: 5G-EPICENTRE Logo [3]

5G-EPICENTRE aims to a creation of a NetApp, where it should be easy to manage and deploy existent 5G solutions. This platform will accommodate and provide open access to the 5G network's resources, where PPDR agencies may find solutions ready to be deployed that will take advantage of the 5G enhancements on communications when compared to previous technologies.

The provision of a NetApp application that enables the deployment of already conceived 5G solutions, might be a game-changer for Small Medium Enterprises (SMEs) that are looking to enter 5G markets, but, are receptive due to the high costs associated with the technologies needed to deploy 5G solutions. Being able to deploy custom PPDR solutions and evaluate their feasibility without having to previously invest into the needed infrastructures to deploy them, might be the boost many companies need to come up with really useful solutions.

Taking into account the novelty of 5G technologies, associated with the cloud-based technologies, applications have been redesigned for this new reality, where the distribution of the internal components, the technologies used, the way the components communicate, etc, have all been adapted (namely, to micro-services). In such a new reality, security aspects must be carefully handled, which leads to a need for security framework updates, redesigns or even new implementations. T2.6 of this project is in charge of the cross-layer security aspects of the NetApp to be created, as well as of the several UCs external and internal communications related aspects. Section 4.6 provides an overview of this task objectives.

4.2 Project Consortium

The project consortium counts with 17 participants with different backgrounds: education facilities (e.g., universities), SMEs actively involved in R&D projects, telco companies and also major players, which embrace several areas, such as the project coordinator company, AIRBUS. Figure 4.2 presents the consortium of the project.

Figure 4.2: 5G-EPICENTRE Consortium [3]

4.3 Main Objectives

As a way of fomenting an overview of the project, next are presented its objectives:

The following objectives have been defined:

- To build an end-to-end 5G experimentation platform specifically tailored to the needs of the public safety and emergency response market players.
- To pilot 5G systems in PPDR-based trials, successfully demonstrating 5G-EPICENTRE on boarded apps as a crucial communications accompaniment to public safety mission-critical communications technologies.
- To cultivate a '5G Experiments as a Service' model, which will enable developers and SMEs to experiment with PPDR applications in parameterized, easily repeatable, and shareable environments.
- To facilitate automation, continuous deployment and multi-access edge computing supported by containerized network functions, so as to reduce service creation time and time-to-market for 5G solutions.
- To leverage Artificial Intelligence for achieving cognitive experiment coordination and lifecycle management, including dynamic 5G slicing, application awareness and insightful ML-driven analytics.
- To implement impact-driven dissemination, standardisation and exploitation.

4.4 Mobitrust Situational Awareness Platform

Mobitrust is used as a UC in 5G-EPICENTRE as a PPDR application. Mobitrust components are starting the integration with 5G technologies in order to deliver an improved field awareness through reliable communications, GPS positioning, high-quality real-time video, among others. Due to the technologies currently being used on its *deployments*, Mobitrust presents itself as the perfect candidate for the application of the security framework being developed, therefore, allowing a practical environment where this framework will be tested and integrated.

The Mobitrust platform (represented into figure 4.3) is a situational awareness platform thought to be used under PPDR use cases. Subject to continuous development, by a specialised OneSource team, it aims to take advantage of the recent technological developments to improve the privacy and security of communication devices, improve the quality of audio and video transmissions and increase awareness over field operations.

Figure 4.3: Mobitrust - Enhancing the operation of field deployed teams [55]

Mobitrust application has a vast field of possible users, such as police forces, fire departments, civil protection, armed forces, workers and emergency medical services, among others. Using technologies such as sensors (biological, environmental and geographic), mobile devices, real-time audio and video transmissions, LTE and 5G (successful tests have already been conducted) communications, among others, Mobitrust offers the following functionalities:

- Integration with 4G and 5G public safety communications
- Data correlation and personalised notifications
- Integration with Commercial-Off-The-Shelf (COTS) devices
- Integration with Mobile Device Management
- Advanced statistics
- A secure mobile platform
- Automatic actions in response to a set of defined events
- Automatic events in case of an anomalous sensor reading

All the video, audio and sensors transmissions are sent to a mobile control centre (if present) and to a central control centre (CCC). From these control centres and taking advantage of the entire panoply of available information, the operators are able to perform informed decisions, which is only possible due to the high level of awareness of the field operations.

Architecture

Thought to be used anywhere, at any time, Mobitrust platform has been developed and adapted to a Cloud-Computing environment, where it can be set up close to an occurrence under the PPDR paradigm. This proximity is one of the characteristics that PPDR applications should allow in the near future. Edge-Computing is the concept behind this idea, it means, among other aspects, taking the solutions next to the occurrence scenario. Such proximity will allow achieving a higher awareness of the field operations, by increasing the quality of audio/video streams received in the central controllers, as well, by reducing the latency of those and other streams (e.g., sensors data). When both these factors are combined, the awareness of the field operations increases and quick actions might be triggered in answer to the alerts generated by the application.

In order to achieve an even higher security level within Mobitrust platform, and considering its migration to a Cloud paradigm, it's now necessary to develop a set of security mechanisms to deal with the threats associated to this fairly new paradigm. One of them is the addition of a component to detect traffic anomalies being originated either within or outside of the platform.

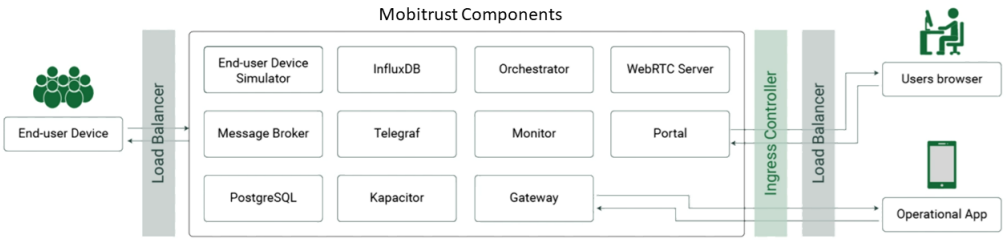


Figure 4.4: Mobitrust K8s Deployment Architecture

Figure 4.4 represents the current Mobitrust architecture when deployed using K8s functionalities. The mentioned security component will be added in parallel with the existent components. Chapter 5 addresses the proposed approach, where this integration is further detailed.

A succinct description of the within Mobitrust micro-services are presented next.

- **End-user Device Simulator**: This component is used for integration tests and debug purposes. It pretends to simulate the data streams usually established between a real end-user device and the Mobitrust several components.

- **InfluxDB**: DB used to store the information collected from the multiple sensors present in the Mobitrust bodykits.
- **Orchestrator**: The orchestrator is responsible for the management of the control data. It deals with the authentication and authorization of users. Moreover, it is responsible for the setup of the end-user device components, including its drivers and the establishment of the data channels both for sensors and communication devices (cameras and microphones).
- **WebRTC Server**: The WebRTC server is the component that deals with audio and video transmission in real-time from the field towards the Command Control Center (CCC).
- **Message Broker**: This component represents the communication backhaul of the system. It follows a publish/subscribe model. The message broker is responsible for all the communication among components.
- **Telegraf**: A plugin-driven server agent for collecting and reporting metrics. Through connecting to the Message Broker, it collects data from the system, mainly sensor data from the end-user devices.
- **Monitor**: This micro-service is responsible for watching and reporting on the state of the end-user devices.
- **Portal**: It is the frontend of the platform, the actual CCC application to be used by human operators. Provides a way to obtain situational awareness by visualising all the data collected by the platform.
- **PostgreSQL**: The relational database that stores the information regarding users, end-user devices, WebRTC mount points and their associations, as well as the access control policies.
- **Kapacitor**: Is a native data processing engine. It can process both streams, as batch data from InfluxDB. With Kapacitor it is possible to plug in custom logic or user-defined functions to process alerts with dynamic thresholds and perform specific actions based on these alerts.
- **Gateway**: The operational controller is responsible for the services provided by the Command and Control Centre. It has all the backend operations that enable the visualisation of the data collected by the platform, as well as the processing of requests of the human operators.

Regarding the remaining components, a brief description is provided hereafter.

- **End-user Devices**: The wearable encompasses all the equipment for 5G communications, sensors data collection, multimedia capture and data preprocessing.
- **Users browser**: Designated by CCC, it presents the front-end of the platform, offering the intended awareness over field operations, through the

presentation of geo-localisation of every field operator, plus specific per-operator data and streams, such as real-time video, real-time communications and sensors data (temperature, heartbeat, etc)

- **Operational App**: Corresponds to the mobile version of CCC. It is used by first responders and is able to present data collected by any wearable that the current user has clearance to access.

4.5 Security Context

Considering the theme of this work ("using ML for anomaly detection over traffic present in service mesh architectures") it is crucial to seek for a good understanding of the major aspects related to this subject and as such, the attained knowledge is portrayed into this section. Starting with the context of service-mesh and the tools commonly used to manage micro-services, in sub-section 4.5.1, followed by the identification of the most common attacks (in sub-section 4.5.2) and of the traditional measures to mitigate such attacks (in sub-section 4.5.3). Finally, it is overviewed the role of AI into network security systems, namely for traffic anomaly detection.

4.5.1 Micro-services Orchestration

To better understand the concept of service-mesh architecture, an investigation was conducted. According to [56], "a Service Mesh is a dedicated infrastructure layer with a set of deployed infrastructure functions that facilitate service-to-service communication through service discovery, routing and internal load balancing, traffic configuration, encryption, authentication, authorization, metrics, and monitoring.". The most common technologies used when implementing this type of service architecture are Kubernetes (K8s) and Istio, therefore, in order to obtain some insights on how this can be done, a special emphasis is given in this section.

Considering the objectives of this work, the policy enforcement upon a set of alerts being generated through the ML algorithm, reflects the intended desired. Thus, the gathering of information on this type of mechanism helps to understand *how* and *what* can be done, after the identification of a threat.

Kubernetes

K8s [57] is a container orchestrator responsible for configuring, maintaining, and automating clusters. By taking advantage of application containerization, K8s enables the distribution of applications in scalable microservices. Some of the most important K8s features are the following ones:

- **Node**: Node (or worker node) is a machine (physical or virtual) that integrates a cluster and provides the execution environment for containers.

- **Reconciliation loop**: The process seeks to reconcile the current state with the desired state of an object. Applied to K8s, the current state of the cluster is compared to the desired state of each resource and the necessary adjustments are made. It is the fundamental principle behind the automation and constant operability of applications in K8s.
- **Pod**: The pod is the fundamental unit of computation in K8s. It specifies a logical aggregation of one or more containers that are executed simultaneously. Containers within a pod run on the same node, a feature that enables out of the box communication between them. Each Pod receives a non-permanent IP, which excludes the possibility of using static IP addresses while establishing communications.
- **Deployment**: Deployment is a resource for the declarative specification of a Pod, its behaviour, and its life cycle. Essentially, deployments are the base resources for installing applications.
- **Sidecar Proxy**: It's present alongside a pod. Its purpose is to proxy or route traffic from and to the instance, it runs allocated to. Each sidecar is the tool responsible to communicate with other sidecar proxies. Each group of sidecar proxies is managed by the Service.
- **Service**: Service is an abstraction of a set of one or more Pods into a service. More specifically, it is a resource for configuring the communication aspects between pods in a cluster. The service allows the definition of a permanent IP that routes traffic directly to the respective pods, similar to a load balancer. A service can be of type: ClusterIP (exists only in the context of the cluster), NodePort (opens a port on the node) and LoadBalancer (uses an external load balancer).
- **Manifest**: Manifests (of resources) are configuration files that describe the desired state of a given resource. They are usually written in YAML [58] files.

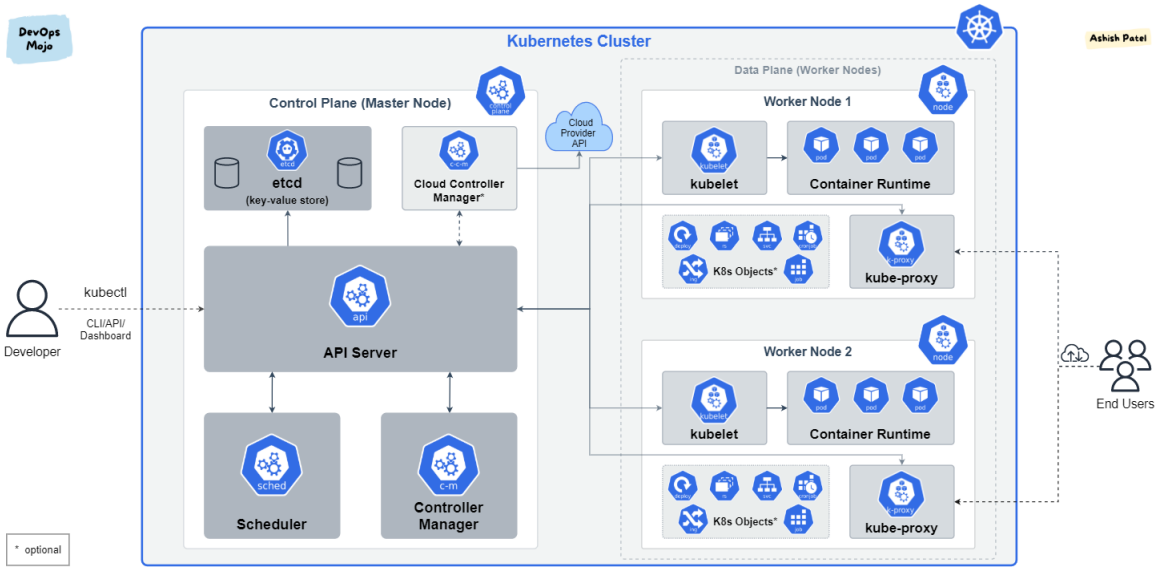


Figure 4.5: Kubernetes Reference Architecture [59]

Figure 4.5 presents the main components that compose the Kubernetes architecture. The major two concepts in a Kubernetes architecture are the Control Plane and the concept of Node(s). The Control Plane, also known as the master or head node it's responsible to manage the worker nodes and the pods in the cluster. This master node usually receives input commands from the user (developer) through an API. Node(s) correspond to worker or computing nodes, usually allocated in virtual machines that include the services needed to run containerised applications. At least one Node needs to exist in a Kubernetes cluster, but, in the most common usages, many nodes are found running simultaneously. Each work node(s) hosts one or several Pods, that are scheduled and orchestrated to run on Nodes by the master node.

The internal components of a Master Node and a minor description of their functions are the following:

- **API Server**: Exposes the Kubernetes API and manages all the clusters.
- **etcd**: key-value stateful and persistent storage;
- **Scheduler**: Schedules pods to worker nodes;
- **Controller Manager**: Manages the state of objects, taking special attention to the current and the desired state of each;
- **Cloud Controller**: Similar to Controller Manager, but, dedicated to control clusters running in a cloud environment;

The internal components of Node and a minor description of their functions are the following:

- **kubelet**: Responsible for Pod Lifecycle and respective activities.
- **kube-proxy**: Responsible for network details (e.g. IP, network rules, etc)
- **Container runtime**: Is the software responsible for running containers (in Pods)

Service Mesh

A *service mesh* is a programmable infrastructure layer thought to deal with a high volume of network-based communication-related processes among multiple several application infrastructure services using APIs. Such layer assures that communication between containerised application infrastructure services is fast, reliable and secure. The mesh principle is responsible for service discovery, load balancing, encryption, observability, authorisation, authentication, betwixt other.

In order to deploy a service mesh architecture and have observability over its internal operations, several tools can be used. One of the most common ones is Istio. Istio [60] is an open-source service mesh, which aims to bring security, management and monitoring of services to a more transparent and centralised level.

It's distinguished by its properties of load balancing, secure service-to-service communication (using TLS encryption) and access control, through a pluggable policy layer.

It runs over Kubernetes and allows a very large set of operations, such as: adding applications to a cluster; extending the mesh to other clusters; connecting VMs or other machines located outside of Kubernetes.

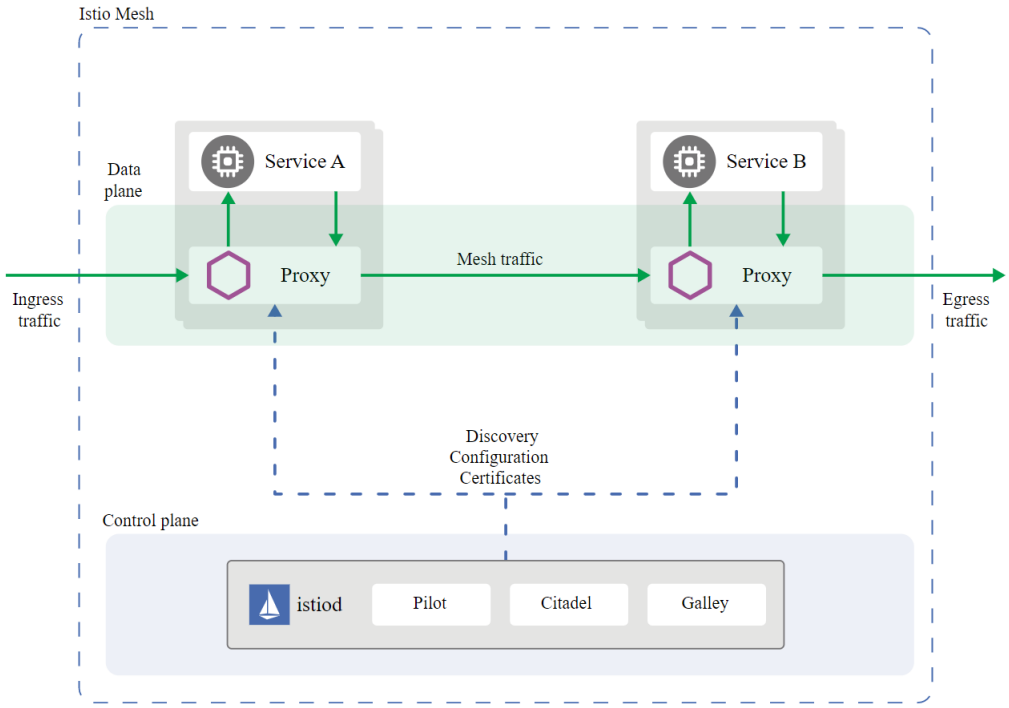


Figure 4.6: Istio Deployment Architecture [61]

Figure 4.6 presents the usual Istio Deployment Architecture, where it is possible to point out the Control plane and the Data plane. The data plane represents the communication between services. Each time a communication exists, it is captured by a proxy, specially designed to intercept all the network traffic. Such communications are analysed and if needed actions are taken. The control plane is responsible for handling service discovery, configuration and certificate management. The *Istiod* is the mechanism that translates high-level routing rules and propagates them towards the sidecars at running time. It also has a strong built-in security library that enables the application of proper authentication mechanisms used within service-to-service and end-user communications.

Policy Enforcement

Under a service-mesh architecture, a set of policy rules can be used to control the communications among Pods and also the communications between Pods and external cluster origins. Most of the policies can be defined prior to the deployment, but also during run-time. This, allow dynamic services to automatically create, delete and apply policies on the fly, which is something of high impor-

tance for security systems considering that these must continuously monitor the state of the network and act (possibly create a new policy and/or apply an existent one) when an anomalous situation is detected.

Open Policy Agent (OPA) [62] is used for handling policies in Service Mesh architectures. It is an open-source, general-purpose policy engine that unifies the implementation of policy enforcement procedures across the IT environments, such as the ones involving Cloud-native applications.

OPA provides a high-level declarative language to specify policies as code, and simple APIs to offload policy decision-making from software. OPA can enforce policies in microservices, Kubernetes, CI/CD pipelines, API gateways, and more. OPA can also be used to control authorization, admission, and other policies in Cloud-native environments, with a focus on Kubernetes.

OPA is a lightweight general-purpose policy engine that can be co-located to the existing services. OPA can be integrated as a sidecar, host-level daemon, or library.

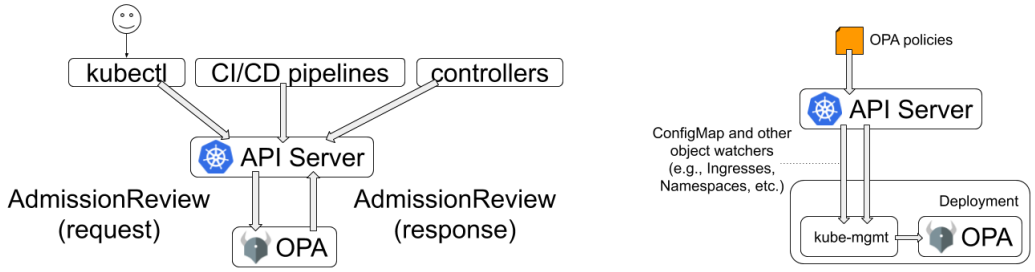
OPA is a general-purpose policy engine that decouples policy decision-making from policy enforcement. Its high-level declarative language provides intuitive ways of specifying policies. It can be used to enforce policies on several environments, namely on microservices and Kubernetes. Whereas Istio policies are limited to networks, OPA allows a more comprehensive strategy to implement distinct policies and take more control over deployments and containers.

Policy decisions to be made by the different applications are supported by queries to OPA, where the supplied input is a set of structured data (e.g., JSON). Declarative policies in OPA's policy are defined in Rego language. OPA generates policy decisions by evaluating the query input against policies and data. OPA and Rego can be used to describe a large number of different policies and they are agnostic to the domain. Some examples of possibilities are:

- Define which resources the users can access;
- Define the traffic that is allowed into egress subnets;
- Define how the workloads are executed within the Clusters;
- Define the origin from where binary registries can be downloaded;
- Define the OS capabilities that the container can use;
- Define time periods where access to the system are allowed;

Policy decisions are not limited to a simple yes/no or allow/deny answers. Like query inputs, policies can generate arbitrary structured data as output. An example of a security policy to be implemented can dictate that the servers should be reachable from the Internet and must not expose the insecure 'http' protocol. A second security policy may involve the servers that are not allowed to expose the 'telnet' protocol. The policy needs to be enforced when servers, networks, and

ports are provisioned, and the compliance team needs to periodically audit the system to find out if any servers are violating the policy.



(a) Kubernetes API Server and OPA

(b) Load Policies Mechanism

Figure 4.7: OPA Reference Architecture [62]

Figure 4.7 presents the reference architecture for two most common situations in OPA. Figure 4.7.a) represents the set of situations when it's being done some operation over the Pods in the system, which leads Kubernetes API Server to query OPA every time a create, update or delete operation is being done. Figure 4.7.b) illustrates the load policies mechanism, where the policies may be uploaded to OPA dynamically using the *kube-mgmt* sidecar container, via ConfigMap objects.

4.5.2 Attacks

When an attacker lunches an attack towards a network, it usually falls into one of the following categories: *active* or *passive* [63]. The first category encompasses situations where the attacker executes commands to disturb the network's normal operation, while the second category, covers situations where there isn't any intent to interfere with the behaviour of the network, instead, the attack is focused on the interception of network data traffic.

Some of the most common *active* types of attacks are [63]: (i) Spoofing; (ii) Modification; (iii) Wormhole; (iv) Fabrication; (v) DoS; (vi) Sinkhole; (vii) Sybil. Regarding passive attacks, the most frequently seen are: (i) traffic analysis; (ii) eavesdropping.

A brief description of the *active* type of attacks mentioned before is presented next:

- **Spoofing**: When someone or something (e.g., a network node) miss-present his identity, aiming to get the system trust so it can have access to sensitive information, therefore, compromising the Confidentiality, Integrity and Availability (CIA) of the data [64].
- **Modification**: When data is tampered with. This type of attack might primarily be considered an integrity attack but could also represent an availability attack. If illegal access is successful and the attacker ends up tempering the content of a file, he just affected the integrity of the data. In some

situations, if the file in question manages the functioning of a network service, the attack may result in the unavailability of that service [65].

- **Wormhole**: Here, the attacker receives packets at one point in the network tunnels them to another malicious node in the network and then replays them into the network from that point ahead. This type of attack has a special negative effect over route discovery protocols since it's possible for attackers to control the routes shared in the network, and mislead a node to communicate through the malicious nodes [66].

- **Fabrication**: Fabrication attacks involve generating data, processes, communications, or other similar activities occurring within a system. The malicious population of a database is a fabrication attack. If enough additional processes, (e.g. network traffic, e-mail, Web traffic), or anything else that consumes resources are generated, the availability of the service that handles such traffic may become unavailable to legitimate users of the system [65].

- **DDoS**: DDoS attacks usually consist of, in a certain period, a big amount of specific data packets being sent directly or via the springboard to the target network, which consumes the network bandwidth and system resources greatly and causes the blocking or even paralysis of the target network. It is usually delivered in a distributed and collaborative way, directly or indirectly attacking the target system or network resources. The most common type of DDoS attacks englobes situations where not only one attack mode is conducted at a time, rather, multiple attack modes can be executed at the same time or even multiple targets can be attacked simultaneously [67].

- **Sinkhole**: In a sinkhole attack, a malicious node tries to obtain the data to it from all neighbouring nodes. So, practically, the node eavesdrops on all the data that is being communicated between its neighbouring nodes. Selective modification, forwarding or dropping of data can be done by using this attack, especially into node base networks, where each node usually depends on others to exchange data among the different elements of the network [68].

- **Sybil**: In Sybil attacks, attackers try to manipulate fake identities or abuse pseudo-identities to compromise the well-functioning of the systems. Sybil accounts not only spread spam and advertisements but also may disseminate malware and fishing websites with the purpose of stilling legit users' sensitive information. Since most Sybil attackers (accounts) behave similarly to normal users, the detection of this type of attack poses some challenges [69].

Similar for *passive* attacks, the following list presents a brief description on the mentioned attacks [63]:

- **Traffic Analysis**: Traffic analysis attacks try to deduce the context information of nodes by analysing the traffic pattern from eavesdropping on

wireless communications. Specifically, adversaries might collect information about the network structure, and deduce the location of strategic nodes through observing the traffic volume and pattern [70].

- **EavesDropping**: Eavesdropping consists of the theft of information being shared in a network. Eavesdropping is a deceptively mild term. Usually, the attackers are after sensitive financial and business information that can be sold for criminal purposes [71].

4.5.3 Traditional Countermeasures

According to [72], the baseline for a network company security system is composed of an authentication process that enforces strong password policy, antimalware on all computers, email system with SPAM filter enabled, stateful or proxy perimeter firewall, reasonable network separation among the different company areas and teams, provide some training awareness to the company workers and apply hard drive encryption on mobile hosts.

For external communications, the use of Virtual Private Networks (VPNs), establishing connections to highly protected and specific *gateways* machines, plus the application of a set of rules defined by *IPTABLES* are other usually seen techniques to guarantee security in a network rule based system.

A philosophy of “we may not be able to anticipate all attacker actions, but, we can diminish their space manoeuvre” is usually followed and is a good practice since it allows to identify the most vulnerable services and create mechanisms to secure them, while at the same time, to eventually create conditions to direct the attackers to non-critical services or isolated networks without any critical data.

In order to detect traffic anomalies, traditional network security systems commonly use Signature-based Network Intrusion Detection System (SNIDS), being the most common Snort and Bro [73].

An Intrusion Detection System (IDS) is a mechanism that monitors networks or systems, looking for malicious activity or policy breaches. There are several types of IDS, being the most common: signature-based, specification-based and anomaly-based.

Signature-based systems are the most common in the companies network security systems and these systems are able to detect intrusions by comparing known attacks signatures with the content of network traffic.

Specification-based systems start by creating "normal-behaviour" profiles, built taking into account the functionalities and the enforced security policies. After this, eventual traffic that falls out of this profile is labelled as malicious.

Anomaly-based detection works by recognising malicious behaviour. There are two major ways how this can be done, the first, is similar to specification-based, in a sense that malicious activity is identified through the comparison with "normal-behaviour" profiles (that are built considering different network traffic insights).

The second, is through the use of AI techniques, which is the main focus of this work and is further detailed on sections 2.3, 3.1 and 4.5.4.

SNIDS usually apply a Deep Packet Inspection (DPI) method, which enables these methods to achieve a high detection rate, but, usually at the cost of low performances. Despite that, the performance of SNIDS is highly influenced by the number of rules to be verified, which raises the need for the rule-set to be adapted to the specific-domain use case.

Currently, there are still some advances being suggested for SNIDS systems, such as, in [74] and [75].

In the first, a high-speed signature-based flow intrusion detection system is presented, aiming to overcome the difficulty inherent to the processing of big amounts of data. This is achieved by the use of *IPFIX Flows* and through the use of the open source network monitoring toolkit, Vermont [76]. This toolkit contains several components developed using assembler, therefore, using direct CPU registers, which contributes to the lowest overall system detection time. Each rule usually has several patterns to compare with and the use of an optimise policy check mechanism (if for a pattern, one check fails, the validation of that rule is aborted) also helps to achieve a lower processing time.

In the second, a new signature-based IDS is presented aiming to improve the detection of DoS and routing attacks. The authors suggest a system with not only centralised IDS systems, but, also, distributed ones, where the proposed IDS accumulates malicious patterns in a specific detection module, responsible to make the bridge between an internal network and the Internet. A set of experiences is conducted, involving Cooja simulator [77] and special attention is given to power consumption aspects, upon a DoS attack.

4.5.4 AI in Network Security

According to [78], near 127 new devices are connected to the internet every second, and it is foreseen that the worldwide number of connected devices will be higher than 27 billion by 2025.

The world is evolving to a state where the simplest and smaller kitchen equipment will be connected to the internet, likely through a control home central. In such a paradigm, the amount of data being transmitted over the different networks will be reflected into a never seen volume of traffic. In order to deal with these high volumes of data and also to assure the correct functioning of the systems that will manage all of these smart devices, evolving to a cloud paradigm will be crucial so it's possible to take advantage of the most promising cloud features, namely the flexibility, mobility, disaster recovery, loss prevention, costs saving and competitive edge, among others [79], to assure the correct network behaviour.

This new reality brings several security concerns, on the device level, considering that the majority of IoT devices are not designed to handle cyberattacks and privacy threats, and, on the network level, since the traditional networks are not

prepared to deal with such high volume of data, or even to apply the needed security policies to all the traffic aiming for guaranteeing the CIA Security Triad of the data CIA [80].

The integration of cloud systems into an already established organisation structure considerably increases the attack surface [81]. This growth adds more levels of difficulty to the several already existent problems: how to manage the volume data, the speed, the complexity of the data, etc. The wider the attack surface, the bigger the amount of data to analyse. That's where the AI components may help, by automating traditional security operation tasks, namely, by providing a first analysis over the spread network data, allowing humans to only focus on higher-level tasks.

The lunch of spontaneous and coordinated attacks has been increasing over the past years. While some organisations are still using manpower to collect and analyse these attacks, the attackers are taking advantage of not yet protected security vulnerabilities. The time frame needed for human-based security systems to detect an ongoing threat may be considerably bigger or not even happen (the threat/attack might not even be detected) which may represent a considerable time frame in which attackers may exploit system vulnerabilities and gain illegal access to data. Knowing this, companies are exploring the use of AI solutions to mitigate this type of problem [82].

There are several ways on how AI is being added to security systems [83], namely by the: (i) Use ML to detect AI-based threats; (ii) Use AI to improve human decision; (iii) Use AI as a Tool and Guideline.

ML techniques are being used to detect traffic anomalies, namely threats, through the analysis and identification of knowledge obtained from past anomalies. Machine learning can help a computer to find anomalies and predict threats more accurately than the average human. Usual technology relies on stale data that cannot provide new scenarios and methods, which is something that *AI* can do, thus, it presents as a good solution for the problem of how to detect anomalies in a Big Data paradigm.

The use of AI into security systems doesn't necessarily mean that human action will no longer be required. In fact, it's quite the opposite, human action should continue to be required as a second layer of verification, while AI process all the data and only escalates to human analysis suspect traffic.

The use of AI as a Tool and Guideline is another useful utilisation of AI mechanisms, that may save a lot of time for humans. Using its capacities for learning, it can map traffic patterns to existing or even new policies. This way, it may help strengthen the network against new threats and attacks, that are continuously being developed by attackers.

Some of the benefits of including AI into a security system are [81]:

- **Prediction and remediation improvement**: Using AI capabilities to detect known and unknown attacks and enhance the response actions.
- **Quick threat detection**: AI is able to detect traffic anomalies much faster

than humans, therefore, it may rapidly quantify the risks and accelerate the process of security policies enforcement.

- **Efficient human resources management**: The use of AI enables cyber-security professionals to focus on higher-level tasks, instead of more time-consuming and lower-level ones.

Some of the most common applications of AI into a network security systems are [84]:

- **Password Protection and Authentication**: A single password is nowadays the only barrier between a valid or invalid session. Two-factor authentication mechanisms may be used, however, in most cases, this can also be easily bypassed. An attempt to increase the security level of authentication mechanisms is being conducted by several companies, through the implementation of AI models that aim to create a model able to recognise someone's face. In some cases, this identification is being done using mathematical expressions to calculate unique relations in a persons face, like the distance between eyes or the distance from ear to ear.

- **Phishing Detection and Prevention Control**: The amount of phishing attacks has been continuously increasing, leaving system administrators with full hands while identifying the origin of these emails. AI models are being applied to detect this type of email, through the analysis of network traffic, allowing for a quicker identification of the threats, which translates into a faster appliance of security policy measures to neutralise the threat.

- **Vulnerability Management**: Many human-based security systems follow a reactive approach, that is, only after some event or some failure, an investigation process is triggered. Using AI it is possible to turn this process proactive, therefore, signalling possible points of failure, before those become a target by attackers.

- **Network Security**: Security systems without AI demand an initial configuration of security policies and then to be kept updated regularly. With the use of *AI*, this task can be far simplified. It will repeatedly learn traffic patterns and suggest new security policies, thus, facilitating human activities in this area, allowing them to focus on the most imperative tasks.

- **Behavioural Analytic**: This is a type of task that without AI is nearly impossible to do. Here, AI models are used to analyse usual user activities. Whenever a user activity falls out of the usual behaviour, the conducted activities may be reported as suspicious. This type of mechanism may be used for several purposes, namely for credit card fraud detection (based on gps position, transaction value,...), invalid user login attempts, etc

4.6 Attack surface decrease and network edge access control

According to the project documentation, Task 2.6 *Attack surface decrease and network edge access control* aims to define secure network policies to deal with the increased attack surface resulting from the migration towards edge Virtual Network Function (VNF) containerisation. A special emphasis is given to Access Control, considering that 5G-EPICENTRE architecture should be flexible enough to enable the network edge to have a certain level of autonomy in terms of decisions to grant access or not. The specifications for the authentication processes shall be put into place, namely, to enable the Policy Enforcement Points deployed at the edge to perform a decision request to an always available Policy Decision Point. The definition of resource guarantees while deploying co-located instances of containerised VNFs is another objective of this task.

It's important to mention that this task is led by OneSource, which has significant experience in the security field over networks, obtained through the continuous engagement into R&D projects.

Under this task, part of the work developed by the company, aiming to secure the several layers of the project architecture, including to deal with cross-layer aspects, a proposal for the security framework as been reflected into [85]. Figure 4.8 illustrates the mentioned proposal.

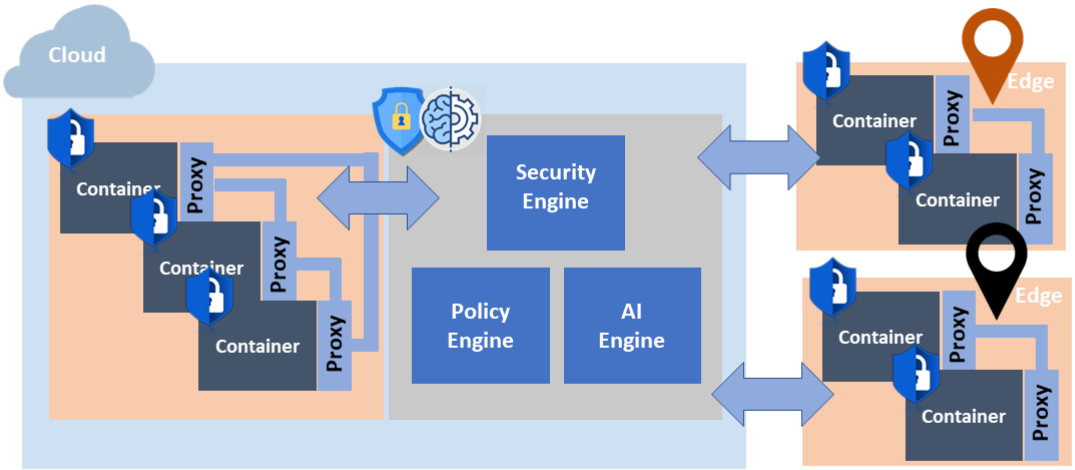


Figure 4.8: 5G-EPICENTRE Security Framework Proposal [85]

The early design of the security framework encompasses three main components: the policy engine; the security engine and an AI engine. The policy engine con-

tains the configuration of the policies at the network and container levels. The security engine comprises the protection to the underlying host OS, where the containers run, by providing access control and authentication mechanisms, network traffic encryption and container isolation methods. The AI engine corresponds to the intended next step, to assist security and policy enforcement.

The connection between this project and specifically this task with the current work, is the need for the development of an AI solution to integrate the mentioned framework to detect traffic anomalies, therefore, enabling the framework to take advantage of the use of an AI solution towards the prompt and accurate signalling of possible threats.

In the ambit of the development of the mentioned security framework, several steps have been taken, namely, the definition of the use cases, requirements and reference architecture of the HSPF. Must be noted that due to the presence of several Master of Science (degree) (MsC) students from DEI currently engaged in internship activities at OneSource, the listed achievements are the result of inter cooperation and discussion among the several students.

4.7 Holistic Security and Privacy Framework (HSPF)

One of the major objectives of this work is to prepare a set of algorithms able to integrate the detection module of the HSPF. Among other purposes, this framework will be used to provide security to the several Mobitrust micro-services, previously described in section 4.4.

4.7.1 Reference Architecture

The current reference architecture for the HSPF developed by OneSource is illustrated in image 4.9. It must be noted that this framework corresponds to the blue blocks identified in Figure 4.8, when the *Policy Engine*, *Security Engine* and *AI Engine* are stated.

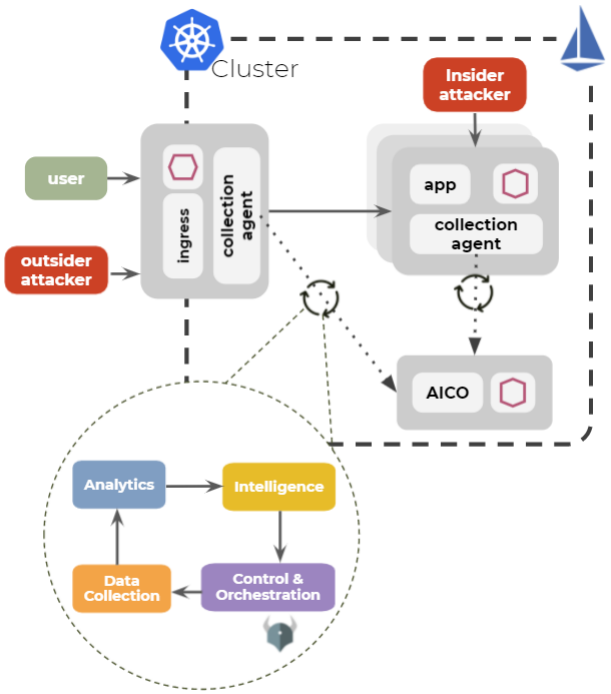


Figure 4.9: Reference Architecture of the HSPF

The reference architecture has two major different components: the set of *collection agents* and the AICO component. Each *collection agent* is responsible to capture traffic data and send it to the AICO component. The AICO is composed by four components: *Data Collection*, *Analytics*, *Intelligence* and the *Control & Orchestration* block's. AICO is envisioned to work in a closed loop, continually producing alerts, every time malicious traffic is detected from the data continuously collected from the collection agents, pre-processed and respectively analysed by the *Intelligence* component.

The algorithm(s) to be developed will integrate the *Intelligence* block, since this block is the one responsible to analyse a set of features extracted from traffic communications and for generating warnings that are later interpreted by the *Control & Orchestration* block, which will later decide which policies to apply, or, alternatively, to drop them. The Data Collection is the mechanism in charge to receive the communications traffic from the several *collection agents*. The *Analytics* block is responsible to extract from the correspondent communication packets the set of relative features and applying pre-processing techniques to the collected data. The processed data is then consumed by the *Intelligence* block.

For deployment purposes, when deploying this framework it is usually considered a Kubernetes environment with Istio's implementation of the Service Mesh. The usage of Kubernetes contributes to the easy integration of this framework, as it allows for the injection of *collection agents* at run-time.

This architecture encompasses two different types of attacker's: outsider and insider. The first aims to represent attacks that are originated outside of the platform (e.g. dictionary attacks over the login form on the Mobitrust web application). The second aims to represent situations where the attacker(s) would take

over a Mobitrust platform component. The use-cases and attacks defined for evaluating this performance are, respectively, described in the sub-sections 4.7.2 and 4.7.4.

4.7.2 Use Cases

The definition of use cases is one of the steps that the HSPF should support. This is a way of assuring that this framework will be prepared to handle, at least, the set of use cases specified. As such, the following table 4.1 represents the use cases that will be used to test the AI algorithm.

Table 4.1: List of use cases

ID	Use Case Information
1	**Name**: Login_1 **Description**: The attacker performs a brute force attack to the Portal log-in form posing as a legitimate user **Attack Type**: Brute Force **Origin**: External
2	**Name**: Login_2 **Description**: Attacker's attempt to inject SQL queries through the fields of the Portal log-in form **Attack Type**: SQL Injection **Origin**: External
3	**Name**: DoS Portal **Description**: DoS attempt against to the Portal, through the use of techniques such as SYN attack and Ping of Death **Attack Type**: DoS **Origin**: External
4	**Name**: DoS MQTT Message Broker **Description**: DoS attempt against to the Message Broker exploring the vulnerabilities detailed at CVE-2021-33175 [86] **Attack Type**: DoS **Origin**: External
5	**Name**: PostgresSQL **Description**: Assuming credential theft, attacker attempts to exploit vulnerabilities detailed at CVE-2021-32027 [87] **Attack Type**: Buffer Overflow **Origin**: Internal
6	**Description**: Exploitation of the vulnerabilities detailed at CVE-2019-20933 [88] against the Influxdb component **Attack Type**: Restriction Bypass **Origin**: Internal

Continues on next page

Table 4.1 – List of use cases (continuance)

ID	Use Case Information
7	**Name**: PortScan_1 **Description**: Assuming knowledge of all external IPs of the application, a port scan is performed using the TCP SYN technique **Attack Type**: PortScan **Origin**: External
8	**Name**: PortScan_2 **Description**: Assuming that the attacker has internal access to the Kubernetes cluster, a port scan is performed using the TCP SYN technique **Attack Type**: PortScan **Origin**: Internal
9	**Name**: Istio **Description**: Exploitation of the vulnerabilities detailed at CVE-2020-10739 [89] against the Istio application **Attack Type**: DoS **Origin**: Internal
10	**Name**: Kubernetes **Description**: Exploitation of the vulnerabilities detailed at CVE-2019-11248 [90] against the Kubernetes platform **Attack Type**: DoS **Origin**: Internal
11	**Name**: WebRTC **Description**: Exploitation of the vulnerabilities detailed at CVE-2022-21667 [91] against the WebRTC internal component **Attack Type**: Buffer Overflow **Origin**: Internal
12	**Name**: WebRTC **Description**: Attacker plays HTTP requests to the Kubernetes API in order to obtain information about the environment **Attack Type**: Enumeration **Origin**: Internal
13	**Name**: Portal XSS **Description**: The attacker attempts to inject malicious scripts through the Portal login form, namely, to illegal update user credentials **Attack Type**: XSS **Origin**: External

4.7.3 Requirements

The gathering of requirements is one of the first steps in the most common software development methodologies. This is usually a process where a set of questions are raised due to the perceptions that each different party involved has over the final product, thus, this process is done to get a concrete picture of what the product to be developed must and must not do.

Considering the problem in hands, the development of a AI component to detect anomalies over network traffic that later will be integrated into a security

framework, the following list of requirements was elaborated (Table 4.2).

Table 4.2: List of requirements

ID	Requirements	Priority
1	A framework must be able to handle network traffic of multiple locations including traffic between multiple microservices on a Cloud-Native environment to be processed in closed-loop	High
2	The framework must have the ability to collect network traffic from multiple locations including external traffic to be processed on the closed loop	High
3	The framework must have the ability to create a set of features and metrics from the captured network traffic to be processed by the AI algorithm (e.g., CIC-IDS2017 dataset features)	High
4	The traffic collection mechanism must create a set of features and metrics from the aggregation of network traffic in intervals of variable and configurable time	Medium
5	The framework must be able to distinguish anomalous traffic with based on an AI algorithm (i.e., binary classification)	High
6	The framework must have the ability to distinguish different types of attacks based on an AI algorithm (i.e., multi-class classification)	Low
7	The framework must have the ability to dynamically define policies based on alerts coming from the AI algorithm	High
8	The framework must have the ability to enforce policies in runtime previously defined	High
9	The framework must allow viewing of security policies	Medium
10	The framework should allow manually adding security policies	Medium
11	The framework must allow removing security policies	Medium
12	The framework must allow disabling security policies	Medium
13	The framework must allow enabling security policies	Medium
14	The framework must allow defining threshold mechanisms depending on the output of the AI algorithm in order to condition the application of policies	Low
15	The framework should allow training the AI algorithm with data collected in real-time in an environment	High
16	The framework must allow training the AI algorithm based on an external dataset previously collected	High
17	The AI algorithm must be able to detect traffic anomalous near real-time	High
18	The framework should allow importing and exporting of AI models	Low
19	The framework must generate notifications/alerts depending on the applied security policies	Medium
20	The framework must provide observability through a log mechanism	Medium
21	The framework must contain two security policy profiles, one for IPs with external source from Mobitrust and the other for IPs with internal source.	High
22	Security policies must have an associated duration pre-defined	Low
23	The framework must allow configuring the duration of policies applied	Low

Continues on next page

Table 4.2 – List of requirements (continuance)

ID	Requirements	Priority
24	The framework must allow associating a policy duration different depending on the origin of the anomaly	Low
25	The framework must allow defining the ranges of internal IPs and external (to allow distinguishing and contextualizing the origin of the anomaly)	Low
26	The framework must have a historical record of events and logs of variable duration in order to be able to meet standards and regulations	High
27	The framework must have an Identity and Access Management (IAM) to control access to various framework components	High
28	The framework must allow the operator to make changes to the AI algorithm decision (e.g. false positives)	High
29	The framework must have an online feedback mechanism that allows to incorporate the correction of incorrect classifications in training the AI algorithm	Low
30	The framework should allow evaluating the performance of the training/testing of the AI algorithm based on a set of indicators (accuracy, precision, recall, F1)	Medium
31	The framework must allow running two or more algorithms AI at the same time to increase detection efficiency	Low
32	In case there are two or more AI algorithms running at the same time, the framework must allow defining the implementation of the decision depending on the decision of the algorithms (ensemble mechanism)	Low
33	Traffic between components must be encrypted	High
34	It must be possible to containerize and package the various platform components in images	High
35	It should be possible to do an automatic deployment of the framework through a set of Kubernetes descriptors	High
36	The framework must allow defining the location of the points collection of traffic depending on the components to be monitor	Low
37	The framework must have a graphical interface that allows the operator view and select traffic collection points	Low
38	The framework must have a graphical interface that allows the operator view general statistics about the process of anomaly detection and policy enforcement (e.g., amount of traffic processed per unit of time, number of anomalies detected, processing and detection times, number of applied policies, etc.)	Low

4.7.4 Attacks

Considering the panoply of existent network traffic attacks, the need to focus on a set of those is essential. Thus, table 4.3 summits the type of attacks that will receive a special attention, namely, when generating attacks data for the custom-

built dataset (abroad in section 4.7.2).

Table 4.3: List of Attacks

Name	Description	Consequences(s)
DoS	Attempt to cause degradation of service quality or even cause its failure, due to the high number of requests	Legit users may lost access to the services
Privilege Escalation	Attempt to manipulate exposed APIs, to tamper with management data (events, configuration files, etc.), or even exploit lateral movements	Monitoring of adulterated values. Impossibility to make appropriate decisions since the real situation is not being reflected.
Brute Force	Attacker's attempt to gain access to a terminated component, through trial-error attempts (e.g. dictionary attacks). On success, it allows the attacker to gain illicit access to a certain component	Monitoring of adulterated values. Impossibility to make appropriate decisions since the real situation is not being reflected.
Vulnerabilities Exploitation	Attempt to exploit application vulnerabilities, versions of used tools. (e.g., in the machine learning/artificial intelligence algorithm	Exploited vulnerabilities compromise system functionalities and security
MitM	Attacker's attempt to intercept communication between two components aiming to tamper with the transmitted data	It results in a lack of trust between the two components and possibly in theft or modification of transmitted information

The attacks previously described may have different origins: DoS attacks might be launched from the outside or from inside the system; Privilege Escalation is usually attempted after the attacker being already within the system; Brute Force is an attempt to overpass some authentication system, which usually happens externally to the system; Vulnerabilities Exploitation is usually tried within the system; MitM attacks may well be executed from the outside or from the inside the system, but, for the purposes of this work, it will only be considered MitM attacks that may possibly happen within the system, which means, among the several distributed components.

4.7.5 Datasets Collection

Two custom datasets were provided during the realization of this internship, regarding to a DoS attack and to a *Port Scan* attack. Despite the fact that this process wasn't conducted by me, for better contextualization it is here represented the generic process of collection of these datasets. Figure 4.10 graphically presents such process.

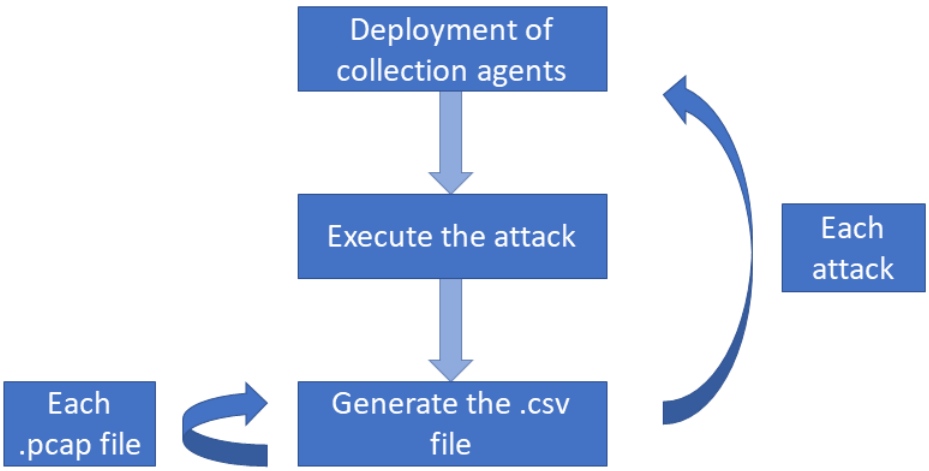


Figure 4.10: Collection of the custom datasets - Overview

The first step consisted in the deployment of the collection agents next to all of the micro-services of the application and it's only purpose was to register the incoming and outgoing traffic for the respective service. Also called as "side-cars", the lifetime of these components was limited to the duration of the simulated attack and its final output was a *pcap* file containing all the network traffic, previously attained using the *tcpdump* [92] tool.

The second step consisted in the lunch of the different attacks.

For the DoS attack, the targeted micro-service was the one containing the MQTT broker used by the application for numerous types of communications. It was used an open source application dominated "MQTT Stresser" [93], with the objective of overloading the service. This application receives as input parameters: the amount of clients, the amount of messages generated by each client and period length of the attack. Table 4.4 represents the values used for the different parameters.

Table 4.4: Parameters while lunching the *DoS* attacks

Number of		Length of the attack (s)
Clients	Messages	
1024	100	60
1024	100	90
1024	500	85
2048	250	60
4096	500	45
5000	1000	60
7000	200	50
10000	250	60
15000	10	60
30600	10	60

The third step consisted on generating the final format of the datasets and to achieve that, the *NFStream* tool [94] has been used. This tool provides a set of helpful features to work with network data structures streams and one of them is the possibility of generating a .csv file from a .pcap one, while managing the set of features present in the final file. During this process, the IP of the attackers was known, thus, the datasets were produced already with the appropriate value of the *LABEL* feature, for each sample.

4.7.6 Current classification life cycle

Aiming for a deeper level of insight over the detection module of the AICO component, an effort was made to better understand what is the path taken by the sample (e.g., flow) since it is collected, classified up to an eventual application of a policy to block the traffic coming from a tagged attacker. Figure 4.11 provides a walkthrough over the different phases.

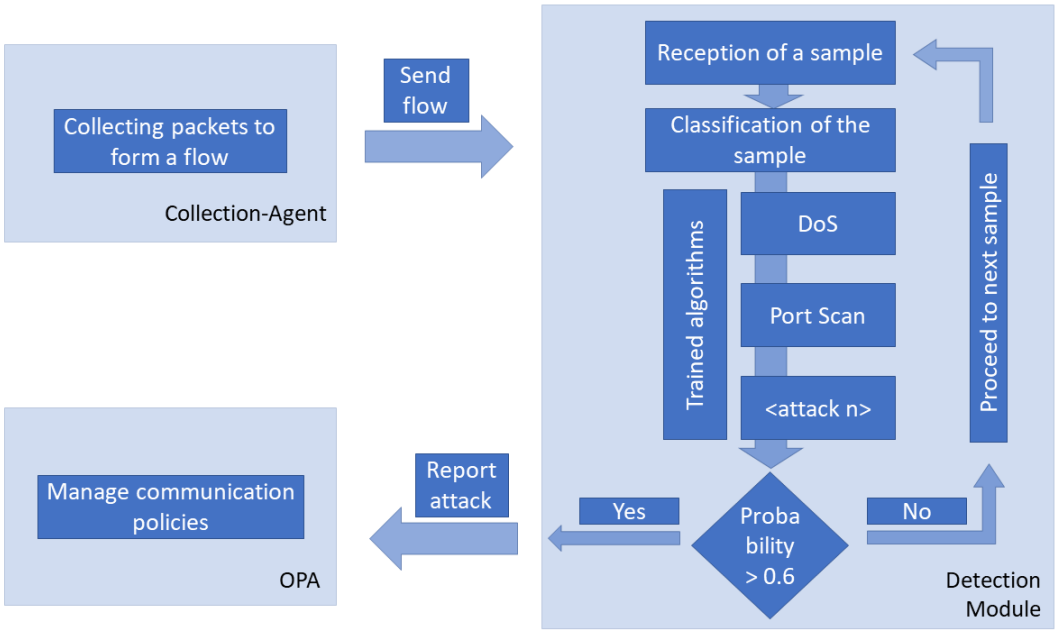


Figure 4.11: AICO classification process - Overview

The cycle begins with the collection of packets up to the formation of a flow on each collection-agent. This is a cycle process, where each collection-agent (placed next to each micro-service) groups several packets and afterwards sends them to the core of the AICO component.

After the reception of the flow, from now on described as "sample", the sample will be submitted for classification, which currently is done as follows: the sample is submitted for classification to all the trained algorithms, for the different attacks. In case at least one of the algorithms classifies the sample as being malignum, with a probability above to 60%, this sample is reported to OPA as an attack.

OPA then manages the communication policies to block the origin of the communication (based on its IP), previously classified as an attack. Recognizing the possibility of false positive classifications, other policies are currently being explored, namely, to introduce some flexibility into the system, that is, not blocking immediately the IP.

An analysis of the existent detection logic and two alternatives to it, are discussed in section 7.5.2.

4.8 Summary

Chapter 4 focus on providing the context where this internship was developed, that is under the 5G-EPICENTRE european project. Besides a proper introduction and project contextualization, this chapter is focused on: (i) understanding the concepts behind the technologies and concepts that allow the management of service-mesh architectures; (ii) having an overview of how network security systems were, before the introduction of AI techniques; (iii) understanding how AI may be useful (and even make the difference) into network security systems; (iv) realize what are the objectives of the project security task; and (v) understand and be aware of the HSPF, developed by OneSource.

It was stated that for micro-services orchestration, the most common technologies used are Kubernetes and Istio, and to enforce security into the communications to and from the internal components of such architecture, OPA presents as an excellent option.

Regarding traditional network-security systems, it was discovered that for anomaly detection, usually are used SNIDS, that take advantage of previously collected knowledge on threats to identify malicious traffic. The list of the more frequently seen attacks is presented, as well, as a brief description of the traditional countermeasures.

The common applications and advantages of using AI in network security systems are reviewed. AI is suggested as a facilitator for traditional heavy human work in an era that security systems need to deal with a never seen widespread network surface plus huge volumes of data that need to be continuously monitored for the identification of malicious traffic.

The attack surface decrease and network edge access control, task 2.6 of 5G-EPICENTRE, is presented and constitutes the bridge between this work and such project. This connection is materialized by the HSPF, which is also properly exposed, through the description of the: reference architecture, use cases, requirements, attacks, datasets collection and current classification logic.

Chapter 5

Methodology

The development of an algorithm able to detect anomalous traffic plus respective integration into an already established and internationally recognised platform, as is Mobitrust, assigned a certain difficult level to the work in hands.

Therefore, it was crucial to previously define a concrete methodology in order to assure a successful outcome from the developed activities, which is explained into this chapter.

Considering the development plan inherited to the implementation of an algorithm, as well, as the several integration phases needed to integrate a new component into an already running application, the defined methodology has several steps.

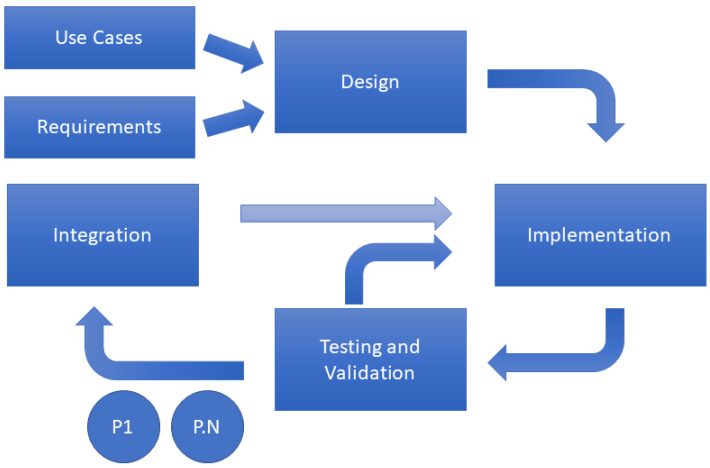


Figure 5.1: Methodology Schema - Overview

Figure 5.1 presents a macro overview of the methodology, which can be divide

into four main stages: Design, Implementation, Testing and Validation and Integration.

The Desing phase corresponds to the first stage of the defined methodology and it corresponded to the time frame where the implementation strategy was produced, among other planification activities.

During the Implementation stage, it is predicted the implementation of the candidate approaches, previously selected considering the reviewed literature. This stage considers the inputs from the Design phase, while the expected outcome of this stage corresponds to a set of implemented approaches, ready to be tested and validated. Further information on this topic may be found on section 5.2.

Over the Testing and Validation phase, it was foreseen the evaluation of the performance of the implemented approaches under different scenarios. As stated in 5.3, it was expected to exist, at least, two evaluation moments, one for the validation with the CIC-IDS2017 and another for the validation with the custom dataset. The expected output of this phase corresponds to the set of implemented approaches that present a good performance during the different validation moments to be submitted to further experimentation activities. Later, the set of approaches with good performance were considered as prototypes and proceed for the Integration phase.

During the integration phase, the prototypes were integrated within the detection module of the security framework. Further performance assessments were conduct (of the implemented approaches), namely, by using further datasets of the different implemented attacks, with different parameters, and also by verifying if the attacks were being properly identified on the security framework, using its dashboard and its *log* system.

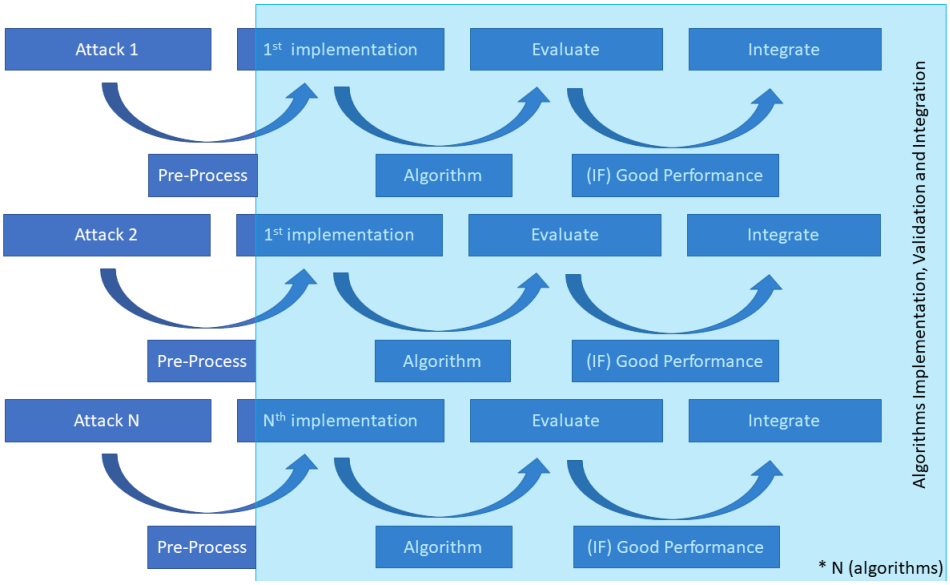


Figure 5.2: Methodology Schema - Implementation Overview

Figure 5.2 presents a macro overview of the methodology, but, specially focused on the pipeline between the collection of the datasets all the way up to the inte-

gration of the prototypes into the detection module of the security framework.

As demonstrated in the previous figure, an horizontal approach was followed, that is, after existing one attack dataset, (i) pre-process it; (ii) implement one of the selected approaches; (iii) evaluate this approach performance; (iv) (if achieved good performance) integrate the implement approach within the detection module of the security framework. Afterwards, using another dataset, repeat the process. Upon all collected and pre-processed datasets being classified with the first implemented approach, repeat the process for each of the candidate approaches.

The reason for selecting an horizontal approach is related to the need of assuring that entire cycles are completed through the pipeline, as soon as possible. This is needed to detect eventual problems into any of the different phase, so they may be earlier addressed, thus, preventing any last minute issues that could compromise the completion of the whole process.

5.1 Design

While designing any kind of implementation process, the definition of the use-cases to validate the system and the specification of requirements of the HSPF were conducted. In this document, the result of this steps are described in subsections 4.7.2 and 4.7.3 respectively.

The definition of the type of approach, even from a macro perspective, is a crucial step that may well compromise the full process, if not defined. Bearing this in mind, a macro overview of the methodology is presented in figure 5.1, while a more tailored overview of the implementation is presented in figure 5.2.

As shown in the mentioned figures, the major three phases of the methodology are: the implementation, the testing and validation and the integration, which are properly described over the next sections. Must also be noticed the horizontal approach defined for the implementation phase, while aiming to assure a full cycle between the reception of a custom attack dataset and the assessment of the different implemented approaches.

5.2 Implementation

This phase was focused on the implementation of several algorithms. During the implementation it was used open source Python [95], through the use of libraries that support artificial intelligence techniques, such as, scikit-learn [96] and TensorFlow [97].

The expected output of this phase is a functional prototype, of each of the candidate approaches to be implemented.

Taking into account the literature review reflected over chapter 3, in concrete, the possible division within ML algorithms, characteristics of each approach and

results attained, the approaches initially predicted to be replicated in this work were:

- One unsupervised ML approach
- One DL approach
- One supervised ML approach

For the unsupervised ML approach, it was envisioned to use as reference the approach described at [32] that considers a *k-means* based approach for anomaly detection. Regarding the DL approach, it was envisioned to use as reference the approach described at [48] that considers a CNN based approach. For the supervised ML approach, it was envisioned the application of a decision tree based approach, a random forest classifier.

5.3 Testing and Validation

After achieving functional prototypes, the next step is the assessment of the performance of each of the algorithms. Such assessment will take place in two separate moments: (i) the algorithm will be tested with the CIC-IDS2017 dataset; (ii) by feeding the algorithm with a custom dataset built using realistic network traffic information collected from Mobitrust communications.

During this phase, techniques like *k-fold division* [98] are envisioned to be used, as part of the validation process. Among other purposes, *k-fold division* technique can be used to estimate the performance that a machine learning model will have when dealing with unseen data.

The results obtained in this initial phase will determine if the algorithm prototype proceeds to the next phase. For this assessment, several of the metrics presented in section 2.4 will be taken into account, with a special emphasis on the following ones: *accuracy*, *recall*, *f1-score* and *precision*. The confusion matrix will also be used to better understand which samples is the algorithm wrongly classifying. For an implemented approach be able to proceed for the integration phase, preferably it should present performance values, for the previously mentioned metrics, above 98.00%. Most be noted that this objective is the result of a discussion with the OneSource team and represents a goal that is acceptable by them. Such value is also aligned with the results attained by the different approaches presented in chapter 3.

The first assessment is envisioned to be conducted in an isolated environment, that means, the algorithm will go through a phase of train, validation and test recurring to the mentioned CIC-IDS2017 dataset. Naturally, the assessment of each implemented prototype will be conducted using the same computer, under "equal" conditions, in order to minimise possible performance differences due to interference.

The second assessment is envisioned to be conducted with the dataset collected from realistic Mobitrust communications. The aim of this step is to understand the performance of these approaches with a dataset that contains samples quite similar to the ones that the prototypes (in the end) will face, when integrated with the detection module of the security framework, which will later integrate the Mobitrust architecture.

After all the existent prototypes being evaluated, a comparison will be conducted and a single approach will be selected to continue to the Integration phase. This comparison will take into account the performance of each candidate approach with each dataset and the decision will be made considering the values attained for each one of the metrics (previously mentioned).

5.4 Integration

The integration of the security framework (containing the selected AI approach) into the Mobitrust architecture was planned to be completed in two phases:

- Integration with a local Mobitrust deployment
- Preparation of the component in order to be deployable anytime, anywhere, similar to the already existent Mobitrust components

The first step was the deployment of a new component (a container) parallel to the existent Mobitrust components in a local Mobitrust deployment. At this stage, a new battery of tests were conducted to validate the capability of the algorithm to detect malicious traffic. The algorithm should present a satisfactory performance before proceeding to the next integration phase. For those approaches that eventually under performed, an unexplored (due to time constraints) but idealize path was to review the implementations, namely, by exploring other values for the configuration parameters or even reviewing previously dropped approaches.

The next step of the integration includes the preparation of an image of the Holistic Security and Privacy Framework, that included the selected AI approach. Similar to the other Mobitrust components, the security framework component should be ready to be launched in parallel with the remaining components, every time the platform is deployed.

5.5 Summary

This chapter focus on presenting the defined methodology for this work. It is described with a special emphasis on the different phases that the AI component, to be developed, must go through. The datasets used to validate the AI component in the different phases are overviewed, as well, as the candidate approaches are presented.

Regarding the different phases of the methodology, three phases are presented as the main steps while developing the AI component: implementation, testing and validation and integration. During the validation phase, two different validation moments are predicted, as well, as in the integration phase, where two levels of integration are also explained. The candidate approaches are identified, being envisioned the implementation of one supervised and unsupervised ML approach, plus a DL approach, summing 3 approaches that are expected to be implemented.

Chapter 6

Implementation

As defined in section 5, and represented in figure 5.2, the development process took into consideration an horizontal approach, that is, the process started by processing one dataset (the CIC-IDS017) and then implement, train and test one ML approach with the collected dataset. Afterwards, the collected dataset from realistic communications of the Mobitrust platform was processed and the performance of the initial algorithm is evaluated, as stated into 5.3. Finally, one-by-one, the other ML and DL approaches were implemented, tested and validated following the same validation phases.

This chapter aims to describe the several steps taken during the implementation phase, namely the pre-processing of the datasets on section 6.1 and the implementation of the candidate approaches on section 6.2.

6.1 Pre-Processing of the Datasets

This section aims to present the steps taken to process the different datasets, as well, to describe their final format and content.

Sub-section 6.1.1 describes the processing conducted over the baseline dataset, the CIC-IDS2017 [19], while sub-section 6.1.2 describes the final set of datasets originated from the collection of data from realistic Mobitrust communications, in addition to the several steps taken to achieve these versions.

6.1.1 CIC-IDS2017 dataset

The first dataset to be processed was the CIC-IDS017. This dataset contains 84 features and approximately 3119345 samples, covering 15 different classes [99].

Since it was decided that the approaches to be implemented would perform binary classification, but keeping in mind that the goal of the usage of this dataset was to compare the performance of the different implemented approaches for this dataset and the ones collected, only a portion of the dataset was used, which

corresponds to files identified as containing samples representative of the some type of attacks generated and collected.

Tables 6.1 and 6.2 illustrate the distribution of the Malignum and Beginum samples, for the portions of the dataset regarding *Dos* and *Port Scan* attacks.

Table 6.1: Class labels and samples for the *DoS* attack

		Malignum	Benignum	N_Samples
Class	0		X	97686
	1	X		128025
Total				225711

Table 6.2: Class labels and samples for the *Port Scan* attack

		Malignum	Benignum	N_Samples
Class	0		X	127292
	1	X		158804
Total				286096

6.1.2 Custom Datasets

Considering the list of attacks presented at 4.7.4 and the list of use cases 4.7.2 defined for the HSPF, two major datasets were collected, one containing a *DoS* attack and other one based on a *Port Scan*. The way the attacks were built, lunched and the originated packets were collected is presented in section 4.7.5.

Nevertheless, it's worth mentioning that it was made a request to the team in charge of collecting the dataset to provide it with the same characteristics as the CIC-IDS2017 (respectively with the same set of features), in order to enable the future comparison of the same algorithm for both datasets, so conclusions may not be taken.

Table 6.3 illustrates the representation of the classes Malignum and Benignum and respective amount of samples, per class, present in the dataset collected during the *DoS* attack.

Table 6.3: Class labels and samples for the *DoS* attack dataset

		Malignum	Benignum	N_Samples
Class	0		X	17911
	1	X		150988
Total				168899

Table 6.4 illustrates the representation of the classes Malignum and Benignum and respective amount of samples, per class, present in the dataset collected during the *Port Scan* attack.

Table 6.4: Class labels and samples for the *Port Scan* attack dataset

		Malignum	Benignum	N_Samples
Class	0		X	8250
	1	X		101801
Total				110051

6.1.3 Datasets Processing

The following points aim to describe different steps taken while processing the datasets. Must be noted that it was followed a modular and generic approach aiming to produce reusable code, so the processing of the initial datasets and the following ones, would be conducted in a similar way.

Initial Process

The first step was to remove a set of different features since it's value couldn't be present in the training dataset ('LABEL') or due to privacy concerns ('SOURCE IP' and 'DESTINATION IP'). Additionally, 'TIMESTAMP' was also discarded since the malicious attacks (per dataset) all contain relative close values, corresponding to the intervals when the datasets were collected. Therefore, the algorithm(s) to be implemented are not intended to establish any relation between the timestamps present in the training dataset and the sample classification, so they become able to correctly classify a malicious sample, even if during the training period, similar malicious samples only occurred with a different timestamp.

The next step was to proceed with the analyse of the data integrity, in specific, to look for missing values. The first approach followed to handle the samples containing missing values was to eliminate them. It was left to a later stage the exploration of other ways to handle this issue. But, due to time constraints it wasn't possible to persuit other approaches. Nevertheless, the percentage of missing values was less then 1%, thus, the impact of removing such samples is expected to don't be quite significant.

Considering the high range of values for some of the features and in order to well define the representation interval, it was decided to proceed with a min-max normalization [7].

Balancing the dataset

Acknowledging the impact that having an imbalanced dataset may have on the performance of the classifiers, it was planned from the beginning to deal with this thematic. As such, two major *oversampling* strategies were implemented: random over sampling [100] and Synthetic Minority Oversampling Technique (SMOTE) [101].

The random over sampling approach randomly duplicates samples of the minority class and adds them to the training dataset, while SMOTE synthesizes new examples from the existing samples. The bigger advantage presented by last method is the introduction of new information in the dataset, instead of just copying the previous existent samples.

Tests were conducted with different algorithms, while using these two methods aiming to explore its impact on their performance.

Finding the most discriminant features

As stated before, the collection of the different datasets from the communications of Mobitrust platform was conducted in a way that the format of the datasets was as much similar to the one presented by the CIC-IDS2017. Recognizing that the dataset contains a high number of features (above seventy-five, after some preliminary elimination) and that some of the chosen algorithms suffer a considerable impact on its performance due to the dimension of the space that the features may represent, it was decided to look for the most discriminant features.

An overview over the thematic of feature selection and reduction was conducted as part of the background knowledge for this work and is presented in section 2.2.

The test used to find the most discriminant features was the Kruskall-Wallis test [9].

Table 6.5: Kruskall-Wallis test results for the *DoS* attack, CIC-IDS2017 dataset

Feature Name	Kruskall-Wallis test
Bwd Packet Length Min	99080.73413
Bwd Packet Length Std	49767.62124
URG Flag Count	48446.67409
Fwd IAT Std	45590.11633
Init_Win_bytes_forward	44829.74887
Fwd Packet Length Max	44057.26197
Fwd Packet Length Mean	41903.22647
Avg Fwd Segment Size	41903.22647
act_data_pkt_fwd	40575.97012
Fwd IAT Max	39292.91485
min_seg_size_forward	39209.76112
Fwd IAT Mean	38771.90921
Fwd IAT Total	36482.94775
Fwd Packet Length Min	32799.73334
Total Fwd Packets	30739.14808
Subflow Fwd Packets	30739.14808
Min Packet Length	25698.67409
Flow IAT Std	25307.1426
Total Length of Fwd Packets	23704.01292
Subflow Fwd Bytes	23704.01292
Bwd Packets/s	23288.20457
Fwd Header Length	22361.18665
Fwd Header Length2	22361.18665
Average Packet Size	22307.97248
Flow Packets/s	20472.24089
Packet Length Mean	20454.1269
Flow IAT Mean	20147.64354
Flow IAT Max	18984.14606
Bwd Packet Length Mean	17358.32028

Table 6.5 presents the thirty more discriminant feature for the *DoS* attack, present in CIC-IDS2017 dataset, according to the Kruskall-Wallis test (the results for all the features may be found in Appendix).

Table 6.6: Kruskall-Wallis test results for the *Port Scan* attack, CIC-IDS2017 dataset

Feature Name	Kruskall-Wallis test
PSH Flag Count	182497.5061
Total Fwd Packets	169326.9104
Subflow Fwd Packets	169326.9104
Bwd Packets/s	167618.9582
Total Length of Fwd Packets	167292.0315
Subflow Fwd Bytes	167292.0315
Fwd Packet Length Mean	167261.7594
Avg Fwd Segment Size	167261.7594
Fwd Packet Length Max	167261.4945
Fwd IAT Total	165790.7197
Fwd IAT Max	165690.3675
Fwd IAT Mean	165651.4417
Fwd IAT Min	163930.0237
act_data_pkt_fwd	155602.1912
Packet Length Mean	143473.0151
Average Packet Size	139493.8806
Flow IAT Std	137817.9143
Flow Duration	132929.6781
Flow IAT Max	130983.8207
Max Packet Length	126206.5685
Flow Packets/s	125125.9742
Bwd IAT Total	119121.3599
Fwd Packets/s	119032.0369
Bwd IAT Max	119026.9261
Bwd IAT Mean	118988.8741
Bwd IAT Min	118118.0237
Flow IAT Mean	105515.8223
Bwd Header Length	92676.38191
Init_Win_bytes_forward	88604.82499

Table 6.6 presents the thirty more discriminant feature for the *DoS* attack, present in CIC-IDS2017 dataset, according to the Kruskall-Wallis test.

Table 6.7: Kruskall-Wallis test results for the custom dataset, *DoS* attack

Feature Name	Kruskall-Wallis test
Fwd Packet Length Max	158444.8442
Fwd PSH Flags	123488.175
Total Fwd Packets	114080.1438
Subflow Fwd Packets	114080.1438
Max Packet Length	107850.9923
Bwd Packet Length Max	106309.6122
ACK Flag Count	97870.38605
Total Backward Packets	96072.98547
Subflow Bwd Packets	96072.98547
Destination Port	91921.522
Bwd Packet Length Std	82499.11625
Bwd PSH Flags	82444.04247
PSH Flag Count	81509.41102
act_data_pkt_fwd	81499.63716
Bwd Packet Length Mean	74409.79332
Avg Bwd Segment Size	74409.79332
Total Length of Bwd Packets	73163.32716
Subflow Bwd Bytes	73163.32716
Bwd Header Length	73162.09978
Fwd Packet Length Std	48923.89282
Fwd Packet Length Mean	45961.87388
Avg Fwd Segment Size	45961.87388
Fwd Header Length	44210.93464
Total Length of Fwd Packets	44162.31432
Subflow Fwd Bytes	44162.31432
FIN Flag Count	44128.27605
Bwd IAT Total	43163.02469
Active Std	42409.57746
Active Max	42126.47179
SYN Flag Count	41352.06701

Table 6.7 presents the thirty more discriminant feature for the dataset collected from Mobitrust communications upon the execution of the *DoS* attack, according to the Kruskall-Wallis test.

Table 6.8: Kruskall-Wallis test results for the custom dataset, *Port Scan* attack

Feature Name	Kruskall-Wallis test
Fwd Packet Length Max	64705.26881
Max Packet Length	64667.59636
Protocol	43137.21341
act_data_pkt_fwd	29344.00212
Fwd PSH Flags	27249.35526
PSH Flag Count	27248.7389
Bwd PSH Flags	26838.29752
Fwd Packet Length Mean	26626.50707
Avg Fwd Segment Size	26626.50707
Init_Win_bytes_forward	26614.94027
SYN Flag Count	23874.67695
Total Length of Fwd Packets	23256.68377
Subflow Fwd Bytes	23256.68377
Average Packet Size	21703.24748
Packet Length Mean	21703.22138
Fwd Packet Length Min	18617.57364
min_seg_size_forward	18617.57364
Min Packet Length	17458.05022
Idle Mean	16962.4111
Flow IAT Mean	16938.11151
Fwd IAT Mean	16896.31238
Flow IAT Max	16801.55596
Fwd IAT Max	16800.98322
Idle Max	16792.30451
Fwd IAT Total	15960.59408
Idle Min	15930.67426
Flow_Duration	15792.04622
FIN Flag Count	14508.28748
Flow IAT Std	13710.23281
Fwd IAT Std	13606.35999

Table 6.8 presents the thirty more discriminant feature for the dataset collected from Mobitrust communications upon the execution of the *Port Scan* attack, according to the Kruskall-Wallis test.

As tables 6.5 and 6.6 illustrate, the list of discriminant features is not the same for the considered datasets. This fact was also verified upon the analysis of the values obtained for the collected dataset(s). Such evidence raises the need for the application of other methods to better determine the set of features to use under a *production* scenario. Despite this, aiming to understand the impact of the amount of features into the final performance of the implemented approaches, for each dataset, the used relevant features were the ones previously presented in tables 6.5, 6.6, 6.7 and 6.8.

Final Formats

- Experimentation Phase

  The following lists shows the final set of datasets used during the experimentation phase. Considering the investigation conducted to evaluate the impact of the balancing strategy, as well as of the identification of the discriminant features, into the performance of the algorithms, several datasets were generated from the original ones in order to allow this exploration.

  The processed datasets attained from the CIC-IDS2017 are described next:

  For the *Dos* attack:

  - cic-ids2017_DoS_a_i : considered all the 79 features (using random oversampling)
  - cic-ids2017_DoS_a_ii : considered all the 79 features (using SMOTE)
  - cic-ids2017_DoS_a_iii : considered all the 79 features (no balancing)
  - cic-ids2017_DoS_b_i : considered the 30 more discriminant features (using random oversampling)
  - cic-ids2017_DoS_b_ii : considered the 30 more discriminant features (using SMOTE)
  - cic-ids2017_DoS_b_iii : considered the 30 more discriminant features (no balancing)
  - cic-ids2017_DoS_c_i : considered the 20 more discriminant features (using random oversampling)
  - cic-ids2017_DoS_c_ii : considered the 20 more discriminant features (using SMOTE)
  - cic-ids2017_DoS_c_iii : considered the 20 more discriminant features (no balancing)
  - cic-ids2017_DoS_d_i : considered the 15 more discriminant features (using random oversampling)
  - cic-ids2017_DoS_d_ii : considered the 15 more discriminant features (using SMOTE)
  - cic-ids2017_DoS_d_iii : considered the 15 more discriminant features (no balancing)
  - cic-ids2017_DoS_e_i : considered the 10 more discriminant features (using random oversampling)
  - cic-ids2017_DoS_e_ii : considered the 10 more discriminant features (using SMOTE)
  - cic-ids2017_DoS_e_iii : considered the 10 more discriminant features (no balancing)

  For the *Port Scan* attack:

  - cic-ids2017_PS_a_i : considered all the 79 features (using random oversampling)

- cic-ids2017_PS_a_ii : considered all the 79 features (using SMOTE)
- cic-ids2017_PS_a_iii : considered all the 79 features (no balancing)
- cic-ids2017_PS_b_i : considered the 30 more discriminant features (using random oversampling)
- cic-ids2017_PS_b_ii : considered the 30 more discriminant features (using SMOTE)
- cic-ids2017_PS_b_iii : considered the 30 more discriminant features (no balancing)
- cic-ids2017_PS_c_i : considered the 20 more discriminant features (using random oversampling)
- cic-ids2017_PS_c_ii : considered the 20 more discriminant features (using SMOTE)
- cic-ids2017_PS_c_iii : considered the 20 more discriminant features (no balancing)
- cic-ids2017_PS_d_i : considered the 15 more discriminant features (using random oversampling)
- cic-ids2017_PS_d_ii : considered the 15 more discriminant features (using SMOTE)
- cic-ids2017_PS_d_iii : considered the 15 more discriminant features (no balancing)
- cic-ids2017_PS_e_i : considered the 10 more discriminant features (using random oversampling)
- cic-ids2017_PS_e_ii : considered the 10 more discriminant features (using SMOTE)
- cic-ids2017_PS_e_iii : considered the 10 more discriminant features (no balancing)

Regarding the final datasets attained after processing the raw sets collected from realistic communications of the Mobitrust application, while collecting the different attacks, they are presented in the next lists:

The ones generated from the *DoS* attack, were:

- DoS_a_i : considered all the 79 features (using random oversampling)
- DoS_a_ii : considered all the 79 features (using SMOTE)
- DoS_a_iii : considered all the 79 features (no balancing)
- DoS_b_i : considered the 30 more discriminant features (using random oversampling)
- DoS_b_ii : considered the 30 more discriminant features (using SMOTE)
- DoS_b_iii : considered the 30 more discriminant features (no balancing)
- DoS_c_i : considered the 20 more discriminant features (using random oversampling)

- DoS_c_ii : considered the 20 more discriminant features (using SMOTE)
- DoS_c_iii : considered the 20 more discriminant features (no balancing)
- DoS_d_i : considered the 15 more discriminant features (using random oversampling)
- DoS_d_ii : considered the 15 more discriminant features (using SMOTE)
- DoS_d_iii : considered the 15 more discriminant features (no balancing)
- DoS_e_i : considered the 10 more discriminant features (using random oversampling)
- DoS_e_ii : considered the 10 more discriminant features (using SMOTE)
- DoS_e_iii : considered the 10 more discriminant features (no balancing)

The ones generated from the *Port Scan* attack, were:

- PS_a_i : considered all the 79 features (using random oversampling)
- PS_a_ii : considered all the 79 features (using SMOTE)
- PS_a_iii : considered all the 79 features (no balancing)
- PS_b_i : considered the 30 more discriminant features (using random oversampling)
- PS_b_ii : considered the 30 more discriminant features (using SMOTE)
- PS_b_iii : considered the 30 more discriminant features (no balancing)
- PS_c_i : considered the 20 more discriminant features (using random oversampling)
- PS_c_ii : considered the 20 more discriminant features (using SMOTE)
- PS_c_iii : considered the 20 more discriminant features (no balancing)
- PS_d_i : considered the 15 more discriminant features (using random oversampling)
- PS_d_ii : considered the 15 more discriminant features (using SMOTE)
- PS_d_iii : considered the 15 more discriminant features (no balancing)
- PS_e_i : considered the 10 more discriminant features (using random oversampling)
- PS_e_ii : considered the 10 more discriminant features (using SMOTE)
- PS_e_iii : considered the 10 more discriminant features (no balancing)

6.2 Implementation of the Candidate Approaches

This section aims to provide an overview of the logic and pipeline behind the implementation of the candidate approaches.

6.2.1 Training, testing and validation overview

As stated on chapter 5, the programming language used during the implementation process was *Python* [95], more in concrete, the Python 3.8.10. The libraries used during the implementation of the different approaches were: scikitlearn [26] and TensorFlow [27].

Figure 6.1 presents an overview of the implementation followed while implementing the different approaches.

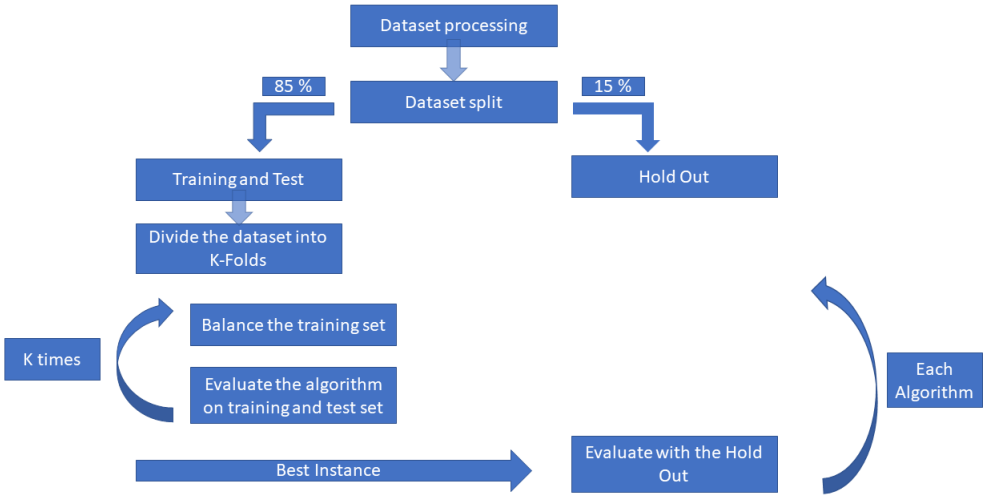


Figure 6.1: Approaches Implementation - Overview

After dealing with the dataset processing, explained in detail in the previous section 6.1, the next step was to divide the dataset in a way to leave an hold out section, later used for validation purposes. Also part of the validation strategy, it was used a *k-fold* cross-validation method, where for each *k-fold*, containing a set for training and for testing, the training set was balanced (using a SMOTE approach) and then the algorithm was trained and evaluated on the training and test sections.

The value used for *K* was 5, acknowledging that 5 or 10 are values that are currently pointed out as appropriate in the literature ([102], [103]). Despite this, at [98] the authors further elaborate on this matter and present mathematical evidence on the importance that the *K* value may have on the performance of the algorithms.

On each iteration, a new instance of the current algorithm was trained with the corresponding set of training (after it being balanced) and later this instance was evaluated with the training and test sets. The metrics used to decide which instance would be selected as the best, were: accuracy, f1-score, precision and recall.

Upon finding the best instance, it was tested with the Hold-Out section of the dataset and it's performance was registered using the same set of metrics previously mentioned.

6.2.2 Implementation Details

This section aims to present some considerations regarding the implementations of the selected approaches. The specific used libraries for each implementation are described here after.

Random Forest

Random Forest was implemented using *sklearn.ensemble.RandomForestClassifier* [104] class. No specific parameters were given, thus, the default values were used. Table 6.9 summits the default values of this implementation.

Table 6.9: Parameter values for Random Forest implementation

Parameter	Value
n_estimators	100
criterion	gini
max_depth	None
min_samples_split	2
min_samples_lead	1
min_weight_fraction_lead	0.0
max_features	sqrt
max_leaf_nodes	None
min_impurity_decrease	0.0
bootstrap	True
oob_score	False
n_jobs	None
random_state	None
verbose	0
warm_start	False
class_weight	None
ccp_alpha	0.0
max_samples	None

K-means

K-means was implemented using *sklearn.cluster.KMeans* class [105]. Considering the importance of the seed into a clustering algorithm, a special emphasis was given to the initial random state of the algorithm. Furthermore, the number of clusters was set into two, which corresponds to the number of classes (binary classification) of the problem in hands, while the remaining parameters where not assigned specific values, thus, the default values were used. Furthermore, table 6.10 presents the values used for the input parameters.

Table 6.10: Parameter values for *K-Means* implementation

Parameter	Value
n_clusters	2
init	k-means++
n_init	10
max_iter	300
tol	$1e-4$
verbose	0
random_state	{1,92,167,208,1942}
copy_x	True
algorithm	*lloyd*

SVM

SVM was implemented using *sklearn.svm.SVC* class [106]. Acknowledging the importance of the C and γ input parameters for the kernel functions, a first attempt to identify the best combination of parameters was conducted. In addition, two kernels were selected for these experiments: "linear" and the "rbf". This selection occurred considering that those represent the most common functions used in similar approaches and also because it wouldn't be possible to be conducting experiments with all the available kernel functions. Table 6.11 summits the input parameter values used during the initial experimentation process.

Table 6.11: Parameter values for *SVM* implementation

Parameter	Value
C	{1.00e-02, 2.5e+00, 6.31e+02, 1.58e+05, 3.98e+07, 1.00e+10}
kernel	{"linear","rbf"}
gamma	{1.00e-09, 2.51e-07, 6.31e-05, 1.58e-02, 3.98e+00, 1.00e+03}
coef0	0.0
shrinking	True
probability	False
tol	1e-3
cache_size	200
class_weight	None
verbose	False
max_iter	-1
decision_function_shape	"ovr"
break_ties	False

Considering the attained results during this first experimentation, C and *gamma* values will be selected to be used during the evaluation of performance with the considered datasets. More information on this may be found under section 7.3.

CNN

CNN was implemented using a sequential model [107], which is a structure that allows to group several layers to build a model, in this case, a neural network.

In addition and having in mind the approach presented at [48], the following list of layers were used: 1D convolution layer [108], Dense layer [109], Flatten layer [110] and a MaxPooling1D layer [111].

Figure 6.2 presents the structure of the implemented approach.

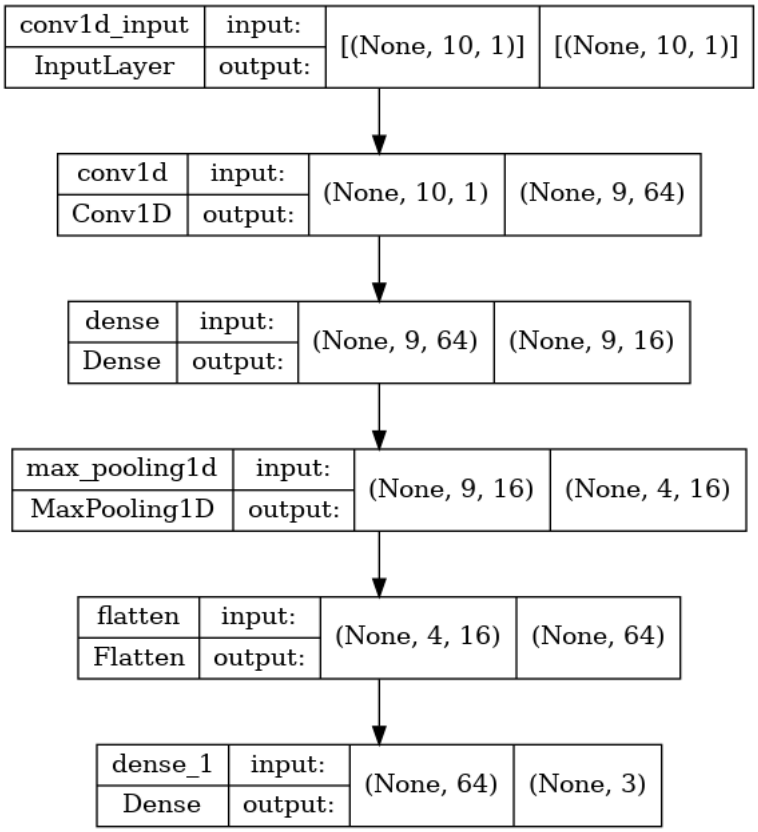


Figure 6.2: CNN Structure

Must be noted that during the several experiments the "input shape" of the 1D Convolution layer was adapted to fit the number of features present within each dataset. As such, figure 6.2 illustrates an occasion where the algorithm was handling a dataset with ten input features.

Due to the amount of possibilities for the input parameters, for each of the used layers, table 6.12 only presents the input parameters that were manually set.

Table 6.12: Set input parameter values for *CNN* implementation

Layer	Parameter	Value
Conv1D	filters	64
	kernel_size	2
	activation	"relu"
	input_shape	«dynamic_value»
Dense	units	16
	activation	"relu"
MaxPooling1D	No specific parameters set	
Flatten	No specific parameters set	
MaxPooling1D	units	3
	activation	"softmax"

Before being trained and evaluated, the model was compiled using the parameters described in table 6.13.

Table 6.13: Parameter values used to compile the *CNN* implementation

Parameter	Value
loss	sparse_categorical_crossentropy
optimizer	adam
metrics	accuracy

6.3 Summary

This chapter covers two major aspects: the pre-processing of the datasets and the implementation of the candidate approaches. As part of the data pre-processing, a description is made of the CIC-IDS2017 and of the custom datasets. The entire process of pre-processing was described in detail, followed with the identification of the most discriminant features, determined using the Kruskall-Wallis test [9].

Must be noted that, as part of the data pre-processing, one of the challenges was the handling of missing values commonly present into datasets. Due to time restrictions and considering that the priority was on the correct implementation and assessment of the selected approaches, it wasn't possible to pursuit the application of different methods to handle the problem of missing values, being the selected approach theirs elimination. Despite this, the number of missing data was not significant, considering the total of samples present into each dataset (less than 1% of the samples). For better identification of all the dataset variants, a list is presented properly identifying all of them.

Afterwards, several aspects related to the implementation of the candidate approaches are presented. First is overviewed the entire training, testing and validation cycle that each implementation was subject to. Then, some details regarding the implementation of the different approaches are presented, with the identification of the used libraries and input parameters.

Chapter 7

Results

The following sections present the results attained for the different implemented approaches, by summing the results into a best/worst performance way, for each of the datasets used during the experimentation phase and also used during the integration phase, for those approaches that proceed to this phase. Furthermore, there are also some indications on the impact of the amount of used features, as well, as on impact of the balancing strategy.

Must be noted that the presented results were attained through a *Python* script that processed the main scripts output, thus, the decimal presented values were formatted using two decimal places, while integer values are presented in its original form. In addition, the "N_Features" column represents the amount of features present in the used sub-set of the dataset, while the "Balancing" column represents the applied balancing strategy. For the tests conducted without the application of any balancing strategy, it is presented an "NA", stating Not Applicable.

The first section (section 7.1) presents the results attained with the Random Forest implementation, followed by the ones attained with the *k-means* implementation (section 7.2). Additionally, section 7.3 presents the results attained with the SVM implementation, while section 7.4 present the results attained with the CNN implementation.

7.1 Random Forest

This section aims to present the results achieved with the *Random Forest* implementation, when trained and evaluated with the CIC-IDS2017 and with the custom datasets.

Random Forest presented an excellent performance for all the majority of the considered datasets, achieving maximum values for *precision*, *recall*, *f1-score* and for *accuracy*. As such, some of the following sections only present one table that corresponds to an example of the values achieved, since in this situation, there isn't a worst performance case to be presented

7.1.1 CIC-IDS2017 (*DoS* Attack)

Table 7.1 presents an example of the performances attained for the cic-ids2017_DoS_* datasets.

Table 7.1: Performance of *Random Forest* with the cic-ids2017_DoS_d_i dataset

Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16627 0] [2 21742]	15	Random OverSampling

Impact of the amount of features

As mentioned before, the *Random Forest* classifier provided classifications with maximum values for the metrics considered, for the majority of the experiments conducted, as such, the effect of the number of features was not evident during these experiments.

Impact of the dataset balancing strategy

As mentioned before, the *Random Forest* classifier provided classifications with maximum values for the metrics considered, for the majority of the experiments conducted, as such, the effect of the balancing strategy was not evident during these experiments.

7.1.2 CIC-IDS2017 (*Port Scan (PS)* Attack)

Table 7.2 presents an example of the performances attained for the cic-ids2017_PS_* datasets.

Table 7.2: Performance of *Random Forest* with the cic-ids2017_PS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21648 0] [5 26984]	79	SMOTE

Impact of the amount of features

The *Random Forest* classifier provided classifications with maximum values for the metrics considered, for all the conducted experiments. As such, the effect of the number of features was not evident during these experiments.

Impact of the dataset balancing strategy

The *Random Forest* classifier provided classifications with maximum values for the metrics considered, for all the conducted experiments. As such, the effect of

the balancing strategy was not evident during these experiments.

7.1.3 Custom Dataset (*DoS* Attack)

Table 7.3 presents an example of the performances attained for the DoS_* datasets.

Table 7.3: Performance of *Random Forest* with the DoS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[3046 0] [0 25667]	20	NA

Impact of the amount of features

The *Random Forest* classifier provided classifications with maximum values for the metrics considered, for all the conducted experiments. As such, the effect of the number of features was not evident during these experiments.

Impact of the dataset balancing strategy

The *Random Forest* classifier provided classifications with maximum values for the metrics considered, for all the conducted experiments. As such, the effect of the balancing strategy was not evident during these experiments.

7.1.4 Custom Dataset (*PS* Attack)

Table 7.3 presents an example of the performances attained for the PS_* datasets.

Table 7.4: Performance of *Random Forest* with the PS_* dataset

Best performance							
Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[1409 1] [0 17298]	30	SMOTE
Worst performance							
Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
0 1	0.91 1.00	1.00 0.99	0.95 1.00	0.99	[1197 0] [114 15197]	10	NA

This classifier presented maximum values for the majority of sub-sets considered while evaluating its performance with the PS attack. Despite this, when reducing the number of features bellow twenty, corresponding to the datasets PS_d,e_i,ii,iii, the performance of the classifier presented a significant reduction. As an example, the "Best Performance" values were attained using the PS_b_iii dataset, whereas the "Worst Performance" values were achieved using the *Hold Out* section of the PS_e_i dataset.

Impact of the amount of features

The experiments conducted with the datasets (PS_b,c_i,ii,iii) didn't show any evidence of the impact of decreasing the amount of features. Despite this, while reducing the amount of features for thirty and for twenty. On other hand, while using only the fifteen and ten most discriminant features, it was possible to notice a slightly decrease on the performance of the algorithm. As presented in table 7.4, for the case of "Worst Performance", it is possible to see an increase of false negatives, which then results in a decrease of precision and f1-score for the malignum class.

Since this behaviour was consistent throughout all of the folds used during the training of the classifier(s), it suggests that the reduction of features, at least, for this particular case, results in a lost of performance by the classifier.

Impact of the dataset balancing strategy

The performance attained by the algorithm for the experiments conducted with the datasets (PS_{a,b,c}_{i,ii,iii}) was ideal, presenting maximum values for all the of the considered metrics. Despite this and similar to the impact of reducing the amount of featues, it was possible to observe an impact on the classifier performance, due to the balancing strategy used. Table 7.5 presents some of the results attained that show this impact.

Table 7.5: Performance of *Random Forest* with the PS_* dataset(s)

Hold Out section of the PS_d_ii							
Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
0 1	0.91 1.00	1.00 0.99	0.95 1.00	0.99	[1197 0] [114 15197]	15	SMOTE
Hold Out section of the PS_d_iii							
Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
0 1	1.00 1.00	0.99 1.00	0.99 1.00	0.99	[1181 16] [0 15311]	10	NA

While for the *Hold Out* section of the PS_d_ii dataset, where SMOTE was used during the training of the algorithm, it's possible to register a considerable decrease on the *precision* and *f1-score* for the malignum classes, due to the number of false negatives, for the *Hold Out* section of the PS_d_iii dataset, where no balancing strategy was applied during the training of the algorithm, it's possible to see an increase of the false positives.

Even though the highlighted impact of the balancing strategy results in a decrease of performance no bigger than 10%, it is still notable and must be considered. As such, at least for this case, it is possible to state that the balancing strategy used during the training phase had an effective impact on the algorithm(s) performance.

7.2 k-means

This section aims to present the results achieved with the implementation, when trained and evaluated with the CIC-IDS2017 and with the custom datasets.

7.2.1 CIC-IDS2017 (*DoS* Attack)

Table 7.6 presents the best and worst performances by the *K-means* for the cic-ids20171_DoS_* dataset.

Table 7.6: Performance of *k-means* with the cic-ids2017_DoS_a_i dataset

Best performance								
Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
1942	0 1	0.66 0.92	0.93 0.63	0.77 0.75	0.76	[15385 1242] [7991 13753]	30	SMOTE
Worst performance								
Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
92	0 1	0.34 0.00	0.68 0.00	0.45 0.00	0.29	[11254 5372] [21745 0]	20	Random Oversampling

The best and worst performance of the *K-means* were achieved, respectively, for the cic-ids2017_DoS_b_ii and for the cic-ids2017_DoS_c_i datasets. Must be noted that similar results were achieved when using the same amount of features, but, with different balancing strategies, thus the one presented here only aims to provide an example of such performance. Furthermore and specifically for the worst performance case, the one presented was selected considering that the algorithm was not able to correctly classify one sample of the malignum class. Despite this, there were classifications that presented lower *accuracy* and lower *precision*, *recall* and *f1-measure* for the benignum class, but, higher values of these metrics for the malignum class (e.g. situations when the algorithm was able to correctly classify some of the malignum samples).

Impact of the *seed*

k-means algorithm is a clustering algorithm, thus, the initial position of the clusters is extremely important for the algorithm ability to correctly classify samples of different classes.

Throughout the inteire experimentation process, for different datasets, within each *k-fold*, it was possible to verify the impact that the seed has on the performance of *k-means* algorithm. Table 7.7 presents an illustrative example of this fact.

Table 7.7: (Partial) Performance of *k-means* with the cic-ids2017_DoS_c_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
3	1	0 1	0.34 0.00	0.67 0.00	0.45 0.00	0.29	[11150 5476] [21744 1]	15	Random Over-sampling
	92	0 1	0.34 0.00	0.67 0.00	0.45 0.00	0.29	[11150 5476] [21744 1]		
	167	0 1	1.00 0.66	0.33 1.00	0.50 0.80	0.71	[5476 11150] [1 21744]		
	208	0 1	1.00 0.66	0.33 1.00	0.50 0.80	0.71	[5476 11150] [1 21744]		
	1942	0 1	0.34 0.00	0.67 0.00	0.45 0.00	0.29	[11150 5476] [21744 1]		

From table 7.7 it is possible to see that for the same *k-fold*, the algorithm attained distinct classifications, where the only relevant difference is precisely the used *Seed*. Having this in mind, it is clear that the *Seed* has a critical impact on the performance of the algorithm.

Impact of the amount of features

Acknowledging the impact that the *Seed* presents on the performance of this algorithm, to better understand the impact of the amount of used features, it is necessary to analyse the same *k-folds* of the different datasets, that is, through the experiments conducted with the different variants of the original dataset (e.g. cic-ids2017_DoS_{a,b,c,d,e}). For the sake of comparison, it will only be considered the experiments conducted where no balancing strategy was used during the training of the algorithm.

Table 7.8: (Partial) Performance of *k-means* with the cic-ids2017_DoS_{a,b,c,d,e}_iii dataset(s)

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix
cic-ids2017_DoS_a_iii							
2	1	0	0.24	0.19	0.21	0.39	[3080 13546]
		1	0.47	0.54	0.50		[9934 11811]
	92	0	0.53	0.81	0.65	0.61	[13546 3080]
		1	0.76	0.46	0.57		[11811 9934]
	167	0	0.24	0.19	0.21	0.39	[3080 13546]
		1	0.47	0.54	0.50		[9934 11811]
	208	0	0.53	0.81	0.65	0.61	[13547 3079]
		1	0.76	0.46	0.57		[11811 9934]
	1942	0	0.24	0.19	0.21	0.39	[3079 13547]
		1	0.47	0.54	0.50		[9934 11811]
cic-ids2017_DoS_b_iii							
2	1	0	0.08	0.07	0.08	0.24	[1222 15404]
		1	0.34	0.36	0.35		[13896 7849]
	92	0	0.66	0.93	0.77	0.76	[15404 1222]
		1	0.92	0.64	0.75		[7849 13896]
	167	0	0.08	0.07	0.08	0.24	[1222 15404]
		1	0.34	0.36	0.35		[13896 7849]
	208	0	0.66	0.93	0.77	0.76	[15404 1222]
		1	0.92	0.64	0.75		[7849 13896]
	1942	0	0.17	0.17	0.17	0.28	[2830 13796]
		1	0.36	0.36	0.36		[13896 7849]
cic-ids2017_DoS_c_iii							
2	1	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
	92	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
	167	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
	208	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
	1942	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
cic-ids2017_DoS_d_iii							
2	1	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
	92	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
	167	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
	208	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
	1942	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
cic-ids2017_DoS_e_iii							
2	1	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	92	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	167	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	208	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	1942	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]

Table 7.8 exhibits the results attained by the *k-means* for the cic-ids2017_DoS_{a,b,c,d,e}_iii datasets, in specific, the results for the second *k-fold*. Even though there was no specific criteria for selecting this *k-fold*, it is presented here as an illustrative example and to facilitate some comparison among the different metric values.

Considering the results attained for the dataset with seventy-nine features and the ones attained for the one with thirty, in the best case, it is possible to observe a clear improvement for all the metrics considered with the *accuracy* improving by 124.59%, reaching 0.76.

For the experiments conducted with the twenty and fifteen features, it is not possible to register any difference among the results, whereas for experiments with ten features the results show an approximation with the best case detected, with the accuracy reaching 0.71 and the f1-measure registering 0.48 and 0.79, for the benignum and malignum classes, respectively.

Considering the results previously presented, it may be stated that the amount of used features has an impact on the performance of the algorithm, even though if this effect is not visible for all the cases, which is eventually associated with the impact of the *Seed*.

Impact of the dataset balancing strategy

Through the analysis of the collected results concerning the performance of *k-means* with all the datasets generated from the cic-ids2017_DoS, it was only possible to evidence a situation where the impact of the balancing strategy is clearly observed.

Table 7.9: (Partial) Performance of *k-means* with the cic-ids2017_DoS_c_{ii,iii} dataset(s)

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix
cic-ids2017_DoS_c_ii							
1	1	0 1	0.34 0.00	0.67 0.00	0.45 0.00	0.29	[11148 5479] [21744 0]
	92	0 1	0.34 0.00	0.67 0.00	0.45 0.00	0.29	[11148 5479] [21744 0]
	167	0 1	0.34 0.00	0.67 0.00	0.45 0.00	0.29	[11148 5479] [21744 0]
	208	0 1	0.34 0.00	0.67 0.00	0.45 0.00	0.29	[11148 5479] [21744 0]
	1942	0 1	0.34 0.00	0.67 0.00	0.45 0.00	0.29	[11148 5479] [21744 0]
cic-ids2017_DoS_c_iii							
1	1	0 1	0.44 0.63	0.87 0.16	0.59 0.26	0.47	[14486 2141] [18157 3587]
	92	0 1	0.37 0.56	0.13 0.84	0.19 0.67	0.53	[2141 14486] [3587 18157]
	167	0 1	0.44 0.63	0.87 0.16	0.59 0.26	0.47	[14486 2141] [18157 3587]
	208	0 1	0.44 0.63	0.87 0.16	0.59 0.26	0.47	[14486 2141] [18157 3587]
	1942	0 1	0.44 0.63	0.87 0.16	0.59 0.26	0.47	[14486 2141] [18157 3587]

As illustrated in table 7.9, for the same *k-fold*, there is a clear improvement between the use of *SMOTE* as balancing strategy and the case where no balancing strategy was used. In the first case, the algorithm is not able to correctly classify any of the malignum samples, where, as for the second case, the amount of correct malignum classifications considerably increases, leading to the improvement of the metric values.

Since it was only possible to identify a case of clear impact when using thirty features, it is not possible to clearly state if the balancing strategy presents a significant impact on the algorithms performance, at least, for the cic-ids2017_DoS_* datasets.

7.2.2 CIC-IDS2017 (*PS* Attack)

Table 7.10 presents the best and worst performances by the *K-means* for the cic-ids20171_PS_* dataset(s).

Table 7.10: Best and Worst Performance of *k-means* with the cic-ids2017_PS_* dataset

Best performance									
K	Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
1	1	0 1	1.00 0.84	0.76 1.00	0.87 0.91	0.89	[16524 5124] [7 26982]	79	Random Oversampling
Worst performance									
K	Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
1	92	0 1	0.16 0.00	0.24 0.00	0.19 0.00	0.11	[5124 16524] [26982 7]	79	Random Oversampling

The best and worst achieved performances were both for the cic-ids2017_PS_a_i dataset. The difference between the results is present throughout the different datasets resulting from the experiments with seventy nine features and with different balancing strategies applied, that is, between the cic-ids2017_PS_a_{i,ii,iii} datasets.

Impact of the *seed*

As presented in table 7.10, for the same dataset and for the same *k-fold*, the algorithm presented contrasting results, for different seeds.

Considering the results collected throughout all of the experimentation activities, it was possible to verify a constant discrepancy between the attained results, for the same *k-fold*. With this is mind, it is possible to state that, at least for this set of datasets, the *Seed* presents a critical impact over the algorithms performance.

Impact of the amount of features

The major and only significant difference between the performances attained by the algorithm were registered for the experiments conducted with 79 features, in comparison to the ones conducted with the remaining amount of features. As such, table 7.11 illustrates this difference.

Table 7.11: (Partial) Performance of *k-means* with the cic-ids2017_PS_a,c_i dataset(s)

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix
cic-ids2017_PS_a_i							
3	1	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
	92	0	1.00	0.76	0.86	0.89	[16374 5273]
		1	0.84	1.00	0.91		[10 26979]
	167	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
	208	0	1.00	0.76	0.86	0.89	[16374 5273]
		1	0.84	1.00	0.91		[10 26979]
	1942	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
cic-ids2017_PS_c_i							
4	1	0	0.59	0.88	0.70	0.67	[18980 2667]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.59	0.88	0.70	0.67	[18980 2667]
		1	0.84	0.50	0.63		[13411 13578]
	167	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	208	0	0.16	0.12	0.14	0.33	[2658 18989]
		1	0.41	0.50	0.45		[13578 13411]
	1942	0	0.16	0.12	0.14	0.33	[2667 18980]
		1	0.41	0.50	0.45		[13578 13411]

It is possible to verify an exchange over the panoply of results for the cic-ids2017_PS_a_i and the cic-ids2017_PS_b_i datasets. For the first, either the algorithm presents its best classification for this dataset either presents its worst. With the reduction for 30 features, the algorithm present less distant classifications, presenting *accuracy* values around 0.33 and 0.67 and *f1-measure* reads around 0.70 and 0.63 for the best case and 0.14 and 0.45 for the worst one, both respectively for the malignum and benignum classes.

The reductions made, while considering the 20, 15 and 10 most discriminant features, didn't exactly cause an impact on the algorithms performance, being this experiments presenting results somehow similar among them and to the ones achieved for the cic-ids2017_PS_b_{i,ii,iii} datasets.

Considering the results previously mentioned, the impact of the used amount of features is evident, specially when reducing from 79 to 30 features, for the considered set of datasets (e.g., cic-ids2017_PS_{a,b,c,d,e}_{i,ii,iii}).

Impact of the dataset balancing strategy

Analysing the results gathered for the set of datasets in question it wasn't possible to highlight any substantial change.

Must be noted that there are some punctual performance metrics values that eventually could be a result of the impact caused by the used balancing strategy, however, since it's a small number of situations, these minor differences were

attributed to the impact of the *Seed* on the algorithms performance.

7.2.3 Custom Dataset (*DoS* Attack)

Table 7.12 presents the best and worst performances by the *K-means* for the DoS_* dataset(s).

Table 7.12: Best and Worst Performance of *k-means* with the DoS_* dataset

Best performance									
K	Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
4	208	0 1	1.00 0.90	0.01 1.00	0.02 0.94	0.90	[36 3010] [0 25667]	79	Random Oversampling
Worst performance									
K	Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
4	1942	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3010 36] [25667 0]	79	Random Oversampling

The performance of the *k-means* algorithm with the *DoS* attack was far way from the ideal, presenting the most of the experiments, quite low values for the considered metrics. For the best performance, despite of the good overall metric values, except for the *recall* and *f1-measure*, must be noted that the algorithm is classifying the majority of the samples as being malignum. For the worst performance case, it is quite of the opposite, that is, the algorithm is tending to classify all of the samples as being benignum.

Impact of the *Seed*

Due to the overall low performance values, being the most of the performances presenting *accuracy* values around 0.10 and *f1-measure* values around 0.19 and 0.0 for the malignum and the benignum classes, respectively, it isn't possible to draw any inferences about the impact of the *Seed*, regarding the considered datasets.

Impact of the amount of features

As stated before, for the considered experiments, this algorithm presented quite low performance values, thus, it is not possible to comprehend if there is any impact caused by the amount of used features.

Impact of the dataset balancing strategy

As stated before, for the considered experiments, this algorithm presented quite low performance values, thus, it is not possible to realize if there is any impact caused by the used balancing strategy.

7.2.4 Custom Dataset (*PS* Attack)

The performances attained by the algorithm for the PS_* datasets were all extremely poor, thus, table 7.12 only presents an example of the achieved classifications.

Table 7.13: Performance of *Random Classifier* classifier with the cic-ids2017_PS_a_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
5	167	0 1	0.07 0.00	0.99 0.00	0.14 0.00	0.07	[1390 20] [17298 0]	79	Random Oversampling

7.2.5 Impact of the *Seed*

As stated before, for the considered experiments, this algorithm presented quite low performance values, thus, it is not possible to state if there is any impact caused by *Seed*.

Impact of the amount of features

As stated before, for the considered experiments, this algorithm presented quite low performance values, thus, it is not possible to comprehend if there is any impact on the amount of used features.

Impact of the dataset balancing strategy

As stated before, for the considered experiments, this algorithm presented quite low performance values, thus, it is not possible to realize if there is any impact caused by the used balancing strategy.

7.3 SVM

This section aims to present the results achieved with the SVM implementation, starting by the results achieved upon the input parameters search grid, followed by the results attained when training and evaluating the implemented approach with the CIC-IDS2017 and with the custom datasets.

7.3.1 Search Grid Parameters

This sub-section presents the results achieved from the search grid conducted to explore the impact of the C and γ parameters into the performance of the algorithm. The tables in this section contain the fifteen results with higher scores,

using the R^2 [14], ordered by the average of the scores attained for each of the k-folds.

Table 7.14: Search grid parameters with the cic-ids2017_DoS_a dataset

C	γ	kernel	K=1	K=2	K=3	K=4	K=5	Average
1.00e+03	1.00e-01	rbf	0.999	0.999	0.999	0.999	0.999	0.999
1.00e+03	1.00e-05	linear	0.999	0.999	0.999	0.999	0.999	0.999
1.00e+03	2.15e-04	linear	0.999	0.999	0.999	0.999	0.999	0.999
1.00e+03	4.64e-03	linear	0.999	0.999	0.999	0.999	0.999	0.999
1.00e+03	1.00e-01	linear	0.999	0.999	0.999	0.999	0.999	0.999
1.00e+03	4.64e-03	rbf	0.987	0.987	0.987	0.987	0.987	0.987
4.64e+00	1.00e-05	linear	0.985	0.985	0.986	0.985	0.985	0.985
4.64e+00	2.15e-04	linear	0.985	0.985	0.986	0.985	0.985	0.985
4.64e+00	4.64e-03	linear	0.985	0.985	0.986	0.985	0.985	0.985
4.64e+00	1.00e-01	linear	0.985	0.985	0.986	0.985	0.985	0.985
4.64e+00	1.00e-01	rbf	0.983	0.984	0.984	0.984	0.983	0.984
1.00e+03	2.15e-04	rbf	0.982	0.983	0.984	0.983	0.983	0.983
4.64e+00	4.64e-03	rbf	0.981	0.982	0.982	0.982	0.982	0.982
2.15e-02	1.00e-05	linear	0.977	0.978	0.979	0.978	0.978	0.978
2.15e-02	2.15e-04	linear	0.977	0.978	0.979	0.978	0.978	0.978
2.15e-02	4.64e-03	linear	0.977	0.978	0.979	0.978	0.978	0.978

Table 7.15: Search grid parameters with the PS_a dataset

C	γ	kernel	K=1	K=2	K=3	K=4	K=5	Average
4.64e+00	1.00e-01	rbf	0.997	0.997	0.997	0.996	0.997	0.997
1.00e+03	4.64e-03	rbf	0.996	0.997	0.997	0.996	0.997	0.996
1.00e+03	1.00e-01	rbf	0.996	0.997	0.997	0.996	0.996	0.996
1.00e+03	1.00e-05	linear	0.984	0.986	0.986	0.985	0.984	0.985
1.00e+03	2.15e-04	linear	0.984	0.986	0.986	0.985	0.984	0.985
1.00e+03	4.64e-03	linear	0.984	0.986	0.986	0.985	0.984	0.985
1.00e+03	1.00e-01	linear	0.984	0.986	0.986	0.985	0.984	0.985
4.64e+00	1.00e-05	linear	0.984	0.985	0.985	0.985	0.984	0.985
4.64e+00	2.15e-04	linear	0.984	0.985	0.985	0.985	0.984	0.985
4.64e+00	4.64e-03	linear	0.984	0.985	0.985	0.985	0.984	0.985
4.64e+00	1.00e-01	linear	0.984	0.985	0.985	0.985	0.984	0.985
1.00e+03	2.15e-04	rbf	0.984	0.985	0.986	0.985	0.984	0.985
4.64e+00	4.64e-03	rbf	0.984	0.985	0.986	0.985	0.984	0.985
2.15e-02	1.00e-05	linear	0.984	0.985	0.985	0.984	0.983	0.984
2.15e-02	2.15e-04	linear	0.984	0.985	0.985	0.984	0.983	0.984

Table 7.16: Search grid parameters with the DoS_a dataset

C	γ	kernel	K=1	K=2	K=3	K=4	K=5	Average
4.64e+00	1.00e-01	rbf	1.000	1.000	1.000	1.000	1.000	1.000
1.00e+03	4.64e-03	rbf	1.000	1.000	1.000	1.000	1.000	1.000
1.00e+03	1.00e-01	rbf	1.000	1.000	1.000	1.000	1.000	1.000
4.64e+00	4.64e-03	rbf	1.000	1.000	1.000	1.000	1.000	1.000
1.00e+03	2.15e-04	rbf	1.000	1.000	1.000	1.000	1.000	1.000
4.64e+00	1.00e-05	linear	1.000	1.000	1.000	1.000	1.000	1.000
4.64e+00	2.15e-04	linear	1.000	1.000	1.000	1.000	1.000	1.000
4.64e+00	4.64e-03	linear	1.000	1.000	1.000	1.000	1.000	1.000
4.64e+00	1.00e-01	linear	1.000	1.000	1.000	1.000	1.000	1.000
1.00e+03	1.00e-05	linear	1.000	1.000	1.000	1.000	1.000	1.000
1.00e+03	2.15e-04	linear	1.000	1.000	1.000	1.000	1.000	1.000
1.00e+03	4.64e-03	linear	1.000	1.000	1.000	1.000	1.000	1.000
1.00e+03	1.00e-01	linear	1.000	1.000	1.000	1.000	1.000	1.000
2.15e-02	1.00e-01	rbf	1.000	1.000	1.000	1.000	1.000	1.000
1.00e+03	1.00e-05	rbf	1.000	1.000	1.000	1.000	1.000	1.000

The common values for C and γ between tables 7.14, 7.15 and 7.16 with higher average score, correspond to the cases with C value as $1.00e+03$ and to γ value as $1.00e-05$. Considering the similar results attained for both of the tested kernels (*rbf* and *linear*) and looking into the average training times, it was decided to proceed with a *linear* option for the remaining tests.

7.3.2 CIC-IDS2017 (*DoS* Attack)

Table 7.17: Performance of SVM classifier with the cic_ids2017_DoS_* dataset

Best performance							
Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16616 10] [18 21727]	79	NONE
Worst performance							
Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
0 1	1.00 0.96	0.94 1.00	0.97 0.98	0.98	[15682 944] [8 21737]	10	NONE

The implemented SVM presented a good performance throughout the different experiments for the mentioned attack. Table 7.17 presents the good and worst performance classifications achieved for the cic_ids2017_DoS_a_i and for the cic_ids2017_DoS_e_iii, respectively. Despite this, most be noted that the tests for several other datasets also presented results equal to the best case.

Furthermore, it is possible to see that the biggest lost of performance is registered for the precision regarding the benignum class that drops by 0.06, associated to a considerable amount of false positives, corresponding to 2.46% of the total amount of samples in this case.

Impact of the amount of features

Considering the set of results attained for this attack it is possible to notice a slightly decrease of performance between the tests conducted with the dataset containing the seventy-nine features and the one containing ten features. Despite this, such difference is not relevant enough to sustain the argument that there is indeed an impact due to the amount of used features.

Impact of the dataset balancing strategy

As stated before, due to the reduced difference between the attained results, it isn't possible to state that there is an impact caused by the balancing strategy.

7.3.3 CIC-IDS2017 (*PS* Attack)

Table 7.18: Performance of SVM with the cic_ids2017_PS_* dataset

Best performance							
Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21583 64] [23 26966]	30	Random Over-sampling
Worst performance							
Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
0 1	1.00 0.99	0.98 1.00	0.99 0.99	0.99	[21316 332] [45 26943]	10	NONE

Table 7.18 shows the results attained by the SVM classifier for the port scan attack of the CIC-IDS2017 dataset. The best performance illustrates the results attained for the majority of the conducted experiments with the algorithm presenting maximum values for the considered performance metrics.

The lowest performance was achieved for the cic_ids2017_PS_e_iii dataset, with the algorithm presenting a considerable amount of false positives and false negatives. Despite this, the biggest drop in terms of performance metrics is of 0.02, for the precision of the benignum classes.

Impact of the amount of features

The conducted tests resulted into maximum metric values, for the majority of the cases. As such, it is possible to state that the variance of the amount of features didn't produce any impact on the performance of the algorithm.

Impact of the dataset balancing strategy

As mentioned before, due to the performance of the algorithm during the respective tests, the dataset balancing strategy seems to have had no impact on the

classification task by the SVM.

7.3.4 Custom Dataset (*DoS* Attack)

Table 7.19: Performance of *SVM* with the DoS_* dataset

Best performance							
Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[3045 0] [0 25668]	30	Random Over-sampling
Worst performance							
Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2978 67] [25668 0]	15	SMOTE

Table 7.19 shows the results attained by the SVM classifier for the *DoS* attack of the custom dataset. The best performance was achieved for the DoS_b_i dataset, while the worst was achieved for the DoS_d_ii dataset.

Contrary to the results attained with the cic_ids2017_* datasets, in this case, the implemented approach exhibited a quite different behaviour, which is well represented into the worst case performance with the algorithm not being able to correctly classify a single malignum sample.

Impact of the amount of features

Analysing into detail the results achieved for the DoS_{a,b,c,d,e} it is possible to notice a decrease of performance starting with the DoS_c dataset, which continues up to the DoS_e dataset. The worst performances for each of these datasets are equal to the worst performance present in table 7.19.

The fact that for the datasets DoS_{a,b} such lowest performances are not verified, that points to an actual impact due to the number of used features.

Impact of the dataset balancing strategy

Focusing on the results achieved for the DoS_{c,d,e} datasets, there are some evidences of the impact caused by the dataset balancing strategy, illustrated on table 7.20.

Table 7.20: SVM classification: Impact of amount of features with DoS_* datasets

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	Dataset
2	0 1	0.88 1.00	1.00 0.98	0.93 0.99	0.98	[3033 12] [423 25245]	DoS_c_i
2	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.11	[3021 24] [25668 0]	DoS_c_ii
4	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.11	[3018 28] [25667 0]	DoS_d_i
4	0 1	1.00 0.89	0.01 1.00	0.02 0.94	0.89	[30 3016] [0 25667]	DoS_d_ii

Considering that the two results presented, were achieved for the same *k-fold*, respectively, it is possible to state that the used balancing strategy did presented an impact on the overall performance of the algorithm.

7.3.5 Custom Dataset (*PS* Attack)

Table 7.21: Performance of *Random Classifier* classifier with the cic_ids2017_PS_a_i dataset

Best performance							
Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1408 3] [290 17008]	30	SMOTE
Worst performance							
Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
0 1	0.78 1.00	1.00 0.98	0.87 0.99	0.98	[1407 4] [405 16893]	15	SMOTE

Table 7.21 shows the results attained by the SVM classifier for the port scan attack of the custom dataset. The difference between the best and worst performance, respectively achieved for the PS_b_i and PS_d_ii, presents a maximum difference of 0.05 for the precision regarding the benignum class, associated to the presence of 405 malignum samples incorrectly classified as benignum.

Impact of the amount of features

The attained results by the SVM for the attack in question revealed performance results with minor differences, thus, it's not possible to infer any impact regarding the number of used features.

Impact of the dataset balancing strategy

Having in mind the results attained by the SVM for the attack in hands, its not possible to state that there was an impact caused by the dataset balancing strategy, since no significant performance differences were found.

7.4 CNN

This section aims to present the results achieved with the CNN implementation, when trained and evaluated with the CIC-IDS2017 and with the custom datasets.

7.4.1 CIC-IDS2017 (*DoS* Attack)

Table 7.22 presents the best and worst performances by the *CNN* for the cic-ids2017_DoS_* dataset(s).

Table 7.22: Performance of *CNN* with the cic-ids2017_DoS_*

Best performance								
K	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
3	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16616 10] [29 21716]	30	SMOTE
Worst performance								
K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
1	0 1	1.00 0.97	0.96 1.00	0.98 0.98	0.98	[15984 643] [20 21724]	20	NA

Overall, the CNN presented great performances for the considered attack. The best performance highlighted in table 7.22, attained with the cic-ids2017_DoS_b_ii dataset, well represent the majority of the achieved performances with the remaining experiments for this attack. The metric values achieved for the worst case, with the cic-ids2017_DoS_c_iii dataset, only occurred once.

Must still be noted that, even for the best (and majority of cases), the algorithm isn't able to achieve a zero percent for false positives and false negatives, which doesn't correspond to the ideal scenario.

Impact of the amount of features

Considering the comprehensive set of results for the mentioned attack and respective algorithm, it is clear that this algorithm presents a very good performance for all the different used datasets. As such, and even if there are slight changes in the registered metric values, usually in the order of 0.01, it is possible to state that the Performance of the algorithm isn't influenced by the reduction of the amount of features, at least, for the considered datasets.

Impact of the dataset balancing strategy

Regarding the impact of the used dataset balancing strategy, and as mentioned before, the attained performance for this algorithm doesn't allow to draw any conclusions about its impact. On other hand, it is worth mentioning that the worst performance, previously presented in table 7.22, for the cic-ids2017_DoS_c_iii dataset, was registered for one of the *k-folds* when no balancing strategy was used,

while for the other two balancing methods this lowest performance didn't happen.

Nevertheless, considering it was an isolated situation and the registered decrease of performance wasn't quite significant, it isn't possible to claim that there is an impact caused by the balancing strategy.

7.4.2 CIC-IDS2017 (*PS* Attack)

Table 7.23 presents the best and worst performances by the *CNN* for the cic-ids2017_PS_* dataset(s).

Table 7.23: Performance of *CNN* with the cic-ids2017_PS_*

Best performance								
K	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
1	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21647 1] [19 26970]	30	SMOTE
Worst performance								
K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
3	0 1	1.00 0.99	0.99 1.00	0.99 0.99	0.99	[21347 300] [10 26979]	20	NA

The used CNN presented excellent classifications and an example of it is the minor difference between the best and worst performance illustrated in table 7.23, both registered for the cic-ids2017_PS_a_i dataset. Despite this and similar to what happened for the *cic-ids2017_DoS_** datasets, the algorithm wasn't able to reach a zero false positive and/or false negative rate.

Impact of the amount of features

The conducted tests resulted into maximum metric values, for the majority of the cases. As such, it is possible to state that the variance of the amount of features didn't produce any impact on the Performance of the neural network.

Impact of the dataset balancing strategy

As mentioned before, the conducted tests resulted into maximum metric values, for the majority of the cases. As such, it is possible to state that the used dataset balancing strategy didn't produce any impact on the Performance of the algorithm.

7.4.3 Custom Dataset (*DoS* Attack)

The CNN presented maximum metric values for all the conducted tests with set of DoS_* dataset(s). Table 7.24 only presents an example of the obtained results, since there isn't a "worst case scenario".

Table 7.24: Performance of *CNN* with the DoS_*

K	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
1	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[3045 0] [0 25668]	20	SMOTE

Despite of the majority of the results being similar to the example presented in table 7.24, there were some cases where the algorithm wrongly classified some samples, being the number of false positives and false negatives different from zero. Despite this, these values were not sufficient to change the (rounded) values of the considered metrics.

Impact of the amount of features

Since maximum metric values were achieved throughout all the conducted tests, it is possible to claim that there is no evidence of any impact on the algorithms performance due to the number of features present in each of the considered datasets.

Impact of the dataset balancing strategy

As mentioned before, due to the Performance of the algorithm during the respective tests, the dataset balancing strategy seems to hasn't caused any impact on the classification of the CNN.

7.4.4 Custom Dataset (*PS* Attack)

Table 7.25 presents the best and worst performances by the *CNN* for the PS_* dataset(s).

Table 7.25: Performance of *CNN* with the PS_*

Best performance								
K	Class	Precision	Recall	F1-Score	Accuracy	C. Matrix	N_Features	Balancing
4	0 1	0.97 1.00	0.98 1.00	0.97 1.00	1.00	[1387 23] [49 17249]	79	Random Oversampling
Worst performance								
K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	N_Features	Balancing
3	0 1	0.85 1.00	1.00 0.99	0.92 0.99	0.99	[1411 0] [254 17044]	20	SMOTE

The best performance was achieved for the PS_a_i dataset, with the algorithm reaching 100% of *accuracy* and 97% and 100% of *f1-score*, respectively for the malignum and benignum classes, whereas the worst performance was attained for the PC_c_ii dataset, with the algorithm wrongly classifying 254 samples as false negatives.

Unlike for other datasets, where the majority of the results attained by the CNN were all close or equal to the best performance case, for this sub-set of datasets, the classifications are dispersed between the worst and best case scenario.

Impact of the amount of features

Considering the attained results, it isn't possible to evidence any impact due to the variance of the amount of features. It is possible to observe a range of metric values throughout all of the experiments, either for datasets with different and equal amount of features.

Impact of the dataset balancing strategy

As stated before, it is possible to observe a range of metric values throughout all of the experiments, which also applies to the ones (for the some amount of features) resulting from the application of different balancing strategies. As such, it isn't possible to verify any impact on the Performance of the algorithm, brought by the used dataset balancing strategy.

7.5 Discussion

The first topic of this section aims to resume and discuss the attained results. The second one, aims to present a critical overview of the existent detection logic within the HSPF, while the third one focus on the integration of the implemented approaches that presented good performance results and, as such, evolve to this phase.

7.5.1 Experimentation Results

Random Forest was the algorithm that presented better and consistent results throughout all of the different conducted experiments, presenting maximum values for the considered performance metrics, into the majority of the tests. For the PS_* datasets it was possible to visualize a slightly decrease on the algorithms performance, likely caused by the number of used features, as well, as due to the used balancing strategy during the experimentation phase.

k-means presented a poor overall performance. As example, it was only able to achieve 0.77 and 0.75 of *f1-score* measure, for its best performance, in contrast to 0.45 and 0.00 for the case of its worst performance (where no malignum sample was correctly classified), both for the benignum and malignum classes, respectively.

For some datasets, the impact of the number of used features was verified, as well as, the impact caused by the used balancing strategy. Furthermore, it was also possible to verify that the *Seed* had a real impact over the algorithms performance.

Due to the unsupervised nature of this algorithm, there was some doubts around its eventual performance, but, it wasn't expected such poor one. Further investigation must be conducted with unsupervised approaches to discover functional approaches of this type, specially considering its application potential, which is abroad in section 7.5.2.

For the SVM implementation, an initial parameters search grid was conducted to find a good common combination of values for C, γ and kernel function. Such process was by far the longest in terms of computational time, but, culminated into the decision of using a C value of 1.00+e3 and using "linear" as kernel function. After the average of classification (of the five k-folds), the average time of execution was the second factor taken into account, which, led to the selection of the previously mentioned values to be used as input parameters.

SVM presented excellent performance metric values (in some cases, even maximum values) for the cic-ids2017_* datasets, where (for these set of datasets) its worst performance was translated into 2.46% of the total dataset samples being miss classified as malignum samples, which resulted into *f1-score* values of 0.97 and 0.98, for the benignum and malignum classes, respectively.

On other hand, during the experiments conducted with set of custom datasets, the performance of the SVM based approach didn't presented such good performances, achieving for the worst classification with the DoS_* datasets values of 0.19 and 0.00 of f1-score, for the benignum and malignum classes, respectively. Regarding the test with the PS_* datasets, the attained performances were not so dispar, being the worst classification represented by the following values for f1-score 0.87 and 0.99, also corresponding to the beningnum and malignum classes, respectively.

The overall quality of the performances achieved by the CNN were similar to the ones attained by the Random Forest. The worst classification of this implementation corresponded to 643 false positives and 20 false negatives, into a universe of 38371 samples, for the DoS attack belonging to the CIC-IDS2017 dataset.

In addition, for the majority of the conducted tests it wasn't possible to verify any impact due to the number of features present into the dataset, neither due to the used balancing strategy during the training phase.

Considering the achieved results by each of the implemented approaches, it was decided to proceed with the Random Forest and with the CNN implementation towards the Integration phase, abroad in section 7.5.3.

7.5.2 Detection Logic

Aiming to improve the existent behaviour of the AICO component and considering the current logic previously presented in section 4.7.6, this section brings some possible alternatives to the existent solution, while also presents some considerations over the lifetime of the used algorithm(s).

Classification Lifecycle

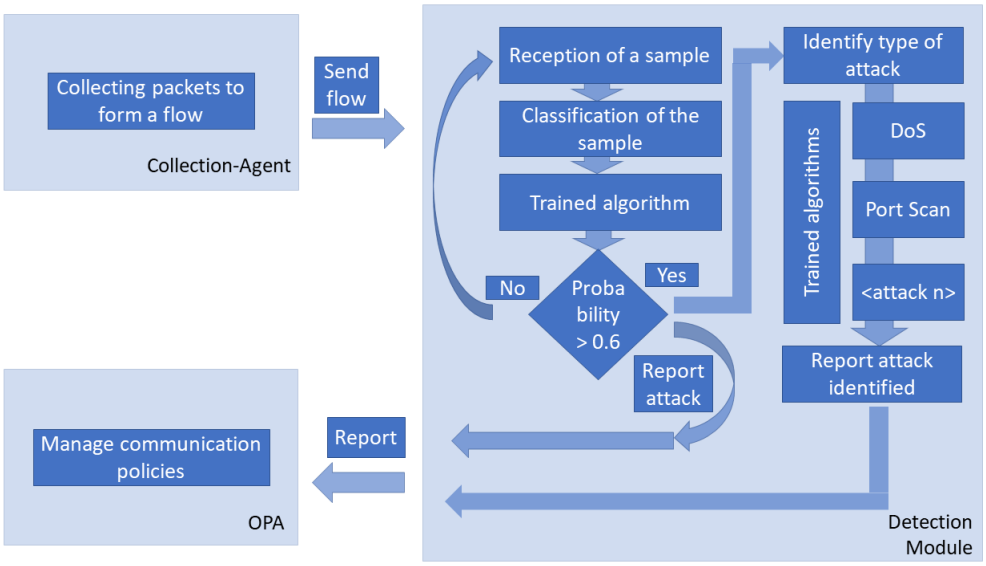


Figure 7.1: HSPF detection logic - Alternative 1

Figure 7.1 illustrates one possible alternative. The major change would be on how it's decided if the sample corresponds to a malignum class or not. In the existent architecture, the sample is fed to all the trained algorithms present in the system and if any of those classifies the sample as malignum, then, the sample is labeled as malignum and OPA is informed.

Such process presents a potential bottleneck related with the amount of time needed for all the classifiers to classify such sample. Having this in mind, the first alternative would be to have only one algorithm providing a first classification and afterwards, in case the sample was classified as malignum, then it would be fed to another set of algorithms that would try to identify the type of attack. Some considerations over this "one algorithm" able to recognize threats of different types are presented in the bottom of this section.

Naturally, after the first positive classification as being a malignum sample, that would be reported to the OPA and proper policies would be applied.

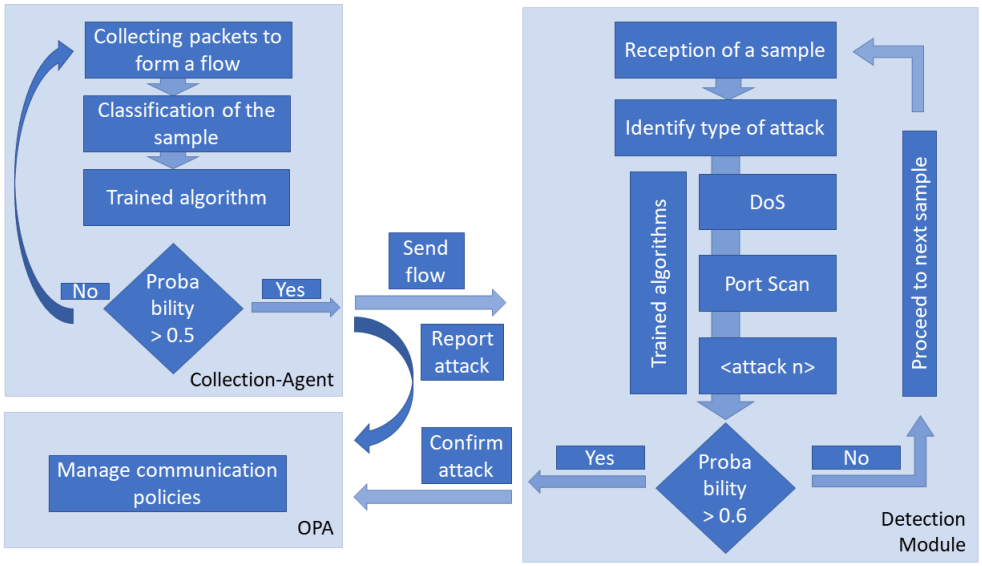


Figure 7.2: HSPF detection logic - Alternative 2

Figure 7.2 represents another possible approach. Instead of being continuously forming the flows and sending them to another component, one way to reduce such over-head would be to incorporate a preliminary classification directly into each collection-agent. Similar to the "one algorithm" previously mentioned, here the idea would be to have a classifier capable to identify a set of threats as malignum, report the identified sample immediately to the OPA and afterwards send it to the Detection Module where further validation would be conducted.

The biggest advantage behind this approach would be to reduce the amount of traffic sent to the Detection Module, thus saving a considerable amount of resources, either the ones associated with the sending process, either the ones associated to having a set of algorithms classifying benignum samples.

Algorithms training

Currently, the algorithms are being externally trained, recurring to the collected custom datasets, and the ones who present good performance are later on included into the Detection Module. Such type of approach is unbearable with the nature of vertical applications, deployed using a Kubernetes (or any similar tool) due to the quickness that is expected into the instant of deployment up to the moment where they are fully functional. Furthermore, considering that it is intended that the HSPF becomes agnostic to any application, thus being able to provide security to any k8s based application, it won't be possible to keep training the algorithms previously in an external environment and then incorporating them into the HSPF detection module.

Having such problem in mind, there are several constraints that should be taken into account:

- It would be necessary to collect benignum data during a specific interval

of time followed by the training of the algorithm(s). Such interval would depend on the amount of traffic shared within the application, but, a significant amount of data should be collected to assure the proper training of the algorithm - e.g., over 50000 samples.

- Malicious data may be provided to be used during the training of the algorithms (retrieved from the reference dataset CIC-IDS2017 [19], as well, from the collected custom datasets)

- Techniques similar to the one presented at [22], where the authors claim that they only used benignum data while training their approach, which would likely reduce the training time and allow to have a more tailored algorithm to the traffic characteristics of a certain micro-service of an application.

- Focusing on providing speed, the system could be prepared to deliver a first version of the needed (trained) algorithms based on a diminished amount of data, while continuing on training another versions with more and recent data. This training process could be something periodic, which would allow the application to adjust their behaviour in functioning of the most recent traffic characteristics.

Further experiments would have to be conducted to validate the described constraints. Despite this, the demanding for a reliable, fast and accurate HSPF is present into the 5G-EPICENTRE [3] project, thus, (hopefully) such considerations will be taken into account during future developments of this framework.

7.5.3 Integration

Parallel to the execution of this internship, there were others occurring in One-Source, one of which resulted into a preliminary version of the HSPF dashboard, illustrated in image 7.3.

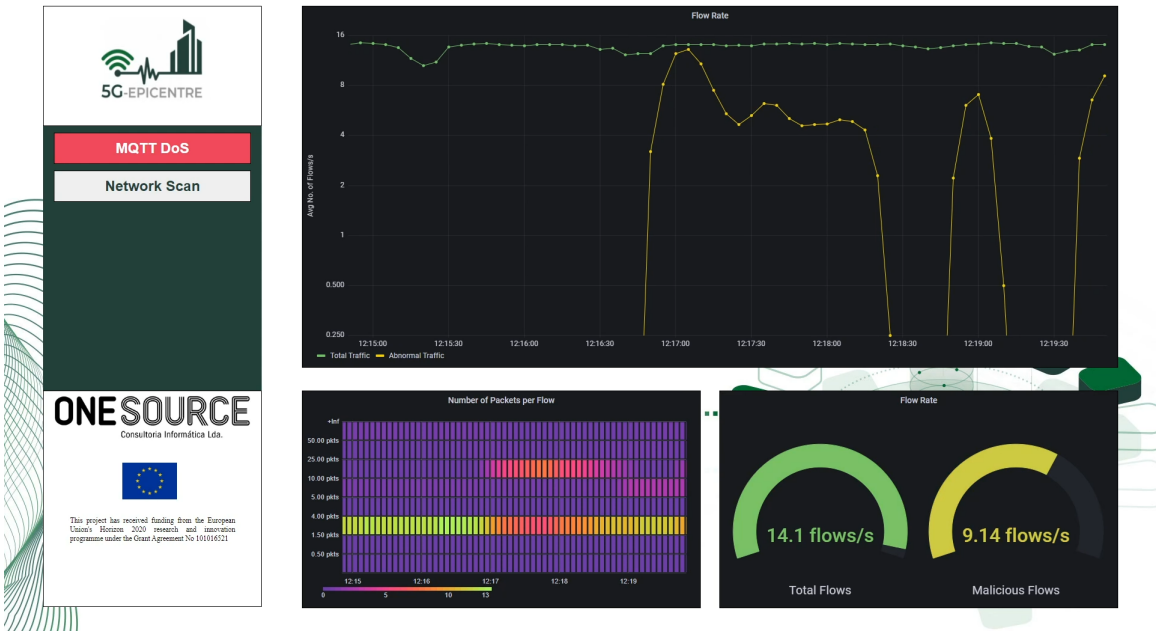


Figure 7.3: Approaches Implementation - Overview

The image illustrates the behaviour of the framework, already enabled with a detection module powered by artificial intelligence. At this time, the algorithms used by the detection module were two, both *Random Forest* implementations, each trained to detect *DoS* and *PS* attacks, as a (one of the) results of this work.

In this dashboard the abnormal flows (e.g., malicious flows) are represented in yellow in the upper graphic, while there is an indication on the left of the type of attack that is currently being simulated. Since this dashboard wasn't yet in a stable version, this was tested in a setup somehow similar to the one used to collect the datasets used during the experimentation phase. There was a component functioning in loop in charge of generating traffic for both of the attacks. Must be noted that this script was not generating traffic in the same manner that the one used for the collection of the datasets, since the most of the parameters were slightly adjusted in order to explore the generalization capability of the trained algorithms.

As the figure demonstrates, it was possible to confirm that the trained algorithms correctly detected both of the attacks, corresponding the longest curves to a *DoS* attack and the shortest one to a *Port Scan* attack.

Chapter 8

Conclusion

Cloud-native technologies have been emerging over the years. Indeed, with the recent introduction of 5G, next-generation applications fully exploring the network and cloud services have been generating considerable amounts of network traffic. Moreover, the flourishing of this type of applications might turn into a Big Data problem, forcing network security systems to adapt to this new reality. One of the solutions for Cloud-Native security systems is often associated with the introduction of intelligent and automated processes. The introduction of AI is envisioned as a promising approach to handle such a high amount of network data and for instance, detect abnormal behaviours.

This document reflects the work of designing, developing and implementing AI based approaches to be included into the detection module of the AI-driven framework for network anomaly detection, being developed by OneSource. The aim of this framework is to detect and mitigate traffic anomalies, specifically under Cloud-Native environments. This work is conducted under the 5G-EPICENTRE European project, where OneSource leads a task focused on investigating various security aspects concerning the 5G-EPICENTRE architecture.

As a starting point, there was an effort to collect some important background knowledge needed for the development of this work. As part of this, the machine learning tribes have been identified, some important concepts into the preprocessing of datasets have been explored, an overall taxonomy of AI anomaly detection techniques have been overviewed and the set of performance metrics to be used during the training, testing and validation phases, were listed.

Additionally, an initial survey covering concepts such as microservice orchestration, network security and AI techniques for network anomaly detection has been conducted. Then, based on the surveyed literature, a selection of possible AI candidates techniques was made. Such a selection included one supervised techniques, one unsupervised and one DL technique focused on SVMs, *k-means* and CNN models, respectively. This selection was elicited due to the lack of consensus between the best reviewed approaches. Indeed, most of the surveyed literature presents good performance and results. Nevertheless, different techniques are assessed using different datasets, leading to a scenario where comparing them becomes difficult, complex and inaccurate.

Afterwards, several steps were performed, namely, the definition of use cases (used to validate the framework), the specification of requirements (that the framework will have to comply with) and a description of the framework itself. The list of the use cases and attacks were discussed considering Mobitrust, a Cloud-Native application. Mobitrust, a situational awareness PPDR application, was chosen as a reference platform to be used to validate the approach that was developed.

The next step was the definition of the methodology, namely the design, implementation and testing and validation phases. Must be noted that the followed methodology was based on an horizontal concept, that is, the implementation of the algorithms was conducted in a sequential way, in a sense that only after the first implementation being completed, the second one was started, ..., following this process for all of the implemented approaches.

The implementation chapter describes into detail the major phases of the process, presenting the several steps taken while pre-processing the datasets and some considerations over the implementation of the candidate approaches, namely the cycle that all the implementations have been submitted to.

Afterwards, the results achieved by the implemented approaches are properly presented. The Random Forest and the CNN based implementations were the ones that presented better results and evolved to the Integration phase. Regarding the implementations of the *k-means* and of the *SVM*, the first presented an overall poor performance, while the second presented good results for the conducted tests with the set of datasets part of the CIC-IDS2017, but, not so good performances for the set of collected datasets. Additionally, it was presented an analysis of the existent detection logic of the HSPF, followed by the exhibition of the preliminary dashboard of the HSPF, where it is presented a continuous cycle of attacks and respective detection using one of the implemented approaches.

Bearing this in mind and returning to the initial work objectives, it is possible to claim that all objectives have been fulfilled. The review of the existing literature is portrayed into chapter 3, completed with the information regarding the background knowledge presented in chapter 2. The identification of the set of features to be present into the collected dataset was achieved during the review of the existent literature, upon the decision of the same set of features that are present into the CIC-IDS2017 dataset.

The techniques applied during the pre-processing of the collected dataset are described into chapter 6, as well as the description of the implementation process. The results of the training, validation and testing are present into chapter 7, with the integration results also being presented into this chapter. In addition, two alternative approaches are suggested aiming to improve the existent detection logic of the HSPF, which complements the analysis of the existent behaviour, previously presented into section 4.7.6.

As future work, it would be interesting to pursuit the exploration of more unsupervised approaches, to explore approaches evolving adversarial ML (as a startup point, at [112] it is conducted an exhaustive review of the State of Art (SoA)) and also to find ways to further validate the implemented approaches that evolved

towards the Integration phase and that are currently being used by the detection module of the HSPF.

References

[1] Onesource. https://onesource.pt/. Accessed: 2022-08-25.

[2] Public protection and disaster relief – transformation center. https://cordis.europa.eu/project/id/313015. Last access: 2022-08-25.

[3] 5g-epicentre. https://www.5gepicentre.eu/. Accessed: 2022-08-25.

[4] Najmul Hassan, Kok-Lim Alvin Yau, and Celimuge Wu. Edge computing in 5g: A review. *IEEE Access*, 7:127276–127289, 2019.

[5] Pedro Domingos. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Penguin Books, 2017.

[6] Al amri Redhwan, Raja Kumar Murugesan, Mustafa Man, Alaa Fareed Abdulateef, Mohammed A. Al-Sharafi, and Ammar Ahmed Alkahtani. A review of machine learning and deep learning techniques for anomaly detection in iot data. *Applied Sciences*, 11(12):5320, Jun 2021.

[7] Hadeel S. Obaid, Saad Ahmed Dheyab, and Sana Sabah Sabry. The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning. In *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, pages 279–283, 2019.

[8] Afnan M. Alhassan and Wan Mohd Nazmee Wan Zainon. Review of feature selection, dimensionality reduction and classification for chronic disease diagnosis. *IEEE Access*, 9:87310–87317, 2021.

[9] Yinglin Xia. Chapter eleven - correlation and association analyses in microbiome study integrating multiomics in health and disease. In Jun Sun, editor, *The Microbiome in Health and Disease*, volume 171 of *Progress in Molecular Biology and Translational Science*, pages 309–491. Academic Press, 2020.

[10] Farzana Anowar, Samira Sadaoui, and Bassant Selim. Conceptual and empirical comparison of dimensionality reduction algorithms (pca, kpca, lda, mds, svd, lle, isomap, le, ica, t-sne). *Computer Science Review*, 40:100378, 2021.

[11] Wasim Ali, Manasa N, Mohammed Fadhel Aljunid, Malika Bendechache, and P. Sandhya. A review of current machine learning approaches for anomaly detection in network traffic. *Journal of Telecommunications and the Digital Economy*, 8:64–95, 12 2020.

[12] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 270–279, Cham, 2018. Springer International Publishing.

[13] Doaa Ahmed Sayed, Sherine Rady, and Mostafa M. Aref. Enhancing clustream algorithm for clustering big data streaming over sliding window. *2020 12th International Conference on Electrical Engineering (ICEENG)*, pages 108–114, 2020.

[14] sklearn.metrics.r2_score. `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html`. Last access: 2022-08-25.

[15] Tuan Phan Vuong, George Loukas, Diane Gan, and Anatolij Bezemskij. Decision tree-based detection of denial of service and command injection attacks on robotic vehicles. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2015.

[16] Mohanad Sarhan, Siamak Layeghy, Nour Moustafa, and Marius Portmann. Towards a standard feature set of NIDS datasets. *CoRR*, abs/2101.11315, 2021.

[17] Mohamed Amine Ferrag, Leandros Maglaras, Ahmed Ahmim, Makhlouf Derdour, and Helge Janicke. Rdtids: Rules and decision tree-based intrusion detection system for internet-of-things networks. *Future Internet*, 12(3), 2020.

[18] Eibe Frank and Ian Witten. Reduced-error pruning with significance tests. 07 1999.

[19] Intrusion detection evaluation dataset (cic-ids2017). `https://www.unb.ca/cic/datasets/ids-2017.html`. Accessed: 2022-08-25.

[20] Nour Moustafa. The bot-iot dataset. `https://dx.doi.org/10.21227/r7v2-x988`. Accessed: 2022-08-25.

[21] Mehdi Hosseinzadeh, Amir Rahmani, Bay Vo, Moazam Bidaki, Mohammad Masdari, and Mehran Zangakani. Improving security using svm-based anomaly detection: issues and challenges. *Soft Computing*, 25:1–29, 02 2021.

[22] Xuedan Miao, Ying Liu, Haiquan Zhao, and Chunguang Li. Distributed online one-class support vector machine for anomaly detection over networks. *IEEE Transactions on Cybernetics*, 49(4):1475–1488, 2019.

[23] Uci machine learning repository. `http://archive.ics.uci.edu/ml/index.php`. Last access: 2022-08-25.

[24] Saman Maroufpoor, Omid Bozorg-Haddad, and Xuefeng Chu. Chapter 9 - geostatistics: principles and methods. In Pijush Samui, Dieu Tien Bui, Subrata Chakraborty, and Ravinesh C. Deo, editors, *Handbook of Probabilistic Models*, pages 229–242. Butterworth-Heinemann, 2020.

[25] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Debo Cheng. Learning k for knn classification. 8(3), jan 2017.

[26] Mathew X Ma, Henry Y.T Ngan, and Wei Liu. Density-based outlier detection by local outlier factor on largescale traffic data. *Electronic Imaging*, 2016(14):1–4, 2016.

[27] Radhia Fezai, Majdi Mansouri, Okba Taouali, Mohamed Faouzi Harkat, and Nasreddine Bouguila. Online reduced kernel principal component analysis for process monitoring. *Journal of Process Control*, 61:1–11, 2018.

[28] Sutharshan Rajasegarar, Christopher Leckie, and Marimuthu Palaniswami. Hyperspherical cluster based distributed anomaly detection in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 74(1):1833–1847, 2014.

[29] Jie Gu, Lihong Wang, Huiwen Wang, and Shanshan Wang. A novel approach to intrusion detection using svm ensemble with feature augmentation. *Computers Security*, 86:53–62, 2019.

[30] Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8), 2020.

[31] Zhuo Wang, Yanghui Zhou, and Gangmin Li. Anomaly detection by using streaming k-means and batch k-means. In *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, pages 11–17, 2020.

[32] Rasim Alguliyev, Ramiz Aliguliyev, and Fargana Abdullayeva. Pso+k-means algorithm for anomaly detection in big data. *Statistics, Optimization Information Computing*, 7, 05 2019.

[33] J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.

[34] Meshal Shutaywi and Nezamoddin N. Kachouie. Silhouette analysis for performance evaluation in machine learning with applications to clustering. *Entropy*, 23(6), 2021.

[35] Evaluation of clustering. `https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html`. Accessed: 2022-08-25.

[36] S5 - a labeled anomaly detection dataset, version 1.0(16m). `https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70`. Accessed: 2022-08-25.

[37] Bhavya Mor, Sunita Garhwal, and Ajay Loura. A systematic review of hidden markov models and their applications. *Archives of Computational Methods in Engineering*, 28, 05 2020.

[38] Wondimu K. Zegeye, Richard A. Dean, and Farzad Moazzami. Multi-layer hidden markov model based intrusion detection system. *Machine Learning and Knowledge Extraction*, 1(1):265–286, 2019.

[39] Frances Y. Kuo and Ian H. Sloan. Lifting the curse of dimensionality. 2005.

[40] Patrick Lee, Tian Bu, and Thomas Woo. On the detection of signaling dos attacks on 3g/wimax wireless networks. *Computer Networks*, 53:2601–2616, 10 2009.

[41] June ho Bang, Young-Jong Cho, and Kyungran Kang. Anomaly detection of network-initiated lte signaling traffic in wireless sensor and actuator networks based on a hidden semi-markov model. *Computers Security*, 65:108–120, 2017.

[42] Kun Zhou, Wenyong Wang, Teng Hu, and Kai Deng. Application of improved asynchronous advantage actor critic reinforcement learning model on anomaly detection. *Entropy*, 23(3), 2021.

[43] Ying-Feng Hsu and Morito Matsuoka. A deep reinforcement learning approach for anomaly network intrusion detection system. In *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*, pages 1–6, 2020.

[44] Guillermo Caminero, Manuel Lopez-Martin, and Belen Carro. Adversarial environment reinforcement learning algorithm for intrusion detection. *Computer Networks*, 159:96–109, 2019.

[45] Mengmeng Ge, Xiping Fu, Naeem Syed, Zubair Baig, Gideon Teo, and Antonio Robles-Kelly. Deep learning-based intrusion detection for iot networks. In *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 256–25609, 2019.

[46] Benjamin Lindemann, Benjamin Maschler, Nada Sahlab, and Michael Weyrich. A survey on anomaly detection for technical systems using lstm networks. *Computers in Industry*, 131:103498, 2021.

[47] V. Kanimozhi and T. Prem Jacob. Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset cse-cic-ids2018 using cloud computing. In *2019 International Conference on Communication and Signal Processing (ICCSP)*, pages 0033–0036, 2019.

[48] Imtiaz Ullah and Qusay H. Mahmoud. Design and development of a deep learning-based model for anomaly detection in iot networks. *IEEE Access*, 9:103906–103926, 2021.

[49] Ming-Yang Su. Real-time anomaly detection systems for denial-of-service attacks by weighted k-nearest-neighbor classifiers. *Expert Syst. Appl.*, 38:3492–3498, 2011.

[50] Mingzhu Tang, Xiangwan Fu, Huawei Wu, Huang Qi, and Qi Zhao. Traffic flow anomaly detection based on robust ridge regression with particle swarm optimization algorithm. *Mathematical Problems in Engineering*, 2020:1–10, 2020.

[51] James Zhang, Ilija Vukotic, and Robert Gardner. Anomaly detection in wide area network mesh using two machine learning anomaly detection algorithms, 2018.

[52] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *CoRR*, abs/2007.02500, 2020.

[53] Ahmed Ahmim, Leandros Maglaras, Mohamed Amine Ferrag, Makhlouf Derdour, and Helge Janicke. A novel hierarchical intrusion detection system based on decision tree and rules-based models. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 228–233, 2019.

[54] Mehdi Hosseinzadeh, Amir Rahmani, Bay Vo, Moazam Bidaki, Mohammad Masdari, and Mehran Zangakani. Improving security using svm-based anomaly detection: issues and challenges. *Soft Computing*, 25:1–29, 02 2021.

[55] Mobitrust. `https://mobitrust.onesource.pt/`. Accessed: 2022-08-25.

[56] Ramaswamy Chandramouli and Zachary Butcher. Building secure microservices-based applications using service-mesh architecture. 2020.

[57] Kubernetes, production-grade container orchestration. `https://kubernetes.io/`. Accessed: 2022-08-25.

[58] Yaml. `https://yaml.org/`. Accessed: 2022-08-25.

[59] Public protection and disaster relief – transformation center. `https://shorturl.at/CHKV1`. Last access: 2022-08-25.

[60] Istio. `https://istio.io/`. Accessed: 2022-08-25.

[61] Istio - deployment architecture. `https://istio.io/latest/docs/ops/deployment/architecture/`. Last access: 2022-08-25.

[62] Open policy agent (opa). `https://www.openpolicyagent.org`. Accessed: 2022-08-25.

[63] Mohan V. Pawar and J. Anuradha. Network security and types of attacks in network. *Procedia Computer Science*, 48:503–506, 2015. International Conference on Computer, Communication and Convergence (ICCC 2015).

[64] What is spoofing? spoofing definition. `https://www.malwarebytes.com/spoofing`. Last access: 2022-08-25.

[65] Jason Andress. Chapter 1 - what is information security? In Jason Andress, editor, *The Basics of Information Security (Second Edition)*, pages 1–22. Syngress, Boston, second edition edition, 2014.

[66] Raja Datta and Ningrinla Marchang. Chapter 7 - security for mobile ad hoc networks. In Sajal K. Das, Krishna Kant, and Nan Zhang, editors, *Handbook on Securing Cyber-Physical Critical Infrastructure*, pages 147–190. Morgan Kaufmann, Boston, 2012.

[67] Wenliang Luo and Wenzhi Han. Ddos defense strategy in software definition networks. In *2019 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, pages 186–190, 2019.

[68] Kuldeep Sharma, Neha Khandelwal, and Prabhakar M. An overview of security problems in manet.

[69] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. Sybil attacks and their defenses in the internet of things. *IEEE Internet of Things Journal*, 1(5):372–383, 2014.

[70] Jin Teng, Wenjun Gu, and Dong Xuan. Chapter 10 - defending against physical attacks in wireless sensor networks. In Sajal K. Das, Krishna Kant, and Nan Zhang, editors, *Handbook on Securing Cyber-Physical Critical Infrastructure*, pages 251–279. Morgan Kaufmann, Boston, 2012.

[71] Madhukar Anand, Zachary Ives, and Insup Lee. papers/176 quantifying eavesdropping vulnerability in sensor networks. pages 3–9, 01 2005.

[72] Network security countermeasures and solutions. `https://shorturl.at/vy047`. Last access: 2022-08-25.

[73] Felix Erlacher and Falko Dressler. Fixids: A high-speed signature-based flow intrusion detection system. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–8, 2018.

[74] Felix Erlacher and Falko Dressler. Fixids: A high-speed signature-based flow intrusion detection system. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–8, 2018.

[75] Philokypros Ioulianou, Vassilios Vassilakis, and Ioannis Moscholios. A signature-based intrusion detection system for the internet of things. 07 2018.

[76] Vermont. `https://github.com/felixe/ccsVermont`. Last access: 2022-08-25.

[77] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. Cross-level sensor network simulation with cooja. *Local Computer Networks, Annual IEEE Conference on*, 0:641–648, 11 2006.

[78] M. Patel, J. Shangkuan, and C. Thomas. What's new with the internet of things? `https://www.mckinsey.com/industries/semiconductors/our-insights/whats-new-with-the-internet-of-things`. Accessed: 2022-08-25.

[79] 12 benefits of cloud computing. https://www.salesforce.com/products/platform/best-practices/benefits-of-cloud-computing/. Accessed: 2022-08-25.

[80] Spyridon Samonas and David Lewis Coss. The cia strikes back: Redefining confidentiality, integrity and availability in security. 2014.

[81] The role of artificial intelligence in cybersecurity. https://www.boozallen.com/s/insight/publication/role-of-artificial-intelligence-in-cyber-security.html. Accessed: 2022-08-25.

[82] Torsten George. The role of artificial intelligence in cyber security. *SecurityWeek*, 2021.

[83] 3 ways to use ai for network security. https://www.toolbox.com/it-security/network-security/articles/3-ways-to-use-ai-for-network-security/. Last access: 2022-08-25.

[84] Applications of ai in cybersecurity. https://shorturl.at/ghs47. Last access: 2022-08-25.

[85] Joao Henriques, Luis Rosa, Andre Gomes, Luis Cordeiro, Konstantinos C. Apostolakis, George Margetis, Constantine Stephanidis, Maria-Andrea R. Anastasi, Christos Skoufis, Apostolos Siokis, and Kostas Ramantas. The 5g-epicentre approach for decreasing attack surface on cross-testbeds cloud-native 5g scenarios. In *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 7–12, 2021.

[86] Cve details - cve-2021-33175. https://www.cvedetails.com/cve/CVE-2021-33175/. Last access: 2022-08-25.

[87] Cve details - cve-2021-32027. https://www.cvedetails.com/vulnerability-list/vendor_id-336/product_id-575/year-2021/opov-1/Postgresql-Postgresql.html. Last access: 2022-08-25.

[88] Cve details - cve-2019-20933. https://www.cvedetails.com/cve/CVE-2019-20933/. Last access: 2022-08-25.

[89] Cve details - cve-2020-10739. https://www.cvedetails.com/cve/CVE-2020-10739/. Last access: 2022-08-25.

[90] Cve details - cve-2019-11248. https://www.cvedetails.com/cve/CVE-2019-11248/. Last access: 2022-08-25.

[91] Cve details - cve-2022-21667. https://www.cvedetails.com/cve/CVE-2022-21667/. Last access: 2022-08-25.

[92] Tcpdump(1) man page. https://www.tcpdump.org/manpages/tcpdump.1.html. Last access: 2022-08-25.

[93] Mqtt stresser. https://github.com/inovex/mqtt-stresser. Last access: 2022-08-25.

[94] Nfstream: Flexible network data analysis framework. https://www.nfstream.org/. Last access: 2022-08-25.

[95] Python. https://www.python.org/about/. Accessed: 2022-08-25.

[96] scikit-learn. https://scikit-learn.org/stable/index.html. Accessed: 2022-08-25.

[97] Tensorflow. https://www.tensorflow.org/learn. Accessed: 2022-08-25.

[98] Tzu-Tsung Wong and Po-Yang Yeh. Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1586–1594, 2020.

[99] Ranjit Panigrahi and Samarjeet Borah. A detailed analysis of cicids2017 dataset for designing intrusion detection systems. 7:479–482, 01 2018.

[100] Randomoversampler. https://imbalanced-learn.org/dev/references/generated/imblearn.over_sampling.RandomOverSampler.html. Last access: 2022-08-25.

[101] Smote. http://glemaitre.github.io/imbalanced-learn/generated/imblearn.over_sampling.SMOTE.html. Last access: 2022-08-25.

[102] Kjell Johnson Max Kuhn. *Applied Predictive Modeling*. 2013.

[103] Gareth James. *An Introduction to Statistical Learning*. 2013.

[104] sklearn.ensemble.randomforestclassifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Last access: 2022-08-25.

[105] sklearn.cluster.kmeans. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. Last access: 2022-08-25.

[106] sklearn.svm.svc. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. Last access: 2022-08-25.

[107] The sequential class. https://keras.io/api/models/sequential/. Last access: 2022-08-25.

[108] Conv1d layer. https://keras.io/api/layers/convolution_layers/convolution1d/. Last access: 2022-08-25.

[109] Dense layer. https://keras.io/api/layers/core_layers/dense/. Last access: 2022-08-25.

[110] Flatten layer. https://keras.io/api/layers/reshaping_layers/flatten/. Last access: 2022-08-25.

[111] Maxpooling1d layer. https://keras.io/api/layers/pooling_layers/max_pooling1d/. Last access: 2022-08-25.

[112] Nuno Martins, Jose Cruz, Tiago Cruz, and Pedro Henriques Abreu. Adversarial machine learning applied to intrusion and malware scenarios: A systematic review. *IEEE Access*, PP:1–1, 02 2020.

[113] 1998 darpa intrusion detection evaluation dataset. `https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset`. Accessed: 2022-08-25.

[114] Cse-cic-ids2018 on aws. `https://www.unb.ca/cic/datasets/ids-2018.html`. Accessed: 2022-08-25.

[115] Cristian Patachia-Sultanoiu, Ion Bogdan, George Suciu, Alexandru Vulpe, Oana Badita, and Bogdan Rusti. Advanced 5g architectures for future netapps and verticals. In *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–6, 2021.

[116] Sarah Kneebone, Liam Smith, and Kelly Fielding. The impact-likelihood matrix: A policy tool for behaviour prioritisation. *Environmental Science Policy*, 70:9–20, 2017.

Appendices

Appendix A

Internship Management

This chapter covers several aspects related to the internship management. The planning of the work to be carried out during the two semesters is translated into two lists of tasks and two Gant charts, one per semester.

Considering the initial proposal for this dissertation, an effort has been made aiming to get well described and proper formulated tasks. Therefore, a set of tasks has been defined for the first and second semesters and are stated in the next two sections.

A.1 1^{st} Semester - Planned Vs Executed

This section covers first semester activities related aspects. A Gant chart is presented containing the planned timeline for the tasks and a description of each task is also conducted. The progress of work conducted so far is also present in this section, providing a per task analysis.

Figure A.2 represents the activities timeline for the first semester.

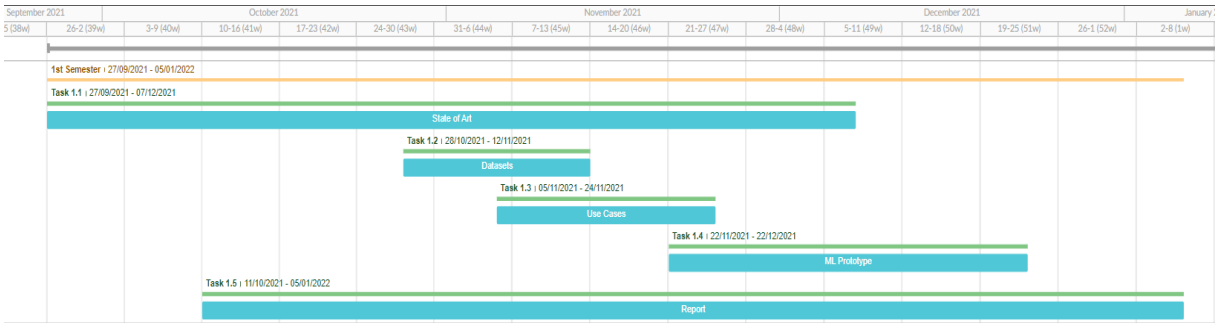


Figure A.1: 1^{st} Semester - Activities Plan

The tasks defined for the first semester are:

T1.1: Analysis of the state of the art and requirements gathering for the ML model

to be developed in order to implement the Security framework.

This task includes investigation activities over the following topics: existing AI models; common threats; privacy policies; security policies; among others. The technologies used to run these AI models will also be a subject of investigation.

T1.2: Analysis of the existing datasets based on network traffic, referenced in the literature (two examples: DARPA 1998 [113] or CSE-CIC-IDS2018 on AWS [114])

Conduct a thorough analysis of datasets mentioned in articles reviewed during T1.1. Some key aspects to consider will be the availability, the scenario in which they were collected and the existence of methods that have already been tested with these datasets, in order to contribute to the validation and evaluation of the model to be implemented.

After an initial selection, the chosen dataset(s) should be deeper analysed, namely by understanding which is the most relevant features.

T1.3: Specification of use cases to demonstrate the applicability of the ML model

This task will have a fundamental contribution to the work since it will reflect the scenario(s) that the model to be developed will be facing. These scenario(s) will raise from the investigation on common threats, conducted in T1.1.

T1.4: Implementation of an ML model prototype for one of the datasets previously analysed in T1.2.

In this section the goal is to implement a first version of the model and validate it with one of the previously reviewed datasets. The implementation process must take into account the use cases previously pointed out, while the dataset must also be properly prepared in order to transmit only the relevant features towards the algorithm.

T1.5: Intermediate report writing

Simultaneously with the tasks described above, the writing of the intermediate report will be a task that will extend itself from the beginning of the semester and will end in December, so that a gap between this internal delivery and the formal presentation exists, allowing for a careful and thoughtful revision by both supervisors.

Progress of Work

Considering the tasks foreseen for the first semester, the following paragraphs provide an overview of the work carried out so far, on a per-task basis.

Task 1.1 was the one with the biggest duration over the first semester. The main activities developed under this task were: (i) the investigation on which technologies (and linked concepts) that are commonly used to implement cloud-native applications that follow service-mesh architectures; (ii) the identification of the

most common attacks, as well, as the traditional methods used in NIDS; (iii) understanding which is the role that AI may have in network intrusion detection systems and which is the taxonomy of AI anomaly-based IDS; (iv) deeper research on concrete AI approaches for network anomaly detection, with a special focus on ML based approaches. In sum, the work carried out under this task is reflected in the majority of the contents described in chapter 3 of this work.

The activities developed under Task 1.2 were conducted simultaneously with the activities of Task 1.1. The identification and brief research on datasets were conducted every time the investigation for new approaches revealed the use of an unseen dataset.

The identification of use cases for the Holistic Security and Privacy Framework, including the ones related with the AI component to be implemented, summits the activities conducted under Task 1.3. Naturally, these will be further updated considering the conclusions attained in the first experiments, planned for the second semester.

The implementation of a ML prototype, respective to Task 1.4 was not conducted. The main reasons for this were related with the need of extra time to better understand concepts (e.g., security-related ones, traditional techniques to detect traffic anomalies) and technologies (e.g., service mesh, Kubernetes, Istio), as well, to understand the existent behaviour of the security framework and how the integration would be conducted with its detection module. Furthermore, the time-frame predicted for this task was also used to define a proposal of approach (presented in chapter 5) for the activities for the second semester, namely: (i) through the definition of the requirements; (ii) definition of the collecting process, for the construction of the realistic dataset; (iii) definition of testing and validation process. With this in mind, the implementation of ML prototypes will be conducted in the time-frame window predicted for Task 2.1, since some of the objectives foreseen for this task have already been satisfied by the current state of the proposed approach (the list of attacks to be performed and the testing and validation scenario).

Besides the writing of this document, as the main activity of Task 1.5, an active contribution was also provided in the writing of two *deliverables*[1] for 5G-EPICENTRE project, *D2.1: Cloud-native security specifications*, that covered a set of cloud-native security aspects, where the addressed Holistic Security and Privacy Framework is presented as a candidate for the different partners to use to protect their NetApps[2] ; and to *D4.1 Integration Plan and Framework Design*, that covered a set of integration-related aspects, namely through the definition of how several components will communicate and also through the definition of integration roadmaps on a consortium member basis.

[1] A deliverable corresponds to a document that reports the knowledge collected or generated, during a period of time, by a set of partners. A deliverable might be elaborated for public or private knowledge, depending on its content

[2] "new and innovative 5G-empowered network applications" [115]

A.2 2^{nd} Semester - RoadMap

Aiming to plan the second semester activities, a Gant Chart has also been elaborated, followed by the respective description of each of the tasks. A risk analysis has been superficially conducted, focusing on the second semester activities.

Figure A.2 represents the activities timeline for the second semester.

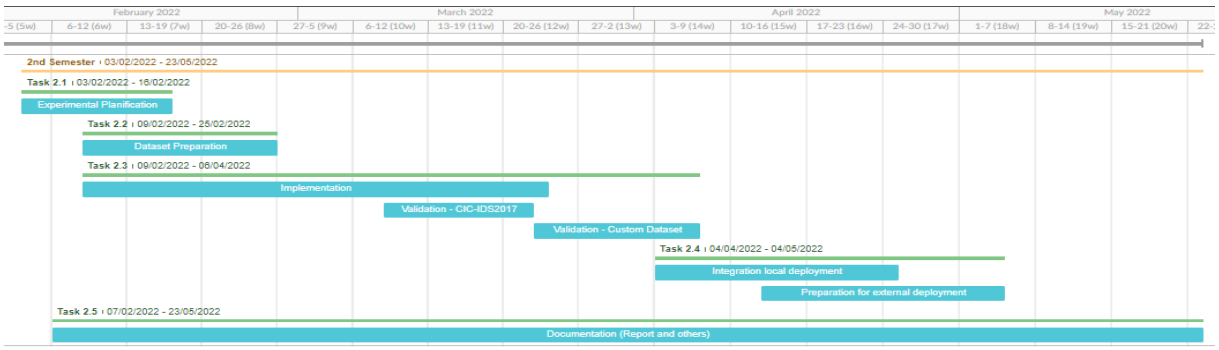


Figure A.2: 2^{nd} Semester - Activities Plan

The tasks defined for the second semester are:

T2.1: Specification of a ML model to be implemented and respective assessment.

During this task, it will be specified: (i) the ML model to be implemented; (ii) the attacks to be performed; (iii) the testing and validation scenario to be used. A document should also be prepared with a brief description of the attacks to be performed, including some information about them: nature, purpose and degree of severity.

This task should result in a set of initial results that will be further analysed in the Discussion phase.

T2.2: Preparation of the dataset to be used during the ML model training, including the definition of relevant features

A dataset should be built based on the network traffic originated between the various components of Mobitrust platform. Next, this dataset should be subject to analysis, resulting in a set of conclusions, such as: what are the most relevant features, if it's necessary to deal with missing data, among others.

T2.3: Validation and evaluation of the ML model.

The ML model should be validated, even before its integration with the other components. During this stage, it will be possible to make small adjustments, if necessary, in order to improve the ML model. The validation will be conducted using two different datasets. This step will produce a set of results that will be discussed in the Discussion of the final report.

T2.4: Integration of the ML model with the Mobitrust platform.

This task targets the major objective of this work, in a sense that, the integration of the model to be developed with the other components of the application, represents the desired outcome. It must be possible to train the ML model with the prepared dataset and to report the results obtained.

The results observed in this task will represent the greatest contribution to the Discussion section since they will be obtained in a considerably realistic environment.

T2.5: Preparation of documentation, including the internship report, technical documents and user manuals.

Simultaneously with the tasks described planned for the second semester, the writing of the final document will be a task that will exist from the beginning of February/2022 and will be finished by the end of May, so that there will be a time-frame between this internal milestone and the formal presentation, in order to allow a careful and thoughtful revision by both supervisors.

Progress of Work

The foreseen tasks for the 2^{nd} semester were all completed, but, there was a delay related to tasks 2.3 and 2.4, which led to a delay into the deliver of this document. Naturally, as part of the implementation phase, several prototypes were implemented, thus, including the activities initially planed for task 1.4. Despite of the delay, the objectives of the work were achieved and four AI approaches were implemented.

Furthermore, two approaches that presented good performance results into the experimentation phase were integrated with the detection module of the HSPF and theirs behaviour was validated through the use of a Graphic User Interface (GUI).

Risk Analysis

Inherit to any development activity there are always risks that must be identified as soon as possible, to allow the definition of mitigation plans. This way, a pro-active approach is followed, allowing for the biggest awareness of possible disruptive factors that may occur, enabling to not only know how to react to them, if they eventually happen but also to develop activities that foresee their occurrence and as such, are conducted in a way to minimise their impact. Figure A.1 presents the risks identified, following a risk analysis approach explained at [116].

Table A.1: List of Risks

ID	Risk Details
1	**Description** **Fact**: Incapacity to work due to temporary disease (e.g., due to covid-19) **Consequence**: Substantial delay on scheduled activities **Likelihood**: 4 **Impact**: 5 **Severity**: 20 **Mitigation Plan**: Plan the several tasks in a way that at least one week will be left as a window time-frame to cover possible delays
2	**Description** **Fact**: Difficulty in accessing appropriate datasets for algorithm validation **Consequence**: Incapacity to evaluate the performance of the implemented algorithm **Likelihood**: 2 **Impact**: 3 **Severity**: 6 **Mitigation Plan**: During the research activities of the first semester, look for several appropriate datasets
3	**Description** **Fact**: Complexity associated with the implementation of AI algorithms **Consequence**: Implementation time may exceed the allocated period **Likelihood**: 3 **Impact**: 3 **Severity**: 9 **Mitigation Plan**: During the research activities, look for public (open-source) libraries that offer AI algorithms implementations
4	**Description** **Fact**: Implemented approach presents poor performance during the validation phase **Consequence**: The approach can not proceed to the integration phase **Likelihood**: 5 **Impact**: 3 **Severity**: 15 **Mitigation Plan**: Select a set of possible candidate solutions and proceed to the implementation of prototypes, no later than the first month of the second semester
5	**Description** **Fact**: IA algorithms need a substantial amount of time to train and test **Consequence**: Training and validation periods may exceed the allocated ones **Likelihood**: 4 **Impact**: 3 **Severity**: 12 **Mitigation Plan**: Address the possibility of having specialized equipment with high processing capacity, next to the hosting company

Continues on next page

Table A.1 – List of Risks (continuance)

ID	Risk Details
6	**Description** **Fact**: Complexity associated with the reproduction of all the use-cases (defined for the security framework) **Consequence**: Incapacity to reproduce all the defined use-cases **Likelihood**: 4 **Impact**: 2 **Severity**: 8 **Mitigation Plan**: Reduce the amount of use-cases, considering only one-case related to each attack type
7	**Description** **Fact**: Complexity associated with the implementation of all the requirements (defined for the security framework) **Consequence**: Incapacity to satisfy all the defined requirements **Likelihood**: 4 **Impact**: 2 **Severity**: 8 **Mitigation Plan**: The development process takes into account the priorities defined for each requirement (implementing first the ones with High priority)

´

A.3 Summary

This chapter covers the basic aspects related to the management of the internship. The description of the tasks predicted for the first and second semester are presented, as well, as *gantt* charts are provided in order to illustrate graphically the order and simultaneity of the mentioned tasks.

A brief risk analysis is conducted and detailed progress of work is provided. Regarding the progress of work, the biggest deviation from the initial plan is related to task 1.4 which envisioned the development of a ML model even during the first semester. This deviation happened due to the development of other activities, namely, related to the planning of the proposed approach to be followed during the second semester. Despite this, the development of the ML prototypes will be conducted in the time frame predicted for Task 2.1, at the beginning of the second semester.

Appendix B

Results

As mentioned in section B.407 the description of the used datasets is the following:

Experimentation Phase

The processed datasets attained from the CIC-IDS2017 are described next:

For the *Dos* attack:

- cic-ids2017_DoS_a_i : considered all the 79 features (using random oversampling)
- cic-ids2017_DoS_a_ii : considered all the 79 features (using SMOTE)
- cic-ids2017_DoS_a_iii : considered all the 79 features (no balancing)
- cic-ids2017_DoS_b_i : considered the 30 more discriminant features (using random oversampling)
- cic-ids2017_DoS_b_ii : considered the 30 more discriminant features (using SMOTE)
- cic-ids2017_DoS_b_iii : considered the 30 more discriminant features (no balancing)
- cic-ids2017_DoS_c_i : considered the 20 more discriminant features (using random oversampling)
- cic-ids2017_DoS_c_ii : considered the 20 more discriminant features (using SMOTE)
- cic-ids2017_DoS_c_iii : considered the 20 more discriminant features (no balancing)
- cic-ids2017_DoS_d_i : considered the 15 more discriminant features (using random oversampling)

- cic-ids2017_DoS_d_ii : considered the 15 more discriminant features (using SMOTE)
- cic-ids2017_DoS_d_iii : considered the 15 more discriminant features (no balancing)
- cic-ids2017_DoS_e_i : considered the 10 more discriminant features (using random oversampling)
- cic-ids2017_DoS_e_ii : considered the 10 more discriminant features (using SMOTE)
- cic-ids2017_DoS_e_iii : considered the 10 more discriminant features (no balancing)

For the *Port Scan* attack:

- cic-ids2017_PS_a_i : considered all the 79 features (using random oversampling)
- cic-ids2017_PS_a_ii : considered all the 79 features (using SMOTE)
- cic-ids2017_PS_a_iii : considered all the 79 features (no balancing)
- cic-ids2017_PS_b_i : considered the 30 more discriminant features (using random oversampling)
- cic-ids2017_PS_b_ii : considered the 30 more discriminant features (using SMOTE)
- cic-ids2017_PS_b_iii : considered the 30 more discriminant features (no balancing)
- cic-ids2017_PS_c_i : considered the 20 more discriminant features (using random oversampling)
- cic-ids2017_PS_c_ii : considered the 20 more discriminant features (using SMOTE)
- cic-ids2017_PS_c_iii : considered the 20 more discriminant features (no balancing)
- cic-ids2017_PS_d_i : considered the 15 more discriminant features (using random oversampling)
- cic-ids2017_PS_d_ii : considered the 15 more discriminant features (using SMOTE)
- cic-ids2017_PS_d_iii : considered the 15 more discriminant features (no balancing)
- cic-ids2017_PS_e_i : considered the 10 more discriminant features (using random oversampling)

- cic-ids2017_PS_e_ii : considered the 10 more discriminant features (using SMOTE)
- cic-ids2017_PS_e_iii : considered the 10 more discriminant features (no balancing)

Regarding the final datasets attained after processing the raw sets collected from realistic communications of the Mobitrust application, while collecting the different attacks, they are summit in the next lists:

The ones generated from the *DoS* attack, were:

- DoS_a_i : considered all the 79 features (using random oversampling)
- DoS_a_ii : considered all the 79 features (using SMOTE)
- DoS_a_iii : considered all the 79 features (no balancing)
- DoS_b_i : considered the 30 more discriminant features (using random oversampling)
- DoS_b_ii : considered the 30 more discriminant features (using SMOTE)
- DoS_b_iii : considered the 30 more discriminant features (no balancing)
- DoS_c_i : considered the 20 more discriminant features (using random oversampling)
- DoS_c_ii : considered the 20 more discriminant features (using SMOTE)
- DoS_c_iii : considered the 20 more discriminant features (no balancing)
- DoS_d_i : considered the 15 more discriminant features (using random oversampling)
- DoS_d_ii : considered the 15 more discriminant features (using SMOTE)
- DoS_d_iii : considered the 15 more discriminant features (no balancing)
- DoS_e_i : considered the 10 more discriminant features (using random oversampling)
- DoS_e_ii : considered the 10 more discriminant features (using SMOTE)
- DoS_e_iii : considered the 10 more discriminant features (no balancing)

The ones generated from the *Port Scan* attack, were:

- PS_a_i : considered all the 79 features (using random oversampling)
- PS_a_ii : considered all the 79 features (using SMOTE)
- PS_a_iii : considered all the 79 features (no balancing)

- PS_b_i : considered the 30 more discriminant features (using random oversampling)
- PS_b_ii : considered the 30 more discriminant features (using SMOTE)
- PS_b_iii : considered the 30 more discriminant features (no balancing)
- PS_c_i : considered the 20 more discriminant features (using random oversampling)
- PS_c_ii : considered the 20 more discriminant features (using SMOTE)
- PS_c_iii : considered the 20 more discriminant features (no balancing)
- PS_d_i : considered the 15 more discriminant features (using random oversampling)
- PS_d_ii : considered the 15 more discriminant features (using SMOTE)
- PS_d_iii : considered the 15 more discriminant features (no balancing)
- PS_e_i : considered the 10 more discriminant features (using random oversampling)
- PS_e_ii : considered the 10 more discriminant features (using SMOTE)
- PS_e_iii : considered the 10 more discriminant features (no balancing)

Integration Phase

«TODO: LIST VERSIONS OF THE DATASETS USED DURING THE INTEGRATION PHASE»

B.1 Random Forest

B.1.1 Results from the experiments with the CIC-IDS2017 dataset

Results for the *DoS* attack

Table B.1: Performance of *Random Forest* with the cic-ids2017_DoS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16624 3]
	1	1.00	1.00	1.00		[1 21743]
2	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[7 21738]
3	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[4 21741]
4	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[6 21738]

Table B.2: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_a_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14554 1]
1	1.00	1.00	1.00		[5 19297]

Table B.3: Performance of *Random Forest* with the cic-ids2017_DoS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16625 2]
	1	1.00	1.00	1.00		[1 21743]
2	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[6 21739]
3	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[4 21741]
4	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[8 21737]
5	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[6 21738]

Table B.4: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14554 1]
1	1.00	1.00	1.00		[5 19297]

Table B.5: Performance of *Random Forest* with the cic-ids2017_DoS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16624 3]
	1	1.00	1.00	1.00		[1 21743]
2	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[7 21738]
3	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[3 21742]
4	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16623 3]
	1	1.00	1.00	1.00		[6 21738]

Table B.6: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14554 1]
1	1.00	1.00	1.00		[4 19298]

Table B.7: Performance of *Random Forest* with the cic-ids2017_DoS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16623 4]
	1	1.00	1.00	1.00		[3 21741]
2	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[10 21735]
3	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[7 21738]
4	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[8 21737]
5	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[7 21737]

Table B.8: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_b_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14552 3]
1	1.00	1.00	1.00		[5 19297]

Table B.9: Performance of *Random Forest* with the cic-ids2017_DoS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16622 5]
	1	1.00	1.00	1.00		[1 21743]
2	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[8 21737]
3	0	1.00	1.00	1.00	1.00	[16623 3]
	1	1.00	1.00	1.00		[7 21738]
4	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[8 21736]

Table B.10: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_b_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14552 3]
1	1.00	1.00	1.00		[7 19295]

Table B.11: Performance of *Random Forest* with the cic-ids2017_DoS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16623 4]
	1	1.00	1.00	1.00		[2 21742]
2	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[9 21736]
3	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[7 21738]
4	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[8 21736]

Table B.12: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14552 3]
1	1.00	1.00	1.00		[5 19297]

Table B.13: Performance of *Random Forest* with the cic-ids2017_DoS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16625 2]
	1	1.00	1.00	1.00		[3 21741]
2	0	1.00	1.00	1.00	1.00	[16623 3]
	1	1.00	1.00	1.00		[10 21735]
3	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[8 21737]
4	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16623 3]
	1	1.00	1.00	1.00		[8 21736]

Table B.14: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_c_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14555 0]
1	1.00	1.00	1.00		[6 19296]

Table B.15: Performance of *Random Forest* with the cic-ids2017_DoS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16625 2]
	1	1.00	1.00	1.00		[3 21741]
2	0	1.00	1.00	1.00	1.00	[16623 3]
	1	1.00	1.00	1.00		[12 21733]
3	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[8 21737]
4	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[8 21736]

Table B.16: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_c_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14554 1]
1	1.00	1.00	1.00		[5 19297]

Table B.17: Performance of *Random Forest* with the cic-ids2017_DoS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16624 3]
	1	1.00	1.00	1.00		[3 21741]
2	0	1.00	1.00	1.00	1.00	[16623 3]
	1	1.00	1.00	1.00		[10 21735]
3	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[8 21737]
4	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[8 21736]

Table B.18: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14553 2]
1	1.00	1.00	1.00		[6 19296]

Table B.19: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14554 1]
1	1.00	1.00	1.00		[5 19297]

Table B.20: Performance of *Random Forest* with the cic-ids2017_DoS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16627 0]
	1	1.00	1.00	1.00		[2 21742]
2	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[10 21735]
3	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[9 21736]
4	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[8 21736]

Table B.21: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_d_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14555 0]
1	1.00	1.00	1.00		[6 19296]

Table B.22: Performance of *Random Forest* with the cic-ids2017_DoS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16627 0]
	1	1.00	1.00	1.00		[4 21740]
2	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[9 21736]
3	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[9 21736]
4	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[8 21736]

Table B.23: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_d_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14555 0]
1	1.00	1.00	1.00		[6 19296]

Table B.24: Performance of *Random Forest* with the cic-ids2017_DoS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16626 1]
	1	1.00	1.00	1.00		[3 21741]
2	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[11 21734]
3	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[9 21736]
4	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[8 21736]

Table B.25: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14555 0]
1	1.00	1.00	1.00		[6 19296]

Table B.26: Performance of *Random Forest* with the cic-ids2017_DoS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16624 3]
	1	1.00	1.00	1.00		[2 21742]
2	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[8 21737]
3	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[7 21738]
4	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[8 21736]

Table B.27: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_e_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14554 1]
1	1.00	1.00	1.00		[5 19297]

Table B.28: Performance of *Random Forest* with the cic-ids2017_DoS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16624 3]
	1	1.00	1.00	1.00		[2 21742]
2	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[8 21737]
3	0	1.00	1.00	1.00	1.00	[16622 4]
	1	1.00	1.00	1.00		[7 21738]
4	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16623 3]
	1	1.00	1.00	1.00		[8 21736]

Table B.29: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_e_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14554 1]
1	1.00	1.00	1.00		[5 19297]

Table B.30: Performance of *Random Forest* with the cic-ids2017_DoS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16619 8]
	1	1.00	1.00	1.00		[0 21744]
2	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[9 21736]
3	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[7 21738]
4	0	1.00	1.00	1.00	1.00	[16626 0]
	1	1.00	1.00	1.00		[7 21738]
5	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[8 21736]

Table B.31: Performance of *Random Forest* with *Hold Out* section of the cic-ids2017_DoS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14555 0]
1	1.00	1.00	1.00		[6 19296]

Results for the *Port Scan* attack

Table B.32: Performance of *Random Forest* with the cic-ids2017_PS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[4 26985]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[7 26982]
3	0	1.00	1.00	1.00	0.99	[21647 0]
	1	1.00	1.00	1.00		[1 26988]
4	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[4 26985]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[2 26986]

Table B.33: Performance of *Random Forest* with *Hold Out* section of the PS_a_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.34: Performance of *Random Forest* with the cic-ids2017_PS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[5 26984]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[5 26984]
3	0	1.00	1.00	1.00	0.99	[21647 0]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[5 26984]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[2 26986]

Table B.35: Performance of *Random Forest* with *Hold Out* section of the PS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.36: Performance of *Random Forest* with the cic-ids2017_PS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[5 26984]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[6 26983]
3	0	1.00	1.00	1.00	0.99	[21647 0]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[4 26985]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[2 26986]

Table B.37: Performance of *Random Forest* with *Hold Out* section of the PS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.38: Performance of *Random Forest* with the cic-ids2017_PS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[4 26985]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[7 26982]
3	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21644 3]
	1	1.00	1.00	1.00		[4 26985]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[1 26987]

Table B.39: Performance of *Random Forest* with *Hold Out* section of the PS_b_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.40: Performance of *Random Forest* with the cic-ids2017_PS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[5 26984]
2	0	1.00	1.00	1.00	0.99	[21645 2]
	1	1.00	1.00	1.00		[7 26982]
3	0	1.00	1.00	1.00	0.99	[21647 0]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[5 26984]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[2 26986]

Table B.41: Performance of *Random Forest* with *Hold Out* section of the PS_b_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.42: Performance of *Random Forest* with the cic-ids2017_PS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[4 26985]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[7 26982]
3	0	1.00	1.00	1.00	0.99	[21647 0]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21644 3]
	1	1.00	1.00	1.00		[5 26984]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[2 26986]

Table B.43: Performance of *Random Forest* with *Hold Out* section of the PS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.44: Performance of *Random Forest* with the cic-ids2017_PS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[4 26985]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[7 26982]
3	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21645 2]
	1	1.00	1.00	1.00		[4 26985]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[2 26986]

Table B.45: Performance of *Random Forest* with *Hold Out* section of the PS_c_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.46: Performance of *Random Forest* with the cic-ids2017_PS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[4 26985]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[7 26982]
3	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21645 2]
	1	1.00	1.00	1.00		[5 26984]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[2 26986]

Table B.47: Performance of *Random Forest* with *Hold Out* section of the PS_c_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.48: Performance of *Random Forest* with the cic-ids2017_PS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[5 26984]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[7 26982]
3	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21645 1]
	1	1.00	1.00	1.00		[5 26984]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[1 26987]

Table B.49: Performance of *Random Forest* with *Hold Out* section of the PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.50: Performance of *Random Forest* with the cic-ids2017_PS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[4 26985]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[5 26984]
3	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21645 2]
	1	1.00	1.00	1.00		[4 26985]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[1 26987]

Table B.51: Performance of *Random Forest* with *Hold Out* section of the PS_d_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.52: Performance of *Random Forest* with the cic-ids2017_PS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[4 26985]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[5 26984]
3	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21645 2]
	1	1.00	1.00	1.00		[4 26985]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[1 26987]

Table B.53: Performance of *Random Forest* with *Hold Out* section of the PS_d_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.54: Performance of *Random Forest* with the cic-ids2017_PS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[4 26985]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[5 26984]
3	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21645 2]
	1	1.00	1.00	1.00		[4 26985]
5	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[1 26987]

Table B.55: Performance of *Random Forest* with *Hold Out* section of the PS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.56: Performance of *Random Forest* with the cic-ids2017_PS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[6 26983]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[6 26983]
3	0	1.00	1.00	1.00	0.99	[21643 4]
	1	1.00	1.00	1.00		[4 26985]
4	0	1.00	1.00	1.00	0.99	[21643 4]
	1	1.00	1.00	1.00		[4 26985]
5	0	1.00	1.00	1.00	0.99	[21647 1]
	1	1.00	1.00	1.00		[2 26986]

Table B.57: Performance of *Random Forest* with *Hold Out* section of the PS_e_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.58: Performance of *Random Forest* with the cic-ids2017_PS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[6 26983]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[6 26983]
3	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[0 26989]
4	0	1.00	1.00	1.00	0.99	[21643 4]
	1	1.00	1.00	1.00		[5 26984]
5	0	1.00	1.00	1.00	0.99	[21647 1]
	1	1.00	1.00	1.00		[2 26986]

Table B.59: Performance of *Random Forest* with *Hold Out* section of the PS_e_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

Table B.60: Performance of *Random Forest* with the cic-ids2017_PS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[21648 0]
	1	1.00	1.00	1.00		[6 26983]
2	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[7 26982]
3	0	1.00	1.00	1.00	0.99	[21646 1]
	1	1.00	1.00	1.00		[1 26988]
4	0	1.00	1.00	1.00	0.99	[21643 4]
	1	1.00	1.00	1.00		[4 26985]
5	0	1.00	1.00	1.00	0.99	[21647 1]
	1	1.00	1.00	1.00		[2 26986]

Table B.61: Performance of *Random Forest* with *Hold Out* section of the PS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[19055 0]
1	1.00	1.00	1.00		[2 23858]

B.1.2 Results from the collected dataset(s)

Results for the *DoS* attack

Table B.62: Performance of *Random Forest* with the DoS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.63: Performance of *Random Forest* with *Hold Out* section of the DoS_a_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.64: Performance of *Random Forest* with the DoS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.65: Performance of *Random Forest* with *Hold Out* section of the DoS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.66: Performance of *Random Forest* with the DoS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.67: Performance of *Random Forest* with *Hold Out* section of the DoS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.68: Performance of *Random Forest* with the DoS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.69: Performance of *Random Forest* with *Hold Out* section of the DoS_b_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.70: Performance of *Random Forest* with the DoS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.71: Performance of *Random Forest* with *Hold Out* section of the DoS_b_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.72: Performance of *Random Forest* with the DoS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.73: Performance of *Random Forest* with *Hold Out* section of the DoS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.74: Performance of *Random Forest* with the DoS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.75: Performance of *Random Forest* with *Hold Out* section of the DoS_c_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.76: Performance of *Random Forest* with the DoS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.77: Performance of *Random Forest* with *Hold Out* section of the DoS_c_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.78: Performance of *Random Forest* with the DoS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.79: Performance of *Random Forest* with *Hold Out* section of the DoS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.80: Performance of *Random Forest* with the DoS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.81: Performance of *Random Forest* with *Hold Out* section of the DoS_d_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.82: Performance of *Random Forest* with the DoS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.83: Performance of *Random Forest* with *Hold Out* section of the DoS_d_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.84: Performance of *Random Forest* with the DoS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	0.99	[3045 1]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.85: Performance of *Random Forest* with *Hold Out* section of the DoS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.86: Performance of *Random Forest* with the DoS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.87: Performance of *Random Forest* with *Hold Out* section of the DoS_e_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.88: Performance of *Random Forest* with the DoS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.89: Performance of *Random Forest* with *Hold Out* section of the DoS_e_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.90: Performance of *Random Forest* with the DoS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.91: Performance of *Random Forest* with *Hold Out* section of the DoS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Results for the *Port Scan* attack

Table B.92: Performance of *Random Forest* with the PS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[2 17296]
2	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[0 17298]
3	0	1.00	1.00	1.00	1.00	[1411 0]
	1	1.00	1.00	1.00		[0 17298]
4	0	1.00	1.00	1.00	0.99	[1408 2]
	1	1.00	1.00	1.00		[1 17297]
5	0	1.00	1.00	1.00	0.99	[1409 1]
	1	1.00	1.00	1.00		[0 17298]

Table B.93: Performance of *Random Forest* with *Hold Out* section of the PS_a_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[1197 0]
1	1.00	1.00	1.00		[0 15311]

Table B.94: Performance of *Random Forest* with the PS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[0 17298]
2	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[0 17298]
3	0	1.00	1.00	1.00	1.00	[1411 0]
	1	1.00	1.00	1.00		[0 17298]
4	0	1.00	1.00	1.00	0.99	[1408 2]
	1	1.00	1.00	1.00		[1 17297]
5	0	1.00	1.00	1.00	0.99	[1409 1]
	1	1.00	1.00	1.00		[0 17298]

Table B.95: Performance of *Random Forest* with *Hold Out* section of the PS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[1197 0]
1	1.00	1.00	1.00		[0 15311]

Table B.96: Performance of *Random Forest* with the PS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[0 17298]
2	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[0 17298]
3	0	1.00	1.00	1.00	1.00	[1411 0]
	1	1.00	1.00	1.00		[0 17298]
4	0	1.00	1.00	1.00	0.99	[1408 2]
	1	1.00	1.00	1.00		[1 17297]
5	0	1.00	1.00	1.00	0.99	[1409 1]
	1	1.00	1.00	1.00		[0 17298]

Table B.97: Performance of *Random Forest* with *Hold Out* section of the PS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[1197 0]
1	1.00	1.00	1.00		[0 15311]

Table B.98: Performance of *Random Forest* with the PS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[2 17296]
2	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[2 17296]
3	0	1.00	1.00	1.00	0.99	[1411 0]
	1	1.00	1.00	1.00		[1 17297]
4	0	1.00	1.00	1.00	0.99	[1408 2]
	1	1.00	1.00	1.00		[0 17298]
5	0	1.00	1.00	1.00	0.99	[1409 1]
	1	1.00	1.00	1.00		[1 17297]

Table B.99: Performance of *Random Forest* with *Hold Out* section of the PS_b_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[1197 0]
1	1.00	1.00	1.00		[0 15311]

Table B.100: Performance of *Random Forest* with the PS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[2 17296]
2	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[1 17297]
3	0	1.00	1.00	1.00	1.00	[1411 0]
	1	1.00	1.00	1.00		[0 17298]
4	0	1.00	1.00	1.00	0.99	[1408 2]
	1	1.00	1.00	1.00		[1 17297]
5	0	1.00	1.00	1.00	0.99	[1409 1]
	1	1.00	1.00	1.00		[0 17298]

Table B.101: Performance of *Random Forest* with *Hold Out* section of the PS_b_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[1197 0]
1	1.00	1.00	1.00		[0 15311]

Table B.102: Performance of *Random Forest* with the PS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[2 17296]
2	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[0 17298]
3	0	1.00	1.00	1.00	0.99	[1411 0]
	1	1.00	1.00	1.00		[0 17298]
4	0	1.00	1.00	1.00	0.99	[1408 2]
	1	1.00	1.00	1.00		[0 17298]
5	0	1.00	1.00	1.00	0.99	[1409 1]
	1	1.00	1.00	1.00		[0 17298]

Table B.103: Performance of *Random Forest* with *Hold Out* section of the PS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[1197 0]
1	1.00	1.00	1.00		[0 15311]

Table B.104: Performance of *Random Forest* with the PS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[5 17293]
2	0	0.99	1.00	0.99	0.99	[1409 2]
	1	1.00	1.00	1.00		[9 17289]
3	0	1.00	1.00	1.00	0.99	[1411 0]
	1	1.00	1.00	1.00		[5 17293]
4	0	1.00	1.00	1.00	0.99	[1408 2]
	1	1.00	1.00	1.00		[5 17293]
5	0	1.00	1.00	1.00	0.99	[1409 1]
	1	1.00	1.00	1.00		[3 17295]

Table B.105: Performance of *Random Forest* with *Hold Out* section of the PS_c_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[1197 0]
1	1.00	1.00	1.00		[4 15307]

Table B.106: Performance of *Random Forest* with the PS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[5 17293]
2	0	0.99	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[8 17290]
3	0	1.00	1.00	1.00	0.99	[1411 0]
	1	0.99	1.00	1.00		[4 17294]
4	0	1.00	1.00	1.00	0.99	[1408 2]
	1	1.00	1.00	1.00		[6 17292]
5	0	1.00	1.00	1.00	0.99	[1409 1]
	1	1.00	1.00	1.00		[1 17297]

Table B.107: Performance of *Random Forest* with *Hold Out* section of the PS_c_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[1197 0]
1	1.00	1.00	1.00		[3 15308]

Table B.108: Performance of *Random Forest* with the PS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[5 17293]
2	0	1.00	1.00	1.00	0.99	[1409 2]
	1	1.00	1.00	1.00		[7 17291]
3	0	1.00	1.00	1.00	1.00	[1411 0]
	1	1.00	1.00	1.00		[4 17294]
4	0	1.00	1.00	1.00	0.99	[1408 2]
	1	1.00	1.00	1.00		[5 17293]
5	0	1.00	1.00	1.00	0.99	[1409 1]
	1	1.00	1.00	1.00		[3 17295]

Table B.109: Performance of *Random Forest* with *Hold Out* section of the PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	0.99	[1197 0]
1	1.00	1.00	1.00		[3 15308]

Table B.110: Performance of *Random Forest* with the PS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.92	1.00	0.96	0.99	[1409 2]
	1	1.00	0.99	1.00		[115 17183]
2	0	0.91	1.00	0.95	0.99	[1409 2]
	1	1.00	0.99	1.00		[139 17159]
3	0	0.91	1.00	0.95	0.99	[1411 0]
	1	1.00	0.99	1.00		[137 17161]
4	0	0.92	1.00	0.96	0.99	[1408 2]
	1	1.00	0.99	1.00		[121 17177]
5	0	0.92	1.00	0.96	0.99	[1410 0]
	1	1.00	0.99	1.00		[122 17176]

Table B.111: Performance of *Random Forest* with *Hold Out* section of the PS_d_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.91	1.00	0.95	0.99	[1197 0]
1	1.00	0.99	1.00		[114 15197]

Table B.112: Performance of *Random Forest* with the PS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.92	1.00	0.96	0.99	[1409 2]
	1	1.00	0.99	1.00		[115 17183]
2	0	0.91	1.00	0.95	0.99	[1409 2]
	1	1.00	0.99	1.00		[139 17159]
3	0	0.91	1.00	0.95	0.99	[1411 0]
	1	1.00	0.99	1.00		[137 17161]
4	0	0.92	1.00	0.96	0.99	[1408 2]
	1	1.00	0.99	1.00		[121 17177]
5	0	0.92	1.00	0.96	0.99	[1410 0]
	1	1.00	0.99	1.00		[122 17176]

Table B.113: Performance of *Random Forest* with *Hold Out* section of the PS_d_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.91	1.00	0.95	0.99	[1197 0]
1	1.00	0.99	1.00		[114 15197]

Table B.114: Performance of *Random Forest* with the PS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.99	0.99	0.99	[1395 16]
	1	1.00	1.00	1.00		[0 17298]
2	0	1.00	0.99	1.00	0.99	[1397 14]
	1	1.00	1.00	1.00		[0 17298]
3	0	1.00	0.99	0.99	0.99	[1393 18]
	1	1.00	1.00	1.00		[0 17298]
4	0	1.00	0.99	0.99	0.99	[1391 19]
	1	1.00	1.00	1.00		[0 17298]
5	0	1.00	0.99	0.99	0.99	[1395 15]
	1	1.00	1.00	1.00		[2 17296]

Table B.115: Performance of *Random Forest* with *Hold Out* section of the PS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.99	0.99	0.99	[1181 16]
1	1.00	1.00	1.00		[0 15311]

Table B.116: Performance of *Random Forest* with the PS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.92	1.00	0.96	0.99	[1409 2]
	1	1.00	0.99	1.00		[115 17183]
2	0	0.91	1.00	0.95	0.99	[1409 2]
	1	1.00	0.99	1.00		[139 17159]
3	0	0.91	1.00	0.95	0.99	[1411 0]
	1	1.00	0.99	1.00		[137 17161]
4	0	0.92	1.00	0.96	0.99	[1408 2]
	1	1.00	0.99	1.00		[121 17177]
5	0	0.92	1.00	0.96	0.99	[1410 0]
	1	1.00	0.99	1.00		[122 17176]

Table B.117: Performance of *Random Forest* with *Hold Out* section of the PS_e_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.91	1.00	0.95	0.99	[1197 0]
1	1.00	0.99	1.00		[114 15197]

Table B.118: Performance of *Random Forest* with the PS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.92	1.00	0.96	0.99	[1409 2]
	1	1.00	0.99	1.00		[115 17183]
2	0	0.91	1.00	0.95	0.99	[1409 2]
	1	1.00	0.99	1.00		[139 17159]
3	0	0.91	1.00	0.95	0.99	[1411 0]
	1	1.00	0.99	1.00		[137 17161]
4	0	0.92	1.00	0.96	0.99	[1408 2]
	1	1.00	0.99	1.00		[121 17177]
5	0	0.92	1.00	0.96	0.99	[1410 0]
	1	1.00	0.99	1.00		[122 17176]

Table B.119: Performance of *Random Forest* with *Hold Out* section of the PS_e_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.91	1.00	0.95	0.99	[1197 0]
1	1.00	0.99	1.00		[114 15197]

Table B.120: Performance of *Random Forest* with the PS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.99	0.99	0.99	[1392 19]
	1	1.00	1.00	1.00		[0 17298]
2	0	1.00	0.99	0.99	0.99	[1393 18]
	1	1.00	1.00	1.00		[0 17298]
3	0	1.00	0.99	0.99	0.99	[1390 21]
	1	1.00	1.00	1.00		[0 17298]
4	0	1.00	0.99	0.99	0.99	[1390 20]
	1	1.00	1.00	1.00		[0 17298]
5	0	1.00	0.99	0.99	0.99	[1394 16]
	1	1.00	1.00	1.00		[0 17298]

Table B.121: Performance of *Random Forest* with *Hold Out* section of the PS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.99	0.99	0.99	[1181 16]
1	1.00	0.99	1.00		[0 15311]

B.2 K-means

B.2.1 Results from the experiments with the CIC-IDS2017 dataset

Results for the *DoS* attack

Table B.122: Performance of *K-means* with the cic-ids2017_DoS_a_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.53	0.80	0.64	0.61	[13315 3312]
		1	0.75	0.46	0.57		[11806 9938]
	92	0	0.53	0.80	0.64	0.61	[13323 3304]
		1	0.75	0.46	0.57		[11806 9938]
	167	0	0.53	0.80	0.64	0.61	[13314 3313]
		1	0.75	0.46	0.57		[11806 9938]
	208	0	0.53	0.80	0.64	0.61	[13323 3304]
		1	0.75	0.46	0.57		[11806 9938]
	1942	0	0.53	0.80	0.64	0.61	[13315 3312]
		1	0.75	0.46	0.57		[11806 9938]
2	1	0	0.25	0.20	0.22	0.39	[3324 13302]
		1	0.47	0.54	0.50		[9934 11811]
	92	0	0.53	0.80	0.64	0.61	[13301 3325]
		1	0.75	0.46	0.57		[11811 9934]
	167	0	0.25	0.20	0.22	0.39	[3325 13301]
		1	0.47	0.54	0.50		[9934 11811]
	208	0	0.25	0.20	0.22	0.39	[3322 13304]
		1	0.47	0.54	0.50		[9934 11811]
	1942	0	0.53	0.80	0.64	0.61	[13304 3322]
		1	0.75	0.46	0.57		[11811 9934]
3	1	0	0.53	0.81	0.64	0.61	[13395 3231]
		1	0.75	0.45	0.57		[11890 9855]
	92	0	0.25	0.19	0.22	0.39	[3230 13396]
		1	0.47	0.55	0.51		[9855 11890]
	167	0	0.53	0.81	0.64	0.61	[13397 3229]
		1	0.75	0.45	0.57		[11890 9855]
	208	0	0.53	0.81	0.64	0.61	[13388 3238]
		1	0.75	0.45	0.57		[11890 9855]
	1942	0	0.53	0.81	0.64	0.61	[13395 3231]
		1	0.75	0.45	0.57		[11890 9855]
4	1	0	0.25	0.20	0.22	0.40	[3314 13312]
		1	0.47	0.55	0.51		[9861 11884]
	92	0	0.25	0.20	0.22	0.40	[3314 13312]
		1	0.47	0.55	0.51		[9861 11884]
	167	0	0.25	0.20	0.22	0.40	[3314 13312]
		1	0.47	0.55	0.51		[9861 11884]
	208	0	0.25	0.20	0.22	0.40	[3314 13312]
		1	0.47	0.55	0.51		[9861 11884]
	1942	0	0.53	0.80	0.64	0.60	[13312 3314]
		1	0.75	0.45	0.56		[11884 9861]
5	1	0	0.53	0.80	0.64	0.61	[13309 3317]
		1	0.75	0.46	0.57		[11634 10110]
	92	0	0.53	0.80	0.64	0.61	[13309 3317]
		1	0.75	0.46	0.57		[11634 10110]
	167	0	0.25	0.20	0.22	0.39	[3316 13310]

Continues on next page

Table B.122 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.47	0.54	0.50		[10110 11634]
	208	0	0.53	0.80	0.64	0.61	[13310 3316]
		1	0.75	0.46	0.57		[11634 10110]
	1942	0	0.25	0.20	0.22	0.39	[3317 13309]
		1	0.47	0.54	0.50		[10110 11634]

Table B.123: Performance of *K-means* with the cic-ids2017_DoS_a_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.25	0.20	0.22	0.39	[3312 13315]
		1	0.47	0.54	0.50		[9938 11806]
	92	0	0.53	0.80	0.64	0.61	[13318 3309]
		1	0.75	0.46	0.57		[11806 9938]
	167	0	0.23	0.25	0.24	0.32	[4176 12451]
		1	0.39	0.37	0.38		[13694 8050]
	208	0	0.53	0.80	0.64	0.61	[13316 3311]
		1	0.75	0.46	0.57		[11806 9938]
	1942	0	0.53	0.80	0.64	0.61	[13312 3315]
		1	0.75	0.46	0.57		[11807 9937]
2	1	0	0.25	0.20	0.22	0.39	[3297 13329]
		1	0.47	0.54	0.50		[9935 11810]
	92	0	0.53	0.80	0.64	0.61	[13329 3297]
		1	0.75	0.46	0.57		[11810 9935]
	167	0	0.53	0.80	0.64	0.61	[13329 3297]
		1	0.75	0.46	0.57		[11810 9935]
	208	0	0.53	0.80	0.64	0.61	[13329 3297]
		1	0.75	0.46	0.57		[11810 9935]
	1942	0	0.25	0.20	0.22	0.39	[3297 13329]
		1	0.47	0.54	0.50		[9935 11810]
3	1	0	0.25	0.20	0.22	0.39	[3259 13367]
		1	0.47	0.55	0.51		[9852 11893]
	92	0	0.53	0.80	0.64	0.61	[13368 3258]
		1	0.75	0.45	0.57		[11893 9852]
	167	0	0.53	0.80	0.64	0.61	[13368 3258]
		1	0.75	0.45	0.57		[11893 9852]
	208	0	0.53	0.80	0.64	0.61	[13368 3258]
		1	0.75	0.45	0.57		[11893 9852]
	1942	0	0.53	0.80	0.64	0.61	[13368 3258]
		1	0.75	0.45	0.57		[11893 9852]
4	1	0	0.25	0.20	0.22	0.40	[3274 13352]
		1	0.47	0.55	0.51		[9861 11884]
	92	0	0.53	0.80	0.64	0.60	[13352 3274]
		1	0.75	0.45	0.57		[11884 9861]
	167	0	0.53	0.80	0.64	0.60	[13352 3274]
		1	0.75	0.45	0.57		[11884 9861]
	208	0	0.53	0.80	0.64	0.60	[13352 3274]
		1	0.75	0.45	0.57		[11884 9861]
	1942	0	0.53	0.80	0.64	0.60	[13352 3274]
		1	0.75	0.45	0.57		[11884 9861]
5	1	0	0.53	0.80	0.64	0.61	[13310 3316]
		1	0.75	0.46	0.57		[11634 10110]
	92	0	0.53	0.80	0.64	0.61	[13310 3316]
		1	0.75	0.46	0.57		[11634 10110]
	167	0	0.25	0.20	0.22	0.39	[3316 13310]

Continues on next page

Table B.123 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.47	0.54	0.50		[10110 11634]
	208	0	0.53	0.80	0.64	0.61	[13310 3316]
		1	0.75	0.46	0.57		[11634 10110]
	1942	0	0.25	0.20	0.22	0.39	[3316 13310]
		1	0.47	0.54	0.50		[10110 11634]

Table B.124: Performance of *K-means* with the cic-ids2017_DoS_a_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.53	0.81	0.64	0.61	[13525 3102]
		1	0.76	0.46	0.57		[11807 9937]
	92	0	0.54	0.83	0.65	0.62	[13831 2796]
		1	0.78	0.46	0.58		[11809 9935]
	167	0	0.53	0.81	0.64	0.61	[13525 3102]
		1	0.76	0.46	0.57		[11807 9937]
	208	0	0.54	0.84	0.66	0.62	[13902 2725]
		1	0.78	0.46	0.58		[11809 9935]
	1942	0	0.53	0.81	0.64	0.61	[13525 3102]
		1	0.76	0.46	0.57		[11807 9937]
2	1	0	0.24	0.19	0.21	0.39	[3080 13546]
		1	0.47	0.54	0.50		[9934 11811]
	92	0	0.53	0.81	0.65	0.61	[13546 3080]
		1	0.76	0.46	0.57		[11811 9934]
	167	0	0.24	0.19	0.21	0.39	[3080 13546]
		1	0.47	0.54	0.50		[9934 11811]
	208	0	0.53	0.81	0.65	0.61	[13547 3079]
		1	0.76	0.46	0.57		[11811 9934]
	1942	0	0.24	0.19	0.21	0.39	[3079 13547]
		1	0.47	0.54	0.50		[9934 11811]
3	1	0	0.23	0.18	0.20	0.39	[2985 13641]
		1	0.47	0.55	0.50		[9852 11893]
	92	0	0.53	0.82	0.65	0.61	[13641 2985]
		1	0.77	0.45	0.57		[11893 9852]
	167	0	0.53	0.82	0.65	0.61	[13641 2985]
		1	0.77	0.45	0.57		[11893 9852]
	208	0	0.53	0.82	0.65	0.61	[13641 2985]
		1	0.77	0.45	0.57		[11893 9852]
	1942	0	0.23	0.18	0.20	0.39	[2985 13641]
		1	0.47	0.55	0.50		[9852 11893]
4	1	0	0.53	0.82	0.65	0.61	[13574 3052]
		1	0.76	0.45	0.57		[11884 9861]
	92	0	0.53	0.82	0.65	0.61	[13574 3052]
		1	0.76	0.45	0.57		[11884 9861]
	167	0	0.24	0.18	0.21	0.39	[3052 13574]
		1	0.47	0.55	0.50		[9861 11884]
	208	0	0.24	0.18	0.21	0.39	[3052 13574]
		1	0.47	0.55	0.50		[9861 11884]
	1942	0	0.53	0.82	0.65	0.61	[13573 3053]
		1	0.76	0.45	0.57		[11884 9861]
5	1	0	0.23	0.18	0.20	0.38	[3052 13574]
		1	0.46	0.54	0.50		[10107 11637]
	92	0	0.54	0.82	0.65	0.62	[13573 3053]
		1	0.77	0.46	0.58		[11637 10107]
	167	0	0.23	0.18	0.20	0.38	[3053 13573]

Continues on next page

Table B.124 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.46	0.54	0.50		[10107 11637]
	208	0	0.54	0.82	0.65	0.62	[13573 3053]
		1	0.77	0.46	0.58		[11637 10107]
	1942	0	0.54	0.82	0.65	0.62	[13573 3053]
		1	0.77	0.46	0.58		[11637 10107]

Table B.125: Performance of *K-means* with the cic-ids2017_DoS_b_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.17	0.17	0.17	0.28	[2893 13734]
		1	0.37	0.37	0.37		[13756 7988]
	92	0	0.66	0.93	0.77	0.76	[15385 1242]
		1	0.92	0.63	0.75		[7991 13753]
	167	0	0.66	0.93	0.77	0.76	[15385 1242]
		1	0.92	0.63	0.75		[7991 13753]
	208	0	0.63	0.83	0.72	0.72	[13734 2893]
		1	0.83	0.63	0.72		[7988 13756]
	1942	0	0.66	0.93	0.77	0.76	[15385 1242]
		1	0.92	0.63	0.75		[7991 13753]
2	1	0	0.66	0.93	0.77	0.76	[15393 1233]
		1	0.92	0.64	0.75		[7849 13896]
	92	0	0.66	0.93	0.77	0.76	[15392 1234]
		1	0.92	0.64	0.75		[7849 13896]
	167	0	0.08	0.07	0.08	0.24	[1234 15392]
		1	0.34	0.36	0.35		[13896 7849]
	208	0	0.17	0.17	0.17	0.28	[2876 13750]
		1	0.36	0.36	0.36		[13897 7848]
	1942	0	0.17	0.17	0.17	0.28	[2876 13750]
		1	0.36	0.36	0.36		[13897 7848]
3	1	0	0.63	0.82	0.71	0.71	[13705 2921]
		1	0.82	0.63	0.72		[8018 13727]
	92	0	0.18	0.18	0.18	0.29	[2922 13704]
		1	0.37	0.37	0.37		[13727 8018]
	167	0	0.63	0.82	0.71	0.71	[13704 2922]
		1	0.82	0.63	0.72		[8018 13727]
	208	0	0.63	0.82	0.71	0.71	[13704 2922]
		1	0.82	0.63	0.72		[8018 13727]
	1942	0	0.18	0.18	0.18	0.29	[2922 13704]
		1	0.37	0.37	0.37		[13727 8018]
4	1	0	0.63	0.82	0.71	0.72	[13679 2947]
		1	0.82	0.63	0.72		[7959 13786]
	92	0	0.08	0.07	0.08	0.24	[1219 15407]
		1	0.34	0.37	0.35		[13784 7961]
	167	0	0.63	0.82	0.71	0.72	[13679 2947]
		1	0.82	0.63	0.72		[7959 13786]
	208	0	0.08	0.07	0.08	0.24	[1219 15407]
		1	0.34	0.37	0.35		[13784 7961]
	1942	0	0.66	0.93	0.77	0.76	[15407 1219]
		1	0.92	0.63	0.75		[7961 13784]
5	1	0	0.66	0.93	0.77	0.76	[15412 1214]
		1	0.92	0.64	0.76		[7805 13939]
	92	0	0.64	0.83	0.72	0.72	[13727 2899]
		1	0.83	0.64	0.72		[7805 13939]
	167	0	0.08	0.07	0.08	0.24	[1214 15412]

Continues on next page

Table B.125 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.34	0.36	0.35		[13939 7805]
	208	0	0.66	0.93	0.77	0.76	[15412 1214]
		1	0.92	0.64	0.76		[7805 13939]
	1942	0	0.17	0.17	0.17	0.28	[2898 13728]
		1	0.36	0.36	0.36		[13939 7805]

Table B.126: Performance of *K-means* with the cic-ids2017_DoS_b_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.17	0.17	0.17	0.28	[2893 13734]
		1	0.37	0.37	0.37		[13756 7988]
	92	0	0.66	0.93	0.77	0.76	[15385 1242]
		1	0.92	0.63	0.75		[7991 13753]
	167	0	0.66	0.93	0.77	0.76	[15385 1242]
		1	0.92	0.63	0.75		[7991 13753]
	208	0	0.63	0.83	0.72	0.72	[13734 2893]
		1	0.83	0.63	0.72		[7988 13756]
	1942	0	0.66	0.93	0.77	0.76	[15385 1242]
		1	0.92	0.63	0.75		[7991 13753]
2	1	0	0.64	0.83	0.72	0.72	[13749 2877]
		1	0.83	0.64	0.72		[7848 13897]
	92	0	0.64	0.83	0.72	0.72	[13749 2877]
		1	0.83	0.64	0.72		[7848 13897]
	167	0	0.08	0.07	0.08	0.24	[1232 15394]
		1	0.34	0.36	0.35		[13896 7849]
	208	0	0.17	0.17	0.17	0.28	[2877 13749]
		1	0.36	0.36	0.36		[13897 7848]
	1942	0	0.17	0.17	0.17	0.28	[2877 13749]
		1	0.36	0.36	0.36		[13897 7848]
3	1	0	0.63	0.82	0.71	0.71	[13704 2922]
		1	0.82	0.63	0.72		[8018 13727]
	92	0	0.63	0.82	0.71	0.71	[13704 2922]
		1	0.82	0.63	0.72		[8018 13727]
	167	0	0.08	0.07	0.08	0.24	[1209 15417]
		1	0.34	0.37	0.35		[13726 8019]
	208	0	0.63	0.82	0.71	0.71	[13704 2922]
		1	0.82	0.63	0.72		[8018 13727]
	1942	0	0.66	0.93	0.77	0.76	[15417 1209]
		1	0.92	0.63	0.75		[8019 13726]
4	1	0	0.63	0.82	0.71	0.72	[13679 2947]
		1	0.82	0.63	0.72		[7959 13786]
	92	0	0.66	0.93	0.77	0.76	[15408 1218]
		1	0.92	0.63	0.75		[7961 13784]
	167	0	0.63	0.82	0.71	0.72	[13679 2947]
		1	0.82	0.63	0.72		[7959 13786]
	208	0	0.08	0.07	0.08	0.24	[1218 15408]
		1	0.34	0.37	0.35		[13784 7961]
	1942	0	0.66	0.93	0.77	0.76	[15408 1218]
		1	0.92	0.63	0.75		[7961 13784]
5	1	0	0.66	0.93	0.77	0.76	[15414 1212]
		1	0.92	0.64	0.76		[7805 13939]
	92	0	0.66	0.93	0.77	0.76	[15414 1212]
		1	0.92	0.64	0.76		[7805 13939]
	167	0	0.08	0.07	0.08	0.24	[1212 15414]

Continues on next page

Table B.126 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.34	0.36	0.35		[13939 7805]
	208	0	0.66	0.93	0.77	0.76	[15414 1212]
		1	0.92	0.64	0.76		[7805 13939]
	1942	0	0.66	0.93	0.77	0.76	[15414 1212]
		1	0.92	0.64	0.76		[7805 13939]

Table B.127: Performance of *K-means* with the cic-ids2017_DoS_b_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.63	0.83	0.72	0.72	[13766 2861]
		1	0.83	0.63	0.72		[7991 13753]
	92	0	0.63	0.83	0.72	0.72	[13766 2861]
		1	0.83	0.63	0.72		[7991 13753]
	167	0	0.63	0.83	0.72	0.72	[13766 2861]
		1	0.83	0.63	0.72		[7991 13753]
	208	0	0.63	0.83	0.72	0.72	[13766 2861]
		1	0.83	0.63	0.72		[7991 13753]
	1942	0	0.66	0.93	0.77	0.76	[15394 1233]
		1	0.92	0.63	0.75		[7991 13753]
2	1	0	0.08	0.07	0.08	0.24	[1222 15404]
		1	0.34	0.36	0.35		[13896 7849]
	92	0	0.66	0.93	0.77	0.76	[15404 1222]
		1	0.92	0.64	0.75		[7849 13896]
	167	0	0.08	0.07	0.08	0.24	[1222 15404]
		1	0.34	0.36	0.35		[13896 7849]
	208	0	0.66	0.93	0.77	0.76	[15404 1222]
		1	0.92	0.64	0.75		[7849 13896]
	1942	0	0.17	0.17	0.17	0.28	[2830 13796]
		1	0.36	0.36	0.36		[13896 7849]
3	1	0	0.08	0.07	0.08	0.24	[1197 15429]
		1	0.34	0.37	0.35		[13726 8019]
	92	0	0.66	0.93	0.77	0.76	[15429 1197]
		1	0.92	0.63	0.75		[8019 13726]
	167	0	0.63	0.83	0.72	0.72	[13749 2877]
		1	0.83	0.63	0.72		[8019 13726]
	208	0	0.63	0.83	0.72	0.72	[13749 2877]
		1	0.83	0.63	0.72		[8019 13726]
	1942	0	0.63	0.83	0.72	0.72	[13749 2877]
		1	0.83	0.63	0.72		[8019 13726]
4	1	0	0.63	0.83	0.72	0.72	[13728 2898]
		1	0.83	0.63	0.72		[7961 13784]
	92	0	0.66	0.93	0.77	0.76	[15420 1206]
		1	0.92	0.63	0.75		[7961 13784]
	167	0	0.17	0.17	0.17	0.28	[2898 13728]
		1	0.37	0.37	0.37		[13784 7961]
	208	0	0.08	0.07	0.08	0.24	[1206 15420]
		1	0.34	0.37	0.35		[13784 7961]
	1942	0	0.66	0.93	0.77	0.76	[15420 1206]
		1	0.92	0.63	0.75		[7961 13784]
5	1	0	0.08	0.07	0.08	0.23	[1202 15424]
		1	0.34	0.36	0.35		[13939 7805]
	92	0	0.66	0.93	0.77	0.77	[15424 1202]
		1	0.92	0.64	0.76		[7805 13939]
	167	0	0.08	0.07	0.08	0.23	[1202 15424]

Continues on next page

Table B.127 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.34	0.36	0.35		[13939 7805]
	208	0	0.17	0.17	0.17	0.28	[2855 13771]
		1	0.36	0.36	0.36		[13939 7805]
	1942	0	0.66	0.93	0.77	0.77	[15424 1202]
		1	0.92	0.64	0.76		[7805 13939]

Table B.128: Performance of *K-means* with the cic-ids2017_DoS_c_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	92	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	167	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	208	0	1.00	0.33	0.50	0.71	[5479 11148]
		1	0.66	1.00	0.80		[0 21744]
	1942	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
2	1	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	167	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	208	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
3	1	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	92	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	167	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	208	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	1942	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
4	1	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	167	0	1.00	0.32	0.49	0.71	[5372 11254]
		1	0.66	1.00	0.79		[0 21745]
	208	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
5	1	0	1.00	0.33	0.49	0.71	[5420 11206]
		1	0.66	1.00	0.80		[1 21743]
	92	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	167	0	0.34	0.67	0.45	0.29	[11206 5420]

Continues on next page

Table B.128 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[21743 1]
	208	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	1942	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]

Table B.129: Performance of *K-means* with the cic-ids2017_DoS_c_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	92	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	167	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	208	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	1942	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
2	1	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	167	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	208	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
3	1	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	92	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	167	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	208	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	1942	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
4	1	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	167	0	1.00	0.32	0.49	0.71	[5372 11254]
		1	0.66	1.00	0.79		[0 21745]
	208	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
5	1	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	92	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	167	0	0.34	0.67	0.45	0.29	[11206 5420]

Continues on next page

Table B.129 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[21743 1]
	208	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	1942	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]

Table B.130: Performance of *K-means* with the cic-ids2017_DoS_c_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.44	0.87	0.59	0.47	[14486 2141]
		1	0.63	0.16	0.26		[18157 3587]
	92	0	0.37	0.13	0.19	0.53	[2141 14486]
		1	0.56	0.84	0.67		[3587 18157]
	167	0	0.44	0.87	0.59	0.47	[14486 2141]
		1	0.63	0.16	0.26		[18157 3587]
	208	0	0.44	0.87	0.59	0.47	[14486 2141]
		1	0.63	0.16	0.26		[18157 3587]
	1942	0	0.44	0.87	0.59	0.47	[14486 2141]
		1	0.63	0.16	0.26		[18157 3587]
2	1	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
	92	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
	167	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
	208	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
	1942	0	0.44	0.87	0.59	0.47	[14448 2178]
		1	0.63	0.17	0.27		[18038 3707]
3	1	0	0.44	0.86	0.59	0.47	[14368 2258]
		1	0.62	0.17	0.26		[18119 3626]
	92	0	0.44	0.86	0.59	0.47	[14368 2258]
		1	0.62	0.17	0.26		[18119 3626]
	167	0	0.44	0.86	0.59	0.47	[14368 2258]
		1	0.62	0.17	0.26		[18119 3626]
	208	0	0.44	0.86	0.59	0.47	[14368 2258]
		1	0.62	0.17	0.26		[18119 3626]
	1942	0	0.44	0.86	0.59	0.47	[14368 2258]
		1	0.62	0.17	0.26		[18119 3626]
4	1	0	0.44	0.86	0.59	0.47	[14366 2260]
		1	0.62	0.17	0.26		[18100 3645]
	92	0	0.44	0.86	0.59	0.47	[14366 2260]
		1	0.62	0.17	0.26		[18100 3645]
	167	0	0.38	0.14	0.20	0.53	[2260 14366]
		1	0.56	0.83	0.67		[3645 18100]
	208	0	0.38	0.14	0.20	0.53	[2260 14366]
		1	0.56	0.83	0.67		[3645 18100]
	1942	0	0.44	0.86	0.59	0.47	[14366 2260]
		1	0.62	0.17	0.26		[18100 3645]
5	1	0	0.38	0.13	0.20	0.53	[2215 14411]
		1	0.56	0.83	0.67		[3591 18153]
	92	0	0.44	0.87	0.59	0.47	[14411 2215]
		1	0.62	0.17	0.26		[18153 3591]
	167	0	0.44	0.87	0.59	0.47	[14411 2215]

Continues on next page

Table B.130 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.62	0.17	0.26		[18153 3591]
	208	0	0.44	0.87	0.59	0.47	[14411 2215]
		1	0.62	0.17	0.26		[18153 3591]
	1942	0	0.38	0.13	0.20	0.53	[2215 14411]
		1	0.56	0.83	0.67		[3591 18153]

Table B.131: Performance of *K-means* with the cic-ids2017_DoS_d_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	1.00	0.33	0.50	0.71	[5479 11148]
		1	0.66	1.00	0.80		[0 21744]
	92	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	167	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	208	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	1942	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
2	1	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	167	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	208	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
3	1	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	92	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	167	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	208	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	1942	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
4	1	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	167	0	1.00	0.32	0.49	0.71	[5372 11254]
		1	0.66	1.00	0.79		[0 21745]
	208	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
5	1	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	92	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	167	0	0.34	0.67	0.45	0.29	[11206 5420]

Continues on next page

Table B.131 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[21743 1]
	208	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	1942	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]

Table B.132: Performance of *K-means* with the cic-ids2017_DoS_d_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	92	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	167	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	208	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	1942	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
2	1	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	167	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	208	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
3	1	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	92	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	167	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	208	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	1942	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
4	1	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	167	0	1.00	0.32	0.49	0.71	[5372 11254]
		1	0.66	1.00	0.79		[0 21745]
	208	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
5	1	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	92	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	167	0	0.34	0.67	0.45	0.29	[11206 5420]

Continues on next page

Table B.132 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[21743 1]
	208	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	1942	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]

Table B.133: Performance of *K-means* with the cic-ids2017_DoS_d_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.44	0.87	0.59	0.47	[14487 2140]
		1	0.63	0.16	0.26		[18163 3581]
	92	0	0.37	0.13	0.19	0.53	[2140 14487]
		1	0.56	0.84	0.67		[3581 18163]
	167	0	0.37	0.13	0.19	0.53	[2140 14487]
		1	0.56	0.84	0.67		[3581 18163]
	208	0	0.44	0.87	0.59	0.47	[14487 2140]
		1	0.63	0.16	0.26		[18163 3581]
	1942	0	0.44	0.87	0.59	0.47	[14487 2140]
		1	0.63	0.16	0.26		[18163 3581]
2	1	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
	92	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
	167	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
	208	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
	1942	0	0.44	0.87	0.59	0.47	[14449 2177]
		1	0.63	0.17	0.27		[18039 3706]
3	1	0	0.44	0.86	0.59	0.47	[14369 2257]
		1	0.62	0.17	0.26		[18119 3626]
	92	0	0.44	0.86	0.59	0.47	[14369 2257]
		1	0.62	0.17	0.26		[18119 3626]
	167	0	0.44	0.86	0.59	0.47	[14369 2257]
		1	0.62	0.17	0.26		[18119 3626]
	208	0	0.38	0.14	0.20	0.53	[2257 14369]
		1	0.56	0.83	0.67		[3626 18119]
	1942	0	0.44	0.86	0.59	0.47	[14369 2257]
		1	0.62	0.17	0.26		[18119 3626]
4	1	0	0.44	0.86	0.59	0.47	[14370 2256]
		1	0.62	0.17	0.26		[18101 3644]
	92	0	0.44	0.86	0.59	0.47	[14370 2256]
		1	0.62	0.17	0.26		[18101 3644]
	167	0	0.44	0.86	0.59	0.47	[14370 2256]
		1	0.62	0.17	0.26		[18101 3644]
	208	0	0.38	0.14	0.20	0.53	[2256 14370]
		1	0.56	0.83	0.67		[3644 18101]
	1942	0	0.44	0.86	0.59	0.47	[14370 2256]
		1	0.62	0.17	0.26		[18101 3644]
5	1	0	0.44	0.87	0.59	0.47	[14414 2212]
		1	0.62	0.17	0.26		[18155 3589]
	92	0	0.44	0.87	0.59	0.47	[14414 2212]
		1	0.62	0.17	0.26		[18155 3589]
	167	0	0.44	0.87	0.59	0.47	[14414 2212]

Continues on next page

Table B.133 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.62	0.17	0.26		[18155 3589]
	208	0	0.38	0.13	0.20	0.53	[2212 14414]
		1	0.56	0.83	0.67		[3589 18155]
	1942	0	0.38	0.13	0.20	0.53	[2212 14414]
		1	0.56	0.83	0.67		[3589 18155]

Table B.134: Performance of *K-means* with the cic-ids2017_DoS_e_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	92	0	1.00	0.33	0.50	0.71	[5479 11148]
		1	0.66	1.00	0.80		[0 21744]
	167	0	1.00	0.33	0.50	0.71	[5479 11148]
		1	0.66	1.00	0.80		[0 21744]
	208	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	1942	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
2	1	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	92	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	167	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	208	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
3	1	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	92	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	167	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	208	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	1942	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
4	1	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	167	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	208	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
5	1	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	92	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	167	0	0.34	0.67	0.45	0.29	[11206 5420]

Continues on next page

Table B.134 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[21743 1]
	208	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	1942	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]

Table B.135: Performance of *K-means* with the cic-ids2017_DoS_e_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	92	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	167	0	1.00	0.33	0.50	0.71	[5479 11148]
		1	0.66	1.00	0.80		[0 21744]
	208	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	1942	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
2	1	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	92	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	167	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	208	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	1942	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
3	1	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	92	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	167	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	208	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	1942	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
4	1	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	167	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	208	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
5	1	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	92	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	167	0	0.34	0.67	0.45	0.29	[11206 5420]

Continues on next page

Table B.135 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[21743 1]
	208	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	1942	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]

Table B.136: Performance of *K-means* with the cic-ids2017_DoS_e_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	92	0	1.00	0.33	0.50	0.71	[5479 11148]
		1	0.66	1.00	0.80		[0 21744]
	167	0	1.00	0.33	0.50	0.71	[5479 11148]
		1	0.66	1.00	0.80		[0 21744]
	208	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
	1942	0	0.34	0.67	0.45	0.29	[11148 5479]
		1	0.00	0.00	0.00		[21744 0]
2	1	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	92	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
	167	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	208	0	1.00	0.32	0.48	0.71	[5310 11316]
		1	0.66	1.00	0.79		[0 21745]
	1942	0	0.34	0.68	0.46	0.29	[11316 5310]
		1	0.00	0.00	0.00		[21745 0]
3	1	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	92	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	167	0	1.00	0.33	0.50	0.71	[5476 11150]
		1	0.66	1.00	0.80		[1 21744]
	208	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
	1942	0	0.34	0.67	0.45	0.29	[11150 5476]
		1	0.00	0.00	0.00		[21744 1]
4	1	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	92	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	167	0	1.00	0.32	0.49	0.71	[5372 11254]
		1	0.66	1.00	0.79		[0 21745]
	208	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
	1942	0	0.34	0.68	0.45	0.29	[11254 5372]
		1	0.00	0.00	0.00		[21745 0]
5	1	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	92	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	167	0	0.34	0.67	0.45	0.29	[11206 5420]

Continues on next page

Table B.136 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[21743 1]
	208	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]
	1942	0	0.34	0.67	0.45	0.29	[11206 5420]
		1	0.00	0.00	0.00		[21743 1]

Table B.137: Performance of *K-means* with *Hold Out* section of the cic-ids2017_DoS_a dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.53	0.81	0.64	0.61	[11824 2731]
1	0.76	0.45	0.57		[10592 8710]

Table B.138: Performance of *K-means* with *Hold Out* section of the cic-ids2017_DoS_b dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.66	0.93	0.77	0.76	[13468 1087]
1	0.92	0.63	0.75		[7054 12248]

Table B.139: Performance of *K-means* with *Hold Out* section of the cic-ids2017_DoS_c dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.44	0.87	0.59	0.47	[12643 1912]
1	0.64	0.17	0.27		[15973 3329]

Table B.140: Performance of *K-means* with *Hold Out* section of the cic-ids2017_DoS_d dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.44	0.87	0.59	0.47	[12641 1914]
1	0.63	0.17	0.27		[15973 3329]

Table B.141: Performance of *K-means* with *Hold Out* section of the cic-ids2017_DoS_e dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.34	0.68	0.45	0.29	[9842 4713]
1	0.00	0.00	0.00		[19301 1]

Results for the *PS* attack

Table B.142: Performance of *K-means* with the cic-ids2017_PS_a_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
	92	0	0.16	0.24	0.19	0.11	[5124 16524]
		1	0.00	0.00	0.00		[26982 7]
	167	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
	208	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
	1942	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
2	1	0	1.00	0.77	0.87	0.90	[16561 5086]
		1	0.84	1.00	0.91		[17 26972]
	92	0	0.16	0.23	0.19	0.10	[5086 16561]
		1	0.00	0.00	0.00		[26972 17]
	167	0	0.16	0.23	0.19	0.10	[5086 16561]
		1	0.00	0.00	0.00		[26972 17]
	208	0	0.16	0.23	0.19	0.10	[5086 16561]
		1	0.00	0.00	0.00		[26972 17]
	1942	0	0.16	0.23	0.19	0.10	[5086 16561]
		1	0.00	0.00	0.00		[26972 17]
3	1	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
	92	0	1.00	0.76	0.86	0.89	[16374 5273]
		1	0.84	1.00	0.91		[10 26979]
	167	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
	208	0	1.00	0.76	0.86	0.89	[16374 5273]
		1	0.84	1.00	0.91		[10 26979]
	1942	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
4	1	0	1.00	0.76	0.86	0.89	[16435 5212]
		1	0.84	1.00	0.91		[12 26977]
	92	0	1.00	0.76	0.86	0.89	[16435 5212]
		1	0.84	1.00	0.91		[12 26977]
	167	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
	208	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
	1942	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
5	1	0	1.00	0.76	0.86	0.89	[16491 5157]
		1	0.84	1.00	0.91		[12 26976]
	92	0	1.00	0.76	0.86	0.89	[16491 5157]
		1	0.84	1.00	0.91		[12 26976]
	167	0	0.16	0.24	0.19	0.11	[5157 16491]
		1	0.00	0.00	0.00		[26976 12]
	208	0	1.00	0.76	0.86	0.89	[16491 5157]
		1	0.84	1.00	0.91		[12 26976]
	1942	0	0.16	0.24	0.19	0.11	[5157 16491]
		1	0.00	0.00	0.00		[26976 12]

Table B.143: Performance of *K-means* with the cic-ids2017_PS_a_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.16	0.24	0.19	0.11	[5124 16524]
		1	0.00	0.00	0.00		[26982 7]
	92	0	0.16	0.24	0.19	0.11	[5124 16524]
		1	0.00	0.00	0.00		[26982 7]
	167	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
	208	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
	1942	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
2	1	0	1.00	0.77	0.87	0.90	[16561 5086]
		1	0.84	1.00	0.91		[17 26972]
	92	0	0.16	0.23	0.19	0.10	[5086 16561]
		1	0.00	0.00	0.00		[26972 17]
	167	0	0.16	0.23	0.19	0.10	[5086 16561]
		1	0.00	0.00	0.00		[26972 17]
	208	0	1.00	0.77	0.87	0.90	[16561 5086]
		1	0.84	1.00	0.91		[17 26972]
	1942	0	1.00	0.77	0.87	0.90	[16561 5086]
		1	0.84	1.00	0.91		[17 26972]
3	1	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
	92	0	1.00	0.76	0.86	0.89	[16374 5273]
		1	0.84	1.00	0.91		[10 26979]
	167	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
	208	0	1.00	0.76	0.86	0.89	[16374 5273]
		1	0.84	1.00	0.91		[10 26979]
	1942	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
4	1	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
	92	0	1.00	0.76	0.86	0.89	[16435 5212]
		1	0.84	1.00	0.91		[12 26977]
	167	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
	208	0	1.00	0.76	0.86	0.89	[16435 5212]
		1	0.84	1.00	0.91		[12 26977]
	1942	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
5	1	0	1.00	0.76	0.86	0.89	[16491 5157]
		1	0.84	1.00	0.91		[12 26976]
	92	0	0.16	0.24	0.19	0.11	[5157 16491]
		1	0.00	0.00	0.00		[26976 12]
	167	0	0.16	0.24	0.19	0.11	[5157 16491]
		1	0.00	0.00	0.00		[26976 12]
	208	0	1.00	0.76	0.86	0.89	[16491 5157]
		1	0.84	1.00	0.91		[12 26976]
	1942	0	0.16	0.24	0.19	0.11	[5157 16491]
		1	0.00	0.00	0.00		[26976 12]

Table B.144: Performance of *K-means* with the cic-ids2017_PS_a_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.16	0.24	0.19	0.11	[5124 16524]
		1	0.00	0.00	0.00		[26982 7]
	92	0	0.16	0.24	0.19	0.11	[5124 16524]
		1	0.00	0.00	0.00		[26982 7]
	167	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
	208	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
	1942	0	1.00	0.76	0.87	0.89	[16524 5124]
		1	0.84	1.00	0.91		[7 26982]
2	1	0	1.00	0.77	0.87	0.90	[16561 5086]
		1	0.84	1.00	0.91		[17 26972]
	92	0	0.16	0.23	0.19	0.10	[5086 16561]
		1	0.00	0.00	0.00		[26972 17]
	167	0	0.16	0.23	0.19	0.10	[5086 16561]
		1	0.00	0.00	0.00		[26972 17]
	208	0	0.16	0.23	0.19	0.10	[5086 16561]
		1	0.00	0.00	0.00		[26972 17]
	1942	0	1.00	0.77	0.87	0.90	[16561 5086]
		1	0.84	1.00	0.91		[17 26972]
3	1	0	1.00	0.76	0.86	0.89	[16374 5273]
		1	0.84	1.00	0.91		[10 26979]
	92	0	1.00	0.76	0.86	0.89	[16374 5273]
		1	0.84	1.00	0.91		[10 26979]
	167	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
	208	0	1.00	0.76	0.86	0.89	[16374 5273]
		1	0.84	1.00	0.91		[10 26979]
	1942	0	0.16	0.24	0.20	0.11	[5273 16374]
		1	0.00	0.00	0.00		[26979 10]
4	1	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
	92	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
	167	0	1.00	0.76	0.86	0.89	[16435 5212]
		1	0.84	1.00	0.91		[12 26977]
	208	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
	1942	0	0.16	0.24	0.19	0.11	[5212 16435]
		1	0.00	0.00	0.00		[26977 12]
5	1	0	1.00	0.76	0.86	0.89	[16491 5157]
		1	0.84	1.00	0.91		[12 26976]
	92	0	0.16	0.24	0.19	0.11	[5157 16491]
		1	0.00	0.00	0.00		[26976 12]
	167	0	0.16	0.24	0.19	0.11	[5157 16491]
		1	0.00	0.00	0.00		[26976 12]
	208	0	0.16	0.24	0.19	0.11	[5157 16491]
		1	0.00	0.00	0.00		[26976 12]
	1942	0	1.00	0.76	0.86	0.89	[16491 5157]
		1	0.84	1.00	0.91		[12 26976]

Table B.145: Performance of *K-means* with *Hold Out* section of the cic-ids2017_PS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.76	0.86	0.89	[14502 4553]
1	0.84	1.00	0.91		[10 23850]

Table B.146: Performance of *K-means* with the cic-ids2017_PS_b_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.16	0.12	0.14	0.32	[2674 18974]
		1	0.41	0.49	0.44		[13869 13120]
	92	0	0.16	0.12	0.14	0.32	[2674 18974]
		1	0.41	0.49	0.44		[13869 13120]
	167	0	0.59	0.88	0.71	0.68	[18973 2675]
		1	0.84	0.51	0.64		[13120 13869]
	208	0	0.59	0.88	0.71	0.68	[18973 2675]
		1	0.84	0.51	0.64		[13120 13869]
	1942	0	0.59	0.88	0.71	0.68	[18973 2675]
		1	0.84	0.51	0.64		[13120 13869]
2	1	0	0.59	0.88	0.70	0.67	[19008 2639]
		1	0.84	0.50	0.63		[13414 13575]
	92	0	0.16	0.12	0.14	0.33	[2639 19008]
		1	0.41	0.50	0.45		[13575 13414]
	167	0	0.59	0.88	0.70	0.67	[19017 2630]
		1	0.84	0.50	0.63		[13414 13575]
	208	0	0.16	0.12	0.14	0.33	[2630 19017]
		1	0.41	0.50	0.45		[13575 13414]
	1942	0	0.59	0.88	0.70	0.67	[19008 2639]
		1	0.84	0.50	0.63		[13414 13575]
3	1	0	0.16	0.12	0.14	0.33	[2699 18948]
		1	0.41	0.49	0.45		[13661 13328]
	92	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	167	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	208	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	1942	0	0.16	0.12	0.14	0.33	[2699 18948]
		1	0.41	0.49	0.45		[13661 13328]
4	1	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.59	0.88	0.70	0.67	[18978 2669]
		1	0.84	0.50	0.63		[13411 13578]
	167	0	0.16	0.12	0.14	0.33	[2669 18978]
		1	0.41	0.50	0.45		[13578 13411]
	208	0	0.59	0.88	0.70	0.67	[18978 2669]
		1	0.84	0.50	0.63		[13411 13578]
	1942	0	0.16	0.12	0.14	0.33	[2669 18978]
		1	0.41	0.50	0.45		[13578 13411]
5	1	0	0.59	0.88	0.70	0.67	[19018 2630]
		1	0.84	0.51	0.63		[13331 13657]
	92	0	0.59	0.88	0.70	0.67	[19018 2630]
		1	0.84	0.51	0.63		[13331 13657]
	167	0	0.16	0.12	0.14	0.33	[2630 19018]

Continues on next page

Table B.146 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.41	0.49	0.45		[13657 13331]
	208	0	0.16	0.12	0.14	0.33	[2630 19018]
		1	0.41	0.49	0.45		[13657 13331]
	1942	0	0.16	0.12	0.14	0.33	[2630 19018]
		1	0.41	0.49	0.45		[13657 13331]

Table B.147: Performance of *K-means* with the cic-ids2017_PS_b_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.16	0.12	0.14	0.32	[2676 18972]
		1	0.41	0.49	0.44		[13869 13120]
	92	0	0.16	0.12	0.14	0.32	[2676 18972]
		1	0.41	0.49	0.44		[13869 13120]
	167	0	0.59	0.88	0.71	0.68	[18972 2676]
		1	0.84	0.51	0.64		[13120 13869]
	208	0	0.59	0.88	0.71	0.68	[18972 2676]
		1	0.84	0.51	0.64		[13120 13869]
	1942	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
2	1	0	0.59	0.88	0.70	0.67	[19006 2641]
		1	0.84	0.50	0.63		[13414 13575]
	92	0	0.59	0.88	0.70	0.67	[19006 2641]
		1	0.84	0.50	0.63		[13414 13575]
	167	0	0.16	0.12	0.14	0.33	[2641 19006]
		1	0.41	0.50	0.45		[13575 13414]
	208	0	0.59	0.88	0.70	0.67	[19006 2641]
		1	0.84	0.50	0.63		[13414 13575]
	1942	0	0.59	0.88	0.70	0.67	[19006 2641]
		1	0.84	0.50	0.63		[13414 13575]
3	1	0	0.59	0.88	0.70	0.67	[18949 2698]
		1	0.84	0.51	0.63		[13328 13661]
	92	0	0.59	0.88	0.70	0.67	[18949 2698]
		1	0.84	0.51	0.63		[13328 13661]
	167	0	0.59	0.88	0.70	0.67	[18949 2698]
		1	0.84	0.51	0.63		[13328 13661]
	208	0	0.59	0.88	0.70	0.67	[18949 2698]
		1	0.84	0.51	0.63		[13328 13661]
	1942	0	0.59	0.88	0.70	0.67	[18949 2698]
		1	0.84	0.51	0.63		[13328 13661]
4	1	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.16	0.12	0.14	0.33	[2658 18989]
		1	0.41	0.50	0.45		[13578 13411]
	167	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	208	0	0.59	0.88	0.70	0.67	[18979 2668]
		1	0.84	0.50	0.63		[13411 13578]
	1942	0	0.16	0.12	0.14	0.33	[2668 18979]
		1	0.41	0.50	0.45		[13578 13411]
5	1	0	0.59	0.88	0.70	0.67	[19018 2630]
		1	0.84	0.51	0.63		[13331 13657]
	92	0	0.59	0.88	0.70	0.67	[19018 2630]
		1	0.84	0.51	0.63		[13331 13657]
	167	0	0.16	0.12	0.14	0.33	[2630 19018]

Continues on next page

Table B.147 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.41	0.49	0.45		[13657 13331]
	208	0	0.59	0.88	0.70	0.67	[19018 2630]
		1	0.84	0.51	0.63		[13331 13657]
	1942	0	0.59	0.88	0.70	0.67	[19018 2630]
		1	0.84	0.51	0.63		[13331 13657]

Table B.148: Performance of *K-means* with the cic-ids2017_PS_b_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.16	0.12	0.14	0.32	[2678 18970]
		1	0.41	0.49	0.44		[13869 13120]
	92	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
	167	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
	208	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
	1942	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
2	1	0	0.16	0.12	0.14	0.33	[2642 19005]
		1	0.41	0.50	0.45		[13575 13414]
	92	0	0.16	0.12	0.14	0.33	[2642 19005]
		1	0.41	0.50	0.45		[13575 13414]
	167	0	0.16	0.12	0.14	0.33	[2642 19005]
		1	0.41	0.50	0.45		[13575 13414]
	208	0	0.59	0.88	0.70	0.67	[19005 2642]
		1	0.84	0.50	0.63		[13414 13575]
	1942	0	0.59	0.88	0.70	0.67	[19005 2642]
		1	0.84	0.50	0.63		[13414 13575]
3	1	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	92	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	167	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	208	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	1942	0	0.16	0.12	0.14	0.33	[2699 18948]
		1	0.41	0.49	0.45		[13661 13328]
4	1	0	0.59	0.88	0.70	0.67	[18978 2669]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.59	0.88	0.70	0.67	[18978 2669]
		1	0.84	0.50	0.63		[13411 13578]
	167	0	0.16	0.12	0.14	0.33	[2669 18978]
		1	0.41	0.50	0.45		[13578 13411]
	208	0	0.59	0.88	0.70	0.67	[18978 2669]
		1	0.84	0.50	0.63		[13411 13578]
	1942	0	0.16	0.12	0.14	0.33	[2669 18978]
		1	0.41	0.50	0.45		[13578 13411]
5	1	0	0.59	0.88	0.70	0.67	[18995 2653]
		1	0.84	0.51	0.63		[13331 13657]
	92	0	0.59	0.88	0.70	0.67	[18995 2653]
		1	0.84	0.51	0.63		[13331 13657]
	167	0	0.16	0.12	0.14	0.33	[2653 18995]

Continues on next page

Table B.148 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.41	0.49	0.45		[13657 13331]
	208	0	0.59	0.88	0.70	0.67	[18995 2653]
		1	0.84	0.51	0.63		[13331 13657]
	1942	0	0.16	0.12	0.14	0.33	[2653 18995]
		1	0.41	0.49	0.45		[13657 13331]

Table B.149: Performance of *K-means* with *Hold Out* section of the cic-ids2017_PS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.58	0.88	0.70	0.67	[16694 2361]
1	0.84	0.50	0.63		[11875 11985]

Table B.150: Performance of *K-means* with the cic-ids2017_PS_c_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.16	0.12	0.14	0.32	[2674 18974]
		1	0.41	0.49	0.44		[13869 13120]
	92	0	0.16	0.12	0.14	0.32	[2674 18974]
		1	0.41	0.49	0.44		[13869 13120]
	167	0	0.16	0.12	0.14	0.32	[2674 18974]
		1	0.41	0.49	0.44		[13869 13120]
	208	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
	1942	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
2	1	0	0.16	0.12	0.14	0.33	[2631 19016]
		1	0.41	0.50	0.45		[13575 13414]
	92	0	0.16	0.12	0.14	0.33	[2630 19017]
		1	0.41	0.50	0.45		[13575 13414]
	167	0	0.16	0.12	0.14	0.33	[2631 19016]
		1	0.41	0.50	0.45		[13575 13414]
	208	0	0.16	0.12	0.14	0.33	[2630 19017]
		1	0.41	0.50	0.45		[13575 13414]
	1942	0	0.59	0.88	0.70	0.67	[19017 2630]
		1	0.84	0.50	0.63		[13414 13575]
3	1	0	0.16	0.12	0.14	0.33	[2699 18948]
		1	0.41	0.49	0.45		[13661 13328]
	92	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	167	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	208	0	0.16	0.12	0.14	0.33	[2699 18948]
		1	0.41	0.49	0.45		[13661 13328]
	1942	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
4	1	0	0.59	0.88	0.70	0.67	[18980 2667]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.59	0.88	0.70	0.67	[18980 2667]
		1	0.84	0.50	0.63		[13411 13578]
	167	0	0.59	0.88	0.70	0.67	[18989 2658]

Continues on next page

Table B.150 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.84	0.50	0.63		[13411 13578]
	208	0	0.16	0.12	0.14	0.33	[2658 18989]
		1	0.41	0.50	0.45		[13578 13411]
	1942	0	0.16	0.12	0.14	0.33	[2667 18980]
		1	0.41	0.50	0.45		[13578 13411]
5	1	0	0.59	0.88	0.70	0.67	[19019 2629]
		1	0.84	0.51	0.63		[13331 13657]
	92	0	0.59	0.88	0.70	0.67	[19018 2630]
		1	0.84	0.51	0.63		[13331 13657]
	167	0	0.16	0.12	0.14	0.33	[2630 19018]
		1	0.41	0.49	0.45		[13657 13331]
	208	0	0.16	0.12	0.14	0.33	[2630 19018]
		1	0.41	0.49	0.45		[13657 13331]
	1942	0	0.16	0.12	0.14	0.33	[2630 19018]
		1	0.41	0.49	0.45		[13657 13331]

Table B.151: Performance of *K-means* with the cic-ids2017_PS_c_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
	92	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
	167	0	0.16	0.12	0.14	0.32	[2674 18974]
		1	0.41	0.49	0.44		[13869 13120]
	208	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
	1942	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
2	1	0	0.59	0.88	0.70	0.67	[19007 2640]
		1	0.84	0.50	0.63		[13414 13575]
	92	0	0.59	0.88	0.70	0.67	[19007 2640]
		1	0.84	0.50	0.63		[13414 13575]
	167	0	0.59	0.88	0.70	0.67	[19007 2640]
		1	0.84	0.50	0.63		[13414 13575]
	208	0	0.59	0.88	0.70	0.67	[19017 2630]
		1	0.84	0.50	0.63		[13414 13575]
	1942	0	0.59	0.88	0.70	0.67	[19007 2640]
		1	0.84	0.50	0.63		[13414 13575]
3	1	0	0.16	0.12	0.14	0.33	[2685 18962]
		1	0.41	0.49	0.45		[13661 13328]
	92	0	0.59	0.88	0.70	0.67	[18962 2685]
		1	0.84	0.51	0.63		[13328 13661]
	167	0	0.59	0.88	0.70	0.67	[18962 2685]
		1	0.84	0.51	0.63		[13328 13661]
	208	0	0.16	0.12	0.14	0.33	[2685 18962]
		1	0.41	0.49	0.45		[13661 13328]
	1942	0	0.16	0.12	0.14	0.33	[2697 18950]
		1	0.41	0.49	0.45		[13661 13328]
4	1	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	167	0	0.59	0.88	0.70	0.67	[18989 2658]

Continues on next page

Table B.151 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.84	0.50	0.63		[13411 13578]
	208	0	0.16	0.12	0.14	0.33	[2658 18989]
		1	0.41	0.50	0.45		[13578 13411]
	1942	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
5	1	0	0.16	0.12	0.14	0.33	[2629 19019]
		1	0.41	0.49	0.45		[13657 13331]
	92	0	0.59	0.88	0.70	0.67	[19019 2629]
		1	0.84	0.51	0.63		[13331 13657]
	167	0	0.16	0.12	0.14	0.33	[2629 19019]
		1	0.41	0.49	0.45		[13657 13331]
	208	0	0.16	0.12	0.14	0.33	[2628 19020]
		1	0.41	0.49	0.45		[13657 13331]
	1942	0	0.16	0.12	0.14	0.33	[2629 19019]
		1	0.41	0.49	0.45		[13657 13331]

Table B.152: Performance of *K-means* with the cic-ids2017_PS_c_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
	92	0	0.16	0.12	0.14	0.32	[2678 18970]
		1	0.41	0.49	0.44		[13869 13120]
	167	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
	208	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
	1942	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
2	1	0	0.59	0.88	0.70	0.67	[19005 2642]
		1	0.84	0.50	0.63		[13414 13575]
	92	0	0.16	0.12	0.14	0.33	[2642 19005]
		1	0.41	0.50	0.45		[13575 13414]
	167	0	0.16	0.12	0.14	0.33	[2642 19005]
		1	0.41	0.50	0.45		[13575 13414]
	208	0	0.59	0.88	0.70	0.67	[19005 2642]
		1	0.84	0.50	0.63		[13414 13575]
	1942	0	0.59	0.88	0.70	0.67	[19005 2642]
		1	0.84	0.50	0.63		[13414 13575]
3	1	0	0.16	0.12	0.14	0.33	[2699 18948]
		1	0.41	0.49	0.45		[13661 13328]
	92	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	167	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	208	0	0.16	0.12	0.14	0.33	[2699 18948]
		1	0.41	0.49	0.45		[13661 13328]
	1942	0	0.16	0.12	0.14	0.33	[2699 18948]
		1	0.41	0.49	0.45		[13661 13328]
4	1	0	0.59	0.88	0.70	0.67	[18979 2668]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.59	0.88	0.70	0.67	[18979 2668]
		1	0.84	0.50	0.63		[13411 13578]
	167	0	0.16	0.12	0.14	0.33	[2668 18979]

Continues on next page

Table B.152 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.41	0.50	0.45		[13578 13411]
	208	0	0.59	0.88	0.70	0.67	[18979 2668]
		1	0.84	0.50	0.63		[13411 13578]
	1942	0	0.16	0.12	0.14	0.33	[2668 18979]
		1	0.41	0.50	0.45		[13578 13411]
5	1	0	0.59	0.88	0.70	0.67	[18995 2653]
		1	0.84	0.51	0.63		[13331 13657]
	92	0	0.59	0.88	0.70	0.67	[18995 2653]
		1	0.84	0.51	0.63		[13331 13657]
	167	0	0.16	0.12	0.14	0.33	[2653 18995]
		1	0.41	0.49	0.45		[13657 13331]
	208	0	0.59	0.88	0.70	0.67	[18995 2653]
		1	0.84	0.51	0.63		[13331 13657]
	1942	0	0.16	0.12	0.14	0.33	[2653 18995]
		1	0.41	0.49	0.45		[13657 13331]

Table B.153: Performance of *K-means* with *Hold Out* section of the cic-ids2017_PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.16	0.12	0.14	0.33	[2361 16694]
1	0.42	0.50	0.45		[11985 11875]

Table B.154: Performance of *K-means* with *Hold Out* section of the cic-ids2017_PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.58	0.88	0.70	0.67	[16694 2361]
1	0.84	0.50	0.63		[11875 11985]

Table B.155: Performance of *K-means* with the cic-ids2017_PS_d_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.59	0.88	0.71	0.68	[18975 2673]
		1	0.84	0.51	0.64		[13120 13869]
	92	0	0.16	0.12	0.14	0.32	[2673 18975]
		1	0.41	0.49	0.44		[13869 13120]
	167	0	0.59	0.88	0.71	0.68	[18975 2673]
		1	0.84	0.51	0.64		[13120 13869]
	208	0	0.59	0.88	0.71	0.68	[18975 2673]
		1	0.84	0.51	0.64		[13120 13869]
	1942	0	0.59	0.88	0.71	0.68	[18975 2673]
		1	0.84	0.51	0.64		[13120 13869]
2	1	0	0.16	0.12	0.14	0.33	[2629 19018]
		1	0.41	0.50	0.45		[13575 13414]
	92	0	0.59	0.88	0.70	0.67	[19018 2629]
		1	0.84	0.50	0.63		[13414 13575]
	167	0	0.59	0.88	0.70	0.67	[19018 2629]
		1	0.84	0.50	0.63		[13414 13575]

Continues on next page

Table B.155 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	208	0	0.16	0.12	0.14	0.33	[2629 19018]
		1	0.41	0.50	0.45		[13575 13414]
	1942	0	0.59	0.88	0.70	0.67	[19017 2630]
		1	0.84	0.50	0.63		[13414 13575]
3	1	0	0.16	0.12	0.14	0.33	[2685 18962]
		1	0.41	0.49	0.45		[13661 13328]
	92	0	0.59	0.88	0.70	0.67	[18950 2697]
		1	0.84	0.51	0.63		[13328 13661]
	167	0	0.16	0.12	0.14	0.33	[2685 18962]
		1	0.41	0.49	0.45		[13661 13328]
	208	0	0.16	0.12	0.14	0.33	[2697 18950]
		1	0.41	0.49	0.45		[13661 13328]
	1942	0	0.16	0.12	0.14	0.33	[2697 18950]
		1	0.41	0.49	0.45		[13661 13328]
4	1	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	167	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	208	0	0.16	0.12	0.14	0.33	[2658 18989]
		1	0.41	0.50	0.45		[13578 13411]
	1942	0	0.16	0.12	0.14	0.33	[2658 18989]
		1	0.41	0.50	0.45		[13578 13411]
5	1	0	0.59	0.88	0.70	0.67	[19020 2628]
		1	0.84	0.51	0.63		[13331 13657]
	92	0	0.59	0.88	0.70	0.67	[19020 2628]
		1	0.84	0.51	0.63		[13331 13657]
	167	0	0.16	0.12	0.14	0.33	[2628 19020]
		1	0.41	0.49	0.45		[13657 13331]
	208	0	0.59	0.88	0.70	0.67	[19020 2628]
		1	0.84	0.51	0.63		[13331 13657]
	1942	0	0.16	0.12	0.14	0.33	[2628 19020]
		1	0.41	0.49	0.45		[13657 13331]

Table B.156: Performance of *K-means* with the cic-ids2017_PS_d_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
	92	0	0.16	0.12	0.14	0.32	[2674 18974]
		1	0.41	0.49	0.44		[13869 13120]
	167	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
	208	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
	1942	0	0.59	0.88	0.71	0.68	[18974 2674]
		1	0.84	0.51	0.64		[13120 13869]
2	1	0	0.16	0.12	0.14	0.33	[2630 19017]
		1	0.41	0.50	0.45		[13575 13414]
	92	0	0.16	0.12	0.14	0.33	[2631 19016]
		1	0.41	0.50	0.45		[13575 13414]
	167	0	0.59	0.88	0.70	0.67	[19017 2630]
		1	0.84	0.50	0.63		[13414 13575]

Continues on next page

Table B.156 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	208	0	0.59	0.88	0.70	0.67	[19016 2631]
		1	0.84	0.50	0.63		[13414 13575]
	1942	0	0.59	0.88	0.70	0.67	[19016 2631]
		1	0.84	0.50	0.63		[13414 13575]
3	1	0	0.59	0.88	0.70	0.67	[18964 2683]
		1	0.84	0.51	0.63		[13328 13661]
	92	0	0.59	0.88	0.70	0.67	[18964 2683]
		1	0.84	0.51	0.63		[13328 13661]
	167	0	0.59	0.88	0.70	0.67	[18964 2683]
		1	0.84	0.51	0.63		[13328 13661]
	208	0	0.59	0.88	0.70	0.67	[18964 2683]
		1	0.84	0.51	0.63		[13328 13661]
	1942	0	0.16	0.12	0.14	0.33	[2683 18964]
		1	0.41	0.49	0.45		[13661 13328]
4	1	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	167	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	208	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
	1942	0	0.59	0.88	0.70	0.67	[18989 2658]
		1	0.84	0.50	0.63		[13411 13578]
5	1	0	0.59	0.88	0.70	0.67	[19020 2628]
		1	0.84	0.51	0.63		[13331 13657]
	92	0	0.59	0.88	0.70	0.67	[19020 2628]
		1	0.84	0.51	0.63		[13331 13657]
	167	0	0.59	0.88	0.70	0.67	[19020 2628]
		1	0.84	0.51	0.63		[13331 13657]
	208	0	0.16	0.12	0.14	0.33	[2628 19020]
		1	0.41	0.49	0.45		[13657 13331]
	1942	0	0.16	0.12	0.14	0.33	[2628 19020]
		1	0.41	0.49	0.45		[13657 13331]

Table B.157: Performance of *K-means* with the cic-ids2017_PS_d_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
	92	0	0.16	0.12	0.14	0.32	[2678 18970]
		1	0.41	0.49	0.44		[13869 13120]
	167	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
	208	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
	1942	0	0.59	0.88	0.71	0.68	[18970 2678]
		1	0.84	0.51	0.64		[13120 13869]
2	1	0	0.59	0.88	0.70	0.67	[19003 2644]
		1	0.84	0.50	0.63		[13414 13575]
	92	0	0.16	0.12	0.14	0.33	[2644 19003]
		1	0.41	0.50	0.45		[13575 13414]
	167	0	0.16	0.12	0.14	0.33	[2644 19003]
		1	0.41	0.50	0.45		[13575 13414]

Continues on next page

Table B.157 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	208	0	0.59	0.88	0.70	0.67	[19003 2644]
		1	0.84	0.50	0.63		[13414 13575]
	1942	0	0.59	0.88	0.70	0.67	[19003 2644]
		1	0.84	0.50	0.63		[13414 13575]
3	1	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	92	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	167	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	208	0	0.59	0.88	0.70	0.67	[18948 2699]
		1	0.84	0.51	0.63		[13328 13661]
	1942	0	0.16	0.12	0.14	0.33	[2699 18948]
		1	0.41	0.49	0.45		[13661 13328]
4	1	0	0.59	0.88	0.70	0.67	[18979 2668]
		1	0.84	0.50	0.63		[13411 13578]
	92	0	0.59	0.88	0.70	0.67	[18979 2668]
		1	0.84	0.50	0.63		[13411 13578]
	167	0	0.16	0.12	0.14	0.33	[2668 18979]
		1	0.41	0.50	0.45		[13578 13411]
	208	0	0.16	0.12	0.14	0.33	[2668 18979]
		1	0.41	0.50	0.45		[13578 13411]
	1942	0	0.16	0.12	0.14	0.33	[2668 18979]
		1	0.41	0.50	0.45		[13578 13411]
5	1	0	0.59	0.88	0.70	0.67	[18995 2653]
		1	0.84	0.51	0.63		[13331 13657]
	92	0	0.59	0.88	0.70	0.67	[18995 2653]
		1	0.84	0.51	0.63		[13331 13657]
	167	0	0.16	0.12	0.14	0.33	[2653 18995]
		1	0.41	0.49	0.45		[13657 13331]
	208	0	0.59	0.88	0.70	0.67	[18995 2653]
		1	0.84	0.51	0.63		[13331 13657]
	1942	0	0.16	0.12	0.14	0.33	[2653 18995]
		1	0.41	0.49	0.45		[13657 13331]

Table B.158: Performance of *K-means* with *Hold Out* section of the cic-ids2017_PS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.58	0.88	0.70	0.67	[16694 2361]
1	0.84	0.50	0.63		[11875 11985]

Table B.159: Performance of *K-means* with the cic-ids2017_PS_e_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.42	0.90	0.57	0.40	[19549 2099]
		1	0.00	0.00	0.00		[26988 1]
	92	0	1.00	0.10	0.18	0.60	[2099 19549]
		1	0.58	1.00	0.73		[1 26988]
	167	0	1.00	0.10	0.18	0.60	[2099 19549]
		1	0.58	1.00	0.73		[1 26988]

Continues on next page

Table B.159 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	208	0	0.42	0.90	0.57	0.40	[19549 2099]
		1	0.00	0.00	0.00		[26988 1]
	1942	0	0.42	0.90	0.57	0.40	[19549 2099]
		1	0.00	0.00	0.00		[26988 1]
2	1	0	0.42	0.90	0.57	0.40	[19431 2216]
		1	0.00	0.00	0.00		[26987 2]
	92	0	0.42	0.90	0.57	0.40	[19431 2216]
		1	0.00	0.00	0.00		[26987 2]
	167	0	0.42	0.90	0.57	0.40	[19431 2216]
		1	0.00	0.00	0.00		[26987 2]
	208	0	0.42	0.90	0.57	0.40	[19431 2216]
		1	0.00	0.00	0.00		[26987 2]
	1942	0	0.42	0.90	0.57	0.40	[19431 2216]
		1	0.00	0.00	0.00		[26987 2]
3	1	0	0.42	0.90	0.57	0.40	[19472 2175]
		1	0.00	0.00	0.00		[26989 0]
	92	0	0.42	0.90	0.57	0.40	[19472 2175]
		1	0.00	0.00	0.00		[26989 0]
	167	0	0.42	0.90	0.57	0.40	[19472 2175]
		1	0.00	0.00	0.00		[26989 0]
	208	0	0.42	0.90	0.57	0.40	[19472 2175]
		1	0.00	0.00	0.00		[26989 0]
	1942	0	1.00	0.10	0.18	0.60	[2175 19472]
		1	0.58	1.00	0.73		[0 26989]
4	1	0	0.42	0.90	0.57	0.40	[19461 2186]
		1	0.00	0.00	0.00		[26988 1]
	92	0	1.00	0.10	0.18	0.60	[2186 19461]
		1	0.58	1.00	0.73		[1 26988]
	167	0	1.00	0.10	0.18	0.60	[2186 19461]
		1	0.58	1.00	0.73		[1 26988]
	208	0	0.42	0.90	0.57	0.40	[19461 2186]
		1	0.00	0.00	0.00		[26988 1]
	1942	0	0.42	0.90	0.57	0.40	[19461 2186]
		1	0.00	0.00	0.00		[26988 1]
5	1	0	1.00	0.10	0.18	0.60	[2183 19465]
		1	0.58	1.00	0.73		[0 26988]
	92	0	0.42	0.90	0.57	0.40	[19465 2183]
		1	0.00	0.00	0.00		[26988 0]
	167	0	0.42	0.90	0.57	0.40	[19465 2183]
		1	0.00	0.00	0.00		[26988 0]
	208	0	0.42	0.90	0.57	0.40	[19465 2183]
		1	0.00	0.00	0.00		[26988 0]
	1942	0	0.42	0.90	0.57	0.40	[19465 2183]
		1	0.00	0.00	0.00		[26988 0]

Table B.160: Performance of *K-means* with the cic-ids2017_PS_e_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.42	0.90	0.57	0.40	[19549 2099]
		1	0.00	0.00	0.00		[26988 1]
	92	0	0.42	0.90	0.57	0.40	[19549 2099]
		1	0.00	0.00	0.00		[26988 1]
	167	0	1.00	0.10	0.18	0.60	[2099 19549]
		1	0.58	1.00	0.73		[1 26988]

Continues on next page

Table B.160 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	208	0	0.42	0.90	0.57	0.40	[19549 2099]
		1	0.00	0.00	0.00		[26988 1]
	1942	0	0.42	0.90	0.57	0.40	[19549 2099]
		1	0.00	0.00	0.00		[26988 1]
2	1	0	0.42	0.90	0.57	0.40	[19431 2216]
		1	0.00	0.00	0.00		[26987 2]
	92	0	0.42	0.90	0.57	0.40	[19431 2216]
		1	0.00	0.00	0.00		[26987 2]
	167	0	0.58	0.87	0.70	0.66	[18753 2894]
		1	0.82	0.50	0.62		[13414 13575]
	208	0	0.42	0.90	0.57	0.40	[19431 2216]
		1	0.00	0.00	0.00		[26987 2]
	1942	0	0.58	0.87	0.70	0.66	[18753 2894]
		1	0.82	0.50	0.62		[13414 13575]
3	1	0	0.42	0.90	0.57	0.40	[19472 2175]
		1	0.00	0.00	0.00		[26989 0]
	92	0	0.42	0.90	0.57	0.40	[19472 2175]
		1	0.00	0.00	0.00		[26989 0]
	167	0	1.00	0.10	0.18	0.60	[2175 19472]
		1	0.58	1.00	0.73		[0 26989]
	208	0	0.42	0.90	0.57	0.40	[19472 2175]
		1	0.00	0.00	0.00		[26989 0]
	1942	0	1.00	0.10	0.18	0.60	[2175 19472]
		1	0.58	1.00	0.73		[0 26989]
4	1	0	0.42	0.90	0.57	0.40	[19461 2186]
		1	0.00	0.00	0.00		[26988 1]
	92	0	0.42	0.90	0.57	0.40	[19461 2186]
		1	0.00	0.00	0.00		[26988 1]
	167	0	1.00	0.10	0.18	0.60	[2186 19461]
		1	0.58	1.00	0.73		[1 26988]
	208	0	0.42	0.90	0.57	0.40	[19461 2186]
		1	0.00	0.00	0.00		[26988 1]
	1942	0	0.42	0.90	0.57	0.40	[19461 2186]
		1	0.00	0.00	0.00		[26988 1]
5	1	0	1.00	0.10	0.18	0.60	[2183 19465]
		1	0.58	1.00	0.73		[0 26988]
	92	0	0.42	0.90	0.57	0.40	[19465 2183]
		1	0.00	0.00	0.00		[26988 0]
	167	0	0.42	0.90	0.57	0.40	[19465 2183]
		1	0.00	0.00	0.00		[26988 0]
	208	0	0.42	0.90	0.57	0.40	[19465 2183]
		1	0.00	0.00	0.00		[26988 0]
	1942	0	0.42	0.90	0.57	0.40	[19465 2183]
		1	0.00	0.00	0.00		[26988 0]

Table B.161: Performance of *K-means* with the cic-ids2017_PS_e_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.59	0.86	0.70	0.67	[18688 2960]
		1	0.82	0.51	0.63		[13120 13869]
	92	0	0.18	0.14	0.15	0.33	[2960 18688]
		1	0.41	0.49	0.45		[13869 13120]
	167	0	0.59	0.86	0.70	0.67	[18688 2960]
		1	0.82	0.51	0.63		[13120 13869]

Continues on next page

Table B.161 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	208	0	0.59	0.86	0.70	0.67	[18688 2960]
		1	0.82	0.51	0.63		[13120 13869]
	1942	0	0.59	0.86	0.70	0.67	[18688 2960]
		1	0.82	0.51	0.63		[13120 13869]
2	1	0	0.58	0.87	0.70	0.66	[18753 2894]
		1	0.82	0.50	0.62		[13414 13575]
	92	0	0.18	0.13	0.15	0.34	[2894 18753]
		1	0.42	0.50	0.45		[13575 13414]
	167	0	0.58	0.87	0.70	0.66	[18753 2894]
		1	0.82	0.50	0.62		[13414 13575]
	208	0	0.58	0.87	0.70	0.66	[18753 2894]
		1	0.82	0.50	0.62		[13414 13575]
	1942	0	0.58	0.87	0.70	0.66	[18753 2894]
		1	0.82	0.50	0.62		[13414 13575]
3	1	0	0.18	0.14	0.16	0.34	[2971 18676]
		1	0.42	0.49	0.45		[13661 13328]
	92	0	0.58	0.86	0.70	0.66	[18676 2971]
		1	0.82	0.51	0.63		[13328 13661]
	167	0	0.58	0.86	0.70	0.66	[18676 2971]
		1	0.82	0.51	0.63		[13328 13661]
	208	0	0.18	0.14	0.16	0.34	[2971 18676]
		1	0.42	0.49	0.45		[13661 13328]
	1942	0	0.18	0.14	0.16	0.34	[2971 18676]
		1	0.42	0.49	0.45		[13661 13328]
4	1	0	0.58	0.86	0.70	0.66	[18712 2935]
		1	0.82	0.50	0.62		[13411 13578]
	92	0	0.18	0.14	0.15	0.34	[2935 18712]
		1	0.42	0.50	0.45		[13578 13411]
	167	0	0.18	0.14	0.15	0.34	[2935 18712]
		1	0.42	0.50	0.45		[13578 13411]
	208	0	0.58	0.86	0.70	0.66	[18712 2935]
		1	0.82	0.50	0.62		[13411 13578]
	1942	0	0.18	0.14	0.15	0.34	[2935 18712]
		1	0.42	0.50	0.45		[13578 13411]
5	1	0	0.58	0.86	0.70	0.67	[18713 2935]
		1	0.82	0.51	0.63		[13331 13657]
	92	0	0.18	0.14	0.15	0.33	[2935 18713]
		1	0.42	0.49	0.45		[13657 13331]
	167	0	0.18	0.14	0.15	0.33	[2935 18713]
		1	0.42	0.49	0.45		[13657 13331]
	208	0	0.18	0.14	0.15	0.33	[2935 18713]
		1	0.42	0.49	0.45		[13657 13331]
	1942	0	0.18	0.14	0.15	0.33	[2935 18713]
		1	0.42	0.49	0.45		[13657 13331]

Table B.162: Performance of *K-means* with *Hold Out* section of the cic-ids2017_PS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.58	0.86	0.69	0.66	[16440 2615]
1	0.82	0.50	0.62		[11875 11985]

B.2.2 Results from the collected dataset(s)

Results for the *DoS* attack

Table B.163: Performance of *K-means* with the DoS_a_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
	208	0	1.00	0.01	0.02	0.90	[36 3010]
		1	0.90	1.00	0.94		[0 25667]
	1942	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]

Continues on next page

Table B.163 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3001 44] [25667 0]

Table B.164: Performance of *K-means* with the DoS_a_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	92	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	167	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	208	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	1942	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
2	1	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	92	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	167	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	208	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	1942	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
3	1	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2989 56] [25668 0]
	92	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2989 56] [25668 0]
	167	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2989 56] [25668 0]
	208	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2989 56] [25668 0]
	1942	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2989 56] [25668 0]
4	1	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
	92	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
	167	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
	208	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
	1942	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
5	1	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3001 44] [25667 0]
	92	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3001 44] [25667 0]
	167	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3001 44] [25667 0]
	208	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3001 44] [25667 0]

Continues on next page

Table B.164 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]

Table B.165: Performance of *K-means* with the DoS_a_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]

Continues on next page

Table B.165 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3002 43] [25667 0]

Table B.166: Performance of *K-means* with *Hold Out* section of the DoS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[2645 40] [22650 0]

Table B.167: Performance of *K-means* with the DoS_b_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	92	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	167	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	208	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	1942	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
2	1	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	92	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	167	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	208	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	1942	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
3	1	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2990 55] [25668 0]
	92	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2990 55] [25668 0]
	167	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2990 55] [25668 0]
	208	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2990 55] [25668 0]
	1942	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2990 55] [25668 0]
4	1	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3010 36] [25667 0]
	92	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3010 36] [25667 0]
	167	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3010 36] [25667 0]
	208	0 1	1.00 0.90	0.01 1.00	0.02 0.94	0.90	[36 3010] [0 25667]

Continues on next page

Table B.167 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]

Table B.168: Performance of *K-means* with the DoS_b_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]

Continues on next page

Table B.168 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]

Table B.169: Performance of *K-means* with the DoS_b_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]

Continues on next page

Table B.169 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]

Table B.170: Performance of *K-means* with *Hold Out* section of the DoS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.10	0.99	0.19	0.10	[2645 40]
1	0.00	0.00	0.00		[22650 0]

Table B.171: Performance of *K-means* with the DoS_c_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]

Continues on next page

Table B.171 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
	208	0	1.00	0.01	0.02	0.90	[36 3010]
		1	0.90	1.00	0.94		[0 25667]
	1942	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]

Table B.172: Performance of *K-means* with the DoS_c_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]

Continues on next page

Table B.172 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]

Table B.173: Performance of *K-means* with the DoS_c_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]

Continues on next page

Table B.173 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2990 55] [25668 0]
4	1	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
	92	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
	167	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
	208	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
	1942	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3009 37] [25667 0]
5	1	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3002 43] [25667 0]
	92	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3002 43] [25667 0]
	167	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3002 43] [25667 0]
	208	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3002 43] [25667 0]
	1942	0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[3002 43] [25667 0]

Table B.174: Performance of *K-means* with *Hold Out* section of the DoS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0 1	0.10 0.00	0.99 0.00	0.19 0.00	0.10	[2645 40] [22650 0]

Table B.175: Performance of *K-means* with the DoS_d_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	92	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	167	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	208	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
	1942	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2999 46] [25668 0]
2	1	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	92	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	167	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]
	208	0 1	0.11 0.00	0.99 0.00	0.19 0.00	0.10	[3012 33] [25668 0]

Continues on next page

Table B.175 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
	208	0	1.00	0.01	0.02	0.90	[36 3010]
		1	0.90	1.00	0.94		[0 25667]
	1942	0	0.10	0.99	0.19	0.10	[3010 36]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]

Table B.176: Performance of *K-means* with the DoS_d_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]

Continues on next page

Table B.176 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2989 56]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25667 0]

Table B.177: Performance of *K-means* with the DoS_d_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2999 46]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]

Continues on next page

Table B.177 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.98	0.19	0.10	[2990 55]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3009 37]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]

Table B.178: Performance of *K-means* with *Hold Out* section of the DoS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.10	0.99	0.19	0.10	[2645 40]
1	0.00	0.00	0.00		[22650 0]

Table B.179: Performance of *K-means* with the DoS_e_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.99	0.19	0.10	[3000 45]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.99	0.19	0.10	[3000 45]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.99	0.19	0.10	[3000 45]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.99	0.19	0.10	[3000 45]
		1	0.00	0.00	0.00		[25668 0]

Continues on next page

Table B.179 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.10	0.99	0.19	0.10	[3000 45]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.99	0.19	0.10	[3006 39]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.99	0.19	0.10	[3006 39]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.99	0.19	0.10	[3006 39]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.99	0.19	0.10	[3006 39]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.99	0.19	0.10	[3006 39]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.11	0.99	0.19	0.10	[3012 34]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.11	0.99	0.19	0.10	[3012 34]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.11	0.99	0.19	0.10	[3012 34]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.11	0.99	0.19	0.10	[3012 34]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 34]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3002 43]
		1	0.00	0.00	0.00		[25667 0]

Table B.180: Performance of *K-means* with the DoS_e_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]

Continues on next page

Table B.180 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.10	[3012 33]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]

Table B.181: Performance of *K-means* with the DoS_e_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]

Continues on next page

Table B.181 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
	1942	0	0.10	0.99	0.19	0.10	[3001 44]
		1	0.00	0.00	0.00		[25668 0]
2	1	0	0.11	0.99	0.19	0.11	[3018 27]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.11	0.99	0.19	0.11	[3018 27]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.11	0.99	0.19	0.11	[3018 27]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.11	0.99	0.19	0.11	[3018 27]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.11	0.99	0.19	0.11	[3018 27]
		1	0.00	0.00	0.00		[25668 0]
3	1	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
	92	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
	167	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
	208	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
	1942	0	0.10	0.99	0.19	0.10	[3005 40]
		1	0.00	0.00	0.00		[25668 0]
4	1	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3011 35]
		1	0.00	0.00	0.00		[25667 0]
5	1	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]
	92	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]
	167	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]
	208	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]
	1942	0	0.10	0.99	0.19	0.10	[3011 34]
		1	0.00	0.00	0.00		[25667 0]

Table B.182: Performance of *K-means* with *Hold Out* section of the DoS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.10	0.99	0.19	0.10	[2652 33]
1	0.00	0.00	0.00		[22650 0]

Results for the *PS* attack

Table B.183: Performance of *K-means* with the PS_a_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.184: Performance of *K-means* with the PS_a_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.185: Performance of *K-means* with the PS_a_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.186: Performance of *K-means* with *Hold Out* section of the PS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.07	0.98	0.13	0.07	[1173 24]
1	0.00	0.00	0.00		[15311 0]

Table B.187: Performance of *K-means* with the PS_b_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]

Continues on next page

Table B.187 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.188: Performance of *K-means* with the PS_b_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]

Continues on next page

Table B.188 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.189: Performance of *K-means* with the PS_b_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]

Continues on next page

Table B.189 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.190: Performance of *K-means* with *Hold Out* section of the PS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.07	0.98	0.13	0.07	[1173 24]
1	0.00	0.00	0.00		[15311 0]

Table B.191: Performance of *K-means* with the PS_c_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]

Continues on next page

Table B.191 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.192: Performance of *K-means* with the PS_c_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]

Continues on next page

Table B.192 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.193: Performance of *K-means* with the PS_c_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]

Continues on next page

Table B.193 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.194: Performance of *K-means* with *Hold Out* section of the PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.07	0.98	0.13	0.07	[1173 24]
1	0.00	0.00	0.00		[15311 0]

Table B.195: Performance of *K-means* with the PS_d_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]

Continues on next page

Table B.195 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.196: Performance of *K-means* with the PS_d_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]

Continues on next page

Table B.196 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.197: Performance of *K-means* with the PS_d_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]

Continues on next page

Table B.197 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.198: Performance of *K-means* with *Hold Out* section of the PS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.07	0.98	0.13	0.07	[1173 24]
1	0.00	0.00	0.00		[15311 0]

Table B.199: Performance of *K-means* with the PS_e_i dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]

Continues on next page

Table B.199 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.200: Performance of *K-means* with the PS_e_ii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]

Continues on next page

Table B.200 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.201: Performance of *K-means* with the PS_e_iii dataset

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	1	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1393 18]
		1	0.00	0.00	0.00		[17298 0]
2	1	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1392 19]

Continues on next page

Table B.201 – Continuation

K	Seed	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1392 19]
		1	0.00	0.00	0.00		[17298 0]
3	1	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 21]
		1	0.00	0.00	0.00		[17298 0]
4	1	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.98	0.14	0.07	[1384 26]
		1	0.00	0.00	0.00		[17298 0]
5	1	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	92	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	167	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	208	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]
	1942	0	0.07	0.99	0.14	0.07	[1390 20]
		1	0.00	0.00	0.00		[17298 0]

Table B.202: Performance of *K-means* with *Hold Out* section of the PS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.07	0.98	0.13	0.07	[1173 24]
1	0.00	0.00	0.00		[15311 0]

B.3 SVM

B.3.1 Search Grid Parameters

Table B.203: *SVM* search grid with the cic-ids2017_DoS_a dataset

C	γ	kernel	K=1	K=2	K=3	K=4	K=5
1.00e-04	1.00e-05	linear	0.943	0.944	0.946	0.945	0.944
	1.00e-05	rbf	0.567	0.567	0.567	0.567	0.567
	2.15e-04	linear	0.943	0.944	0.946	0.945	0.944
	2.15e-04	rbf	0.567	0.567	0.567	0.567	0.567
	4.64e-03	linear	0.943	0.944	0.946	0.945	0.944
	4.64e-03	rbf	0.567	0.567	0.567	0.567	0.567
	1.00e-01	linear	0.943	0.944	0.946	0.945	0.944
	1.00e-01	rbf	0.857	0.859	0.862	0.861	0.858
2.15e-02	1.00e-05	linear	0.977	0.978	0.979	0.978	0.978
	1.00e-05	rbf	0.567	0.567	0.567	0.567	0.567
	2.15e-04	linear	0.977	0.978	0.979	0.978	0.978
	2.15e-04	rbf	0.581	0.581	0.581	0.582	0.582
	4.64e-03	linear	0.977	0.978	0.979	0.978	0.978
	4.64e-03	rbf	0.954	0.955	0.957	0.957	0.955
	1.00e-01	linear	0.977	0.978	0.979	0.978	0.978
	1.00e-01	rbf	0.971	0.971	0.972	0.973	0.972
4.64e+00	1.00e-05	linear	0.985	0.985	0.986	0.985	0.985
	1.00e-05	rbf	0.944	0.944	0.947	0.945	0.944
	2.15e-04	linear	0.985	0.985	0.986	0.985	0.985
	2.15e-04	rbf	0.968	0.969	0.970	0.970	0.969
	4.64e-03	linear	0.985	0.985	0.986	0.985	0.985
	4.64e-03	rbf	0.981	0.982	0.982	0.982	0.982
	1.00e-01	linear	0.985	0.985	0.986	0.985	0.985
	1.00e-01	rbf	0.983	0.984	0.984	0.984	0.983
1.00e+03	1.00e-05	linear	0.999	0.999	0.999	0.999	0.999
	1.00e-05	rbf	0.977	0.977	0.979	0.978	0.977
	2.15e-04	linear	0.999	0.999	0.999	0.999	0.999
	2.15e-04	rbf	0.982	0.983	0.984	0.983	0.983
	4.64e-03	linear	0.999	0.999	0.999	0.999	0.999
	4.64e-03	rbf	0.987	0.987	0.987	0.987	0.987
	1.00e-01	linear	0.999	0.999	0.999	0.999	0.999
	1.00e-01	rbf	0.999	0.999	0.999	0.999	0.999

Table B.204: *SVM* search grid with the DoS_a dataset

C	γ	kernel	K=1	K=2	K=3	K=4	K=5
1.00e-02	1.00e-09	linear	1.000	0.999	1.000	1.000	0.999
	1.00e-09	rbf	0.894	0.894	0.894	0.894	0.894
	2.51e-07	linear	1.000	0.999	1.000	1.000	0.999
	2.51e-07	rbf	0.894	0.894	0.894	0.894	0.894
	6.31e-05	linear	1.000	0.999	1.000	1.000	0.999
	6.31e-05	rbf	0.988	0.989	0.989	0.989	0.988
	1.58e-02	linear	1.000	0.999	1.000	1.000	0.999
	1.58e-02	rbf	1.000	0.999	0.999	1.000	1.000
	3.98e+00	linear	1.000	0.999	1.000	1.000	0.999
	3.98e+00	rbf	0.998	0.998	0.998	0.998	0.999

Continues on next page

Table B.204 – Continuation

C	γ	kernel	K=1	K=2	K=3	K=4	K=5
	1.00e+03	linear	1.000	0.999	1.000	1.000	0.999
	1.00e+03	rbf	0.953	0.953	0.956	0.954	0.954
2.51e+00	1.00e-09	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-09	rbf	0.894	0.894	0.894	0.894	0.894
	2.51e-07	linear	1.000	1.000	1.000	1.000	1.000
	2.51e-07	rbf	0.921	0.920	0.916	0.921	0.919
	6.31e-05	linear	1.000	1.000	1.000	1.000	1.000
	6.31e-05	rbf	0.999	0.999	0.999	0.999	0.999
	1.58e-02	linear	1.000	1.000	1.000	1.000	1.000
	1.58e-02	rbf	1.000	1.000	1.000	1.000	1.000
	3.98e+00	linear	1.000	1.000	1.000	1.000	1.000
	3.98e+00	rbf	1.000	1.000	1.000	1.000	1.000
	1.00e+03	linear	1.000	1.000	1.000	1.000	1.000
	1.00e+03	rbf	0.999	0.998	0.999	0.998	0.999
6.31e+02	1.00e-09	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-09	rbf	0.895	0.895	0.895	0.895	0.895
	2.51e-07	linear	1.000	1.000	1.000	1.000	1.000
	2.51e-07	rbf	0.999	0.998	0.999	0.999	0.999
	6.31e-05	linear	1.000	1.000	1.000	1.000	1.000
	6.31e-05	rbf	1.000	1.000	1.000	1.000	1.000
	1.58e-02	linear	1.000	1.000	1.000	1.000	1.000
	1.58e-02	rbf	1.000	1.000	1.000	1.000	1.000
	3.98e+00	linear	1.000	1.000	1.000	1.000	1.000
	3.98e+00	rbf	1.000	1.000	1.000	1.000	1.000
	1.00e+03	linear	1.000	1.000	1.000	1.000	1.000
	1.00e+03	rbf	0.999	0.998	0.999	0.998	0.999
1.58e+05	1.00e-09	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-09	rbf	0.924	0.924	0.923	0.924	0.923
	2.51e-07	linear	1.000	1.000	1.000	1.000	1.000
	2.51e-07	rbf	1.000	1.000	1.000	1.000	1.000
	6.31e-05	linear	1.000	1.000	1.000	1.000	1.000
	6.31e-05	rbf	1.000	1.000	1.000	1.000	1.000
	1.58e-02	linear	1.000	1.000	1.000	1.000	1.000
	1.58e-02	rbf	1.000	1.000	1.000	1.000	1.000
	3.98e+00	linear	1.000	1.000	1.000	1.000	1.000
	3.98e+00	rbf	1.000	1.000	1.000	1.000	1.000
	1.00e+03	linear	1.000	1.000	1.000	1.000	1.000
	1.00e+03	rbf	0.999	0.998	0.999	0.998	0.999
3.98e+07	1.00e-09	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-09	rbf	0.924	0.924	0.923	0.924	0.923
	2.51e-07	linear	1.000	1.000	1.000	1.000	1.000
	2.51e-07	rbf	1.000	1.000	1.000	1.000	1.000
	6.31e-05	linear	1.000	1.000	1.000	1.000	1.000
	6.31e-05	rbf	1.000	1.000	1.000	1.000	1.000
	1.58e-02	linear	1.000	1.000	1.000	1.000	1.000
	1.58e-02	rbf	1.000	1.000	1.000	1.000	1.000
	3.98e+00	linear	1.000	1.000	1.000	1.000	1.000
	3.98e+00	rbf	1.000	1.000	1.000	1.000	1.000
	1.00e+03	linear	1.000	1.000	1.000	1.000	1.000
	1.00e+03	rbf	0.999	0.998	0.999	0.998	0.999
1.00e+10	1.00e-09	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-09	rbf	0.924	0.924	0.923	0.924	0.923
	2.51e-07	linear	1.000	1.000	1.000	1.000	1.000
	2.51e-07	rbf	1.000	1.000	1.000	1.000	1.000
	6.31e-05	linear	1.000	1.000	1.000	1.000	1.000

Continues on next page

Table B.204 – Continuation

C	γ	kernel	K=1	K=2	K=3	K=4	K=5
	6.31e-05	rbf	1.000	1.000	1.000	1.000	1.000
	1.58e-02	linear	1.000	1.000	1.000	1.000	1.000
	1.58e-02	rbf	1.000	1.000	1.000	1.000	1.000
	3.98e+00	linear	1.000	1.000	1.000	1.000	1.000
	3.98e+00	rbf	1.000	1.000	1.000	1.000	1.000
	1.00e+03	linear	1.000	1.000	1.000	1.000	1.000
	1.00e+03	rbf	0.999	0.998	0.999	0.998	0.999

Table B.205: *SVM* search grid with the DoS_a dataset

C	γ	kernel	K=1	K=2	K=3	K=4	K=5
	1.00e-05	linear	0.990	0.991	0.991	0.991	0.990
	1.00e-05	rbf	0.894	0.894	0.894	0.894	0.894
	2.15e-04	linear	0.990	0.991	0.991	0.991	0.990
	2.15e-04	rbf	0.894	0.894	0.894	0.894	0.894
	4.64e-03	linear	0.990	0.991	0.991	0.991	0.990
	4.64e-03	rbf	0.985	0.986	0.987	0.987	0.986
1.00e-04	1.00e-01	linear	0.990	0.991	0.991	0.991	0.990
	1.00e-01	rbf	0.994	0.993	0.994	0.994	0.994
	1.00e-05	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-05	rbf	0.922	0.921	0.920	0.922	0.921
	2.15e-04	linear	1.000	1.000	1.000	1.000	1.000
	2.15e-04	rbf	0.989	0.990	0.990	0.990	0.989
	4.64e-03	linear	1.000	1.000	1.000	1.000	1.000
	4.64e-03	rbf	1.000	0.999	0.999	1.000	0.999
2.15e-02	1.00e-01	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-01	rbf	1.000	1.000	1.000	1.000	1.000
	1.00e-05	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-05	rbf	0.998	0.999	0.999	0.999	0.999
	2.15e-04	linear	1.000	1.000	1.000	1.000	1.000
	2.15e-04	rbf	1.000	0.999	1.000	1.000	1.000
	4.64e-03	linear	1.000	1.000	1.000	1.000	1.000
	4.64e-03	rbf	1.000	1.000	1.000	1.000	1.000
4.64e+00	1.00e-01	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-01	rbf	1.000	1.000	1.000	1.000	1.000
	1.00e-05	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-05	rbf	1.000	1.000	1.000	1.000	1.000
	2.15e-04	linear	1.000	1.000	1.000	1.000	1.000
	2.15e-04	rbf	1.000	1.000	1.000	1.000	1.000
	4.64e-03	linear	1.000	1.000	1.000	1.000	1.000
	4.64e-03	rbf	1.000	1.000	1.000	1.000	1.000
1.00e+03	1.00e-01	linear	1.000	1.000	1.000	1.000	1.000
	1.00e-01	rbf	1.000	1.000	1.000	1.000	1.000

Table B.206: *SVM* search grid with the PS_a dataset

C	γ	kernel	K=1	K=2	K=3	K=4	K=5
	1.00e-09	linear	0.982	0.983	0.983	0.983	0.981
	1.00e-09	rbf	0.925	0.925	0.925	0.925	0.925
	2.51e-07	linear	0.982	0.983	0.983	0.983	0.981
	2.51e-07	rbf	0.925	0.925	0.925	0.925	0.925
	6.31e-05	linear	0.982	0.983	0.983	0.983	0.981
1.00e-02	6.31e-05	rbf	0.926	0.926	0.926	0.926	0.926

Continues on next page

Table B.206 – Continuation

C	γ	kernel	K=1	K=2	K=3	K=4	K=5
	1.58e-02	linear	0.982	0.983	0.983	0.983	0.981
	1.58e-02	rbf	0.981	0.982	0.983	0.982	0.980
	3.98e+00	linear	0.982	0.983	0.983	0.983	0.981
	3.98e+00	rbf	0.988	0.987	0.988	0.988	0.987
	1.00e+03	linear	0.982	0.983	0.983	0.983	0.981
	1.00e+03	rbf	0.977	0.976	0.976	0.976	0.975
2.51e+00	1.00e-09	linear	0.984	0.985	0.986	0.985	0.984
	1.00e-09	rbf	0.925	0.925	0.925	0.925	0.925
	2.51e-07	linear	0.984	0.985	0.986	0.985	0.984
	2.51e-07	rbf	0.926	0.926	0.926	0.926	0.926
	6.31e-05	linear	0.984	0.985	0.986	0.985	0.984
	6.31e-05	rbf	0.981	0.982	0.983	0.982	0.980
	1.58e-02	linear	0.984	0.985	0.986	0.985	0.984
	1.58e-02	rbf	0.984	0.985	0.986	0.985	0.984
	3.98e+00	linear	0.984	0.985	0.986	0.985	0.984
	3.98e+00	rbf	0.999	0.999	0.999	0.999	0.999
	1.00e+03	linear	0.984	0.985	0.986	0.985	0.984
	1.00e+03	rbf	0.997	0.996	0.997	0.997	0.997
6.31e+02	1.00e-09	linear	0.984	0.985	0.986	0.985	0.984
	1.00e-09	rbf	0.926	0.926	0.926	0.926	0.926
	2.51e-07	linear	0.984	0.985	0.986	0.985	0.984
	2.51e-07	rbf	0.981	0.982	0.983	0.982	0.980
	6.31e-05	linear	0.984	0.985	0.986	0.985	0.984
	6.31e-05	rbf	0.984	0.985	0.986	0.985	0.984
	1.58e-02	linear	0.984	0.985	0.986	0.985	0.984
	1.58e-02	rbf	0.997	0.997	0.996	0.996	0.996
	3.98e+00	linear	0.984	0.985	0.986	0.985	0.984
	3.98e+00	rbf	0.999	0.999	0.999	0.999	0.999
	1.00e+03	linear	0.984	0.985	0.986	0.985	0.984
	1.00e+03	rbf	0.997	0.996	0.997	0.997	0.997
1.58e+05	1.00e-09	linear	0.984	0.985	0.985	0.984	0.984
	1.00e-09	rbf	0.912	0.912	0.911	0.913	0.910
	2.51e-07	linear	0.984	0.985	0.985	0.984	0.984
	2.51e-07	rbf	0.982	0.983	0.983	0.983	0.981
	6.31e-05	linear	0.984	0.985	0.985	0.984	0.984
	6.31e-05	rbf	0.984	0.985	0.985	0.985	0.984
	1.58e-02	linear	0.984	0.985	0.985	0.984	0.984
	1.58e-02	rbf	0.997	0.997	0.997	0.996	0.997
	3.98e+00	linear	0.984	0.985	0.985	0.984	0.984
	3.98e+00	rbf	0.999	0.999	0.999	0.999	0.999
	1.00e+03	linear	0.984	0.985	0.985	0.984	0.984
	1.00e+03	rbf	0.997	0.996	0.997	0.997	0.997
3.98e+07	1.00e-09	linear	0.983	0.985	0.984	0.984	0.983
	1.00e-09	rbf	0.909	0.909	0.907	0.911	0.906
	2.51e-07	linear	0.983	0.985	0.984	0.984	0.983
	2.51e-07	rbf	0.987	0.987	0.987	0.987	0.985
	6.31e-05	linear	0.983	0.985	0.984	0.984	0.983
	6.31e-05	rbf	0.947	0.948	0.947	0.948	0.948
	1.58e-02	linear	0.983	0.985	0.984	0.984	0.983
	1.58e-02	rbf	0.996	0.996	0.996	0.996	0.996
	3.98e+00	linear	0.983	0.985	0.984	0.984	0.983
	3.98e+00	rbf	0.999	0.999	0.999	0.999	0.999
	1.00e+03	linear	0.983	0.985	0.984	0.984	0.983
	1.00e+03	rbf	0.997	0.996	0.997	0.997	0.997
	1.00e-09	linear	0.983	0.984	0.984	0.984	0.983

Continues on next page

1.00e+10

Table B.206 – Continuation

C	γ	kernel	K=1	K=2	K=3	K=4	K=5
	1.00e-09	rbf	0.803	0.909	0.907	0.911	0.803
	2.51e-07	linear	0.983	0.984	0.984	0.984	0.983
	2.51e-07	rbf	0.987	0.987	0.987	0.987	0.985
	6.31e-05	linear	0.983	0.984	0.984	0.984	0.983
	6.31e-05	rbf	0.947	0.948	0.947	0.948	0.948
	1.58e-02	linear	0.983	0.984	0.984	0.984	0.983
	1.58e-02	rbf	0.996	0.996	0.996	0.996	0.996
	3.98e+00	linear	0.983	0.984	0.984	0.984	0.983
	3.98e+00	rbf	0.999	0.999	0.999	0.999	0.999
	1.00e+03	linear	0.983	0.984	0.984	0.984	0.983
	1.00e+03	rbf	0.997	0.996	0.997	0.997	0.997

Table B.207: *SVM* search grid with the PS_a dataset

C	γ	kernel	K=1	K=2	K=3	K=4	K=5
	1.00e-05	linear	0.975	0.976	0.976	0.974	0.973
	1.00e-05	rbf	0.925	0.925	0.925	0.925	0.925
	2.15e-04	linear	0.975	0.976	0.976	0.974	0.973
	2.15e-04	rbf	0.925	0.925	0.925	0.925	0.925
	4.64e-03	linear	0.975	0.976	0.976	0.974	0.973
	4.64e-03	rbf	0.925	0.925	0.925	0.925	0.925
1.00e-04	1.00e-01	linear	0.975	0.976	0.976	0.974	0.973
	1.00e-01	rbf	0.925	0.925	0.925	0.925	0.925
	1.00e-05	linear	0.984	0.985	0.985	0.984	0.983
	1.00e-05	rbf	0.926	0.926	0.926	0.926	0.926
	2.15e-04	linear	0.984	0.985	0.985	0.984	0.983
	2.15e-04	rbf	0.928	0.929	0.928	0.929	0.929
	4.64e-03	linear	0.984	0.985	0.985	0.984	0.983
	4.64e-03	rbf	0.975	0.976	0.977	0.977	0.974
2.15e-02	1.00e-01	linear	0.984	0.985	0.985	0.984	0.983
	1.00e-01	rbf	0.982	0.983	0.983	0.983	0.981
	1.00e-05	linear	0.984	0.985	0.985	0.985	0.984
	1.00e-05	rbf	0.973	0.974	0.974	0.972	0.971
	2.15e-04	linear	0.984	0.985	0.985	0.985	0.984
	2.15e-04	rbf	0.981	0.982	0.983	0.982	0.980
	4.64e-03	linear	0.984	0.985	0.985	0.985	0.984
4.64e+00	4.64e-03	rbf	0.984	0.985	0.986	0.985	0.984
	1.00e-01	linear	0.984	0.985	0.985	0.985	0.984
	1.00e-01	rbf	0.997	0.997	0.997	0.996	0.997
	1.00e-05	linear	0.984	0.986	0.986	0.985	0.984
	1.00e-05	rbf	0.983	0.984	0.985	0.984	0.983
	2.15e-04	linear	0.984	0.986	0.986	0.985	0.984
	2.15e-04	rbf	0.984	0.985	0.986	0.985	0.984
	4.64e-03	linear	0.984	0.986	0.986	0.985	0.984
1.00e+03	4.64e-03	rbf	0.996	0.997	0.997	0.996	0.997
	1.00e-01	linear	0.984	0.986	0.986	0.985	0.984
	1.00e-01	rbf	0.996	0.997	0.997	0.996	0.996

B.3.2 Results from the experiments with the CIC-IDS2017 dataset

Results for the *DoS* attack

Table B.208: Performance of *svm* classifier with the cic-ids2017_DoS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16619 8]
	1	1.00	1.00	1.00		[23 21721]
2	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[29 21716]
3	0	1.00	1.00	1.00	1.00	[16617 9]
	1	1.00	1.00	1.00		[25 21720]
4	0	1.00	1.00	1.00	1.00	[16616 10]
	1	1.00	1.00	1.00		[18 21727]
5	0	1.00	1.00	1.00	1.00	[16609 17]
	1	1.00	1.00	1.00		[19 21725]

Table B.209: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14547 8]
1	1.00	1.00	1.00		[27 19275]

Table B.210: Performance of *svm* classifier with the cic-ids2017_DoS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16619 8]
	1	1.00	1.00	1.00		[20 21724]
2	0	1.00	1.00	1.00	1.00	[16625 1]
	1	1.00	1.00	1.00		[29 21716]
3	0	1.00	1.00	1.00	1.00	[16617 9]
	1	1.00	1.00	1.00		[23 21722]
4	0	1.00	1.00	1.00	1.00	[16617 9]
	1	1.00	1.00	1.00		[17 21728]
5	0	1.00	1.00	1.00	1.00	[16609 17]
	1	1.00	1.00	1.00		[18 21726]

Table B.211: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14547 8]
1	1.00	1.00	1.00		[25 19277]

Table B.212: Performance of *svm* classifier with the cic-ids2017_DoS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16617 10]
	1	1.00	1.00	1.00		[20 21724]
2	0	1.00	1.00	1.00	1.00	[16623 3]
	1	1.00	1.00	1.00		[27 21718]
3	0	1.00	1.00	1.00	1.00	[16615 11]
	1	1.00	1.00	1.00		[23 21722]
4	0	1.00	1.00	1.00	1.00	[16618 8]
	1	1.00	1.00	1.00		[17 21728]
5	0	1.00	1.00	1.00	1.00	[16608 18]
	1	1.00	1.00	1.00		[18 21726]

Table B.213: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14547 8]
1	1.00	1.00	1.00		[25 19277]

Table B.214: Performance of *svm* classifier with the cic-ids2017_DoS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.98	0.99	0.99	[16348 279]
	1	0.99	1.00	0.99		[28 21716]
2	0	1.00	0.98	0.99	0.99	[16359 267]
	1	0.99	1.00	0.99		[33 21712]
3	0	1.00	0.99	0.99	0.99	[16387 239]
	1	0.99	1.00	0.99		[29 21716]
4	0	1.00	0.98	0.99	0.99	[16318 308]
	1	0.99	1.00	0.99		[15 21730]
5	0	1.00	0.98	0.99	0.99	[16307 319]
	1	0.99	1.00	0.99		[17 21727]

Table B.215: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_b_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.98	0.99	0.99	[14278 277]
1	0.99	1.00	0.99		[23 19279]

Table B.216: Performance of *svm* classifier with the cic-ids2017_DoS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.98	0.99	0.99	[16357 270]
	1	0.99	1.00	0.99		[28 21716]
2	0	1.00	0.98	0.99	0.99	[16360 266]
	1	0.99	1.00	0.99		[34 21711]
3	0	1.00	0.99	0.99	0.99	[16384 242]
	1	0.99	1.00	0.99		[30 21715]
4	0	1.00	0.98	0.99	0.99	[16301 325]
	1	0.99	1.00	0.99		[13 21732]
5	0	1.00	0.98	0.99	0.99	[16316 310]
	1	0.99	1.00	0.99		[17 21727]

Table B.217: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_b_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.98	0.99	0.99	[14295 260]
1	0.99	1.00	0.99		[26 19276]

Table B.218: Performance of *svm* classifier with the cic-ids2017_DoS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.98	0.99	0.99	[16323 304]
	1	0.99	1.00	0.99		[22 21722]
2	0	1.00	0.98	0.99	0.99	[16322 304]
	1	0.99	1.00	0.99		[26 21719]
3	0	1.00	0.98	0.99	0.99	[16368 258]
	1	0.99	1.00	0.99		[28 21717]
4	0	1.00	0.98	0.99	0.99	[16281 345]
	1	0.98	1.00	0.99		[11 21734]
5	0	1.00	0.98	0.99	0.99	[16285 341]
	1	0.98	1.00	0.99		[13 21731]

Table B.219: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.98	0.99	0.99	[14270 285]
1	0.99	1.00	0.99		[24 19278]

Table B.220: Performance of *svm* classifier with the cic-ids2017_DoS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.96	0.98	0.98	[15924 703]
	1	0.97	1.00	0.98		[11 21733]
2	0	1.00	0.96	0.98	0.98	[15905 721]
	1	0.97	1.00	0.98		[16 21729]
3	0	1.00	0.99	0.99	1.00	[16476 150]
	1	0.99	1.00	1.00		[27 21718]
4	0	1.00	0.96	0.98	0.98	[15912 714]
	1	0.97	1.00	0.98		[7 21738]
5	0	1.00	0.96	0.98	0.98	[15939 687]
	1	0.97	1.00	0.98		[11 21733]

Table B.221: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_c_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.99	0.99	1.00	[14422 133]
1	0.99	1.00	1.00		[18 19284]

Table B.222: Performance of *svm* classifier with the cic-ids2017_DoS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.96	0.98	0.98	[15961 666]
	1	0.97	1.00	0.98		[11 21733]
2	0	1.00	0.96	0.98	0.98	[15964 662]
	1	0.97	1.00	0.98		[21 21724]
3	0	1.00	1.00	1.00	1.00	[16583 43]
	1	1.00	1.00	1.00		[31 21714]
4	0	1.00	0.96	0.98	0.98	[15929 697]
	1	0.97	1.00	0.98		[8 21737]
5	0	1.00	0.96	0.98	0.98	[15946 680]
	1	0.97	1.00	0.98		[11 21733]

Table B.223: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_c_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14517 38]
1	1.00	1.00	1.00		[23 19279]

Table B.224: Performance of *svm* classifier with the cic-ids2017_DoS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.96	0.98	0.98	[15918 709]
	1	0.97	1.00	0.98		[13 21731]
2	0	1.00	0.96	0.98	0.98	[15896 730]
	1	0.97	1.00	0.98		[16 21729]
3	0	1.00	0.96	0.98	0.98	[15987 639]
	1	0.97	1.00	0.99		[16 21729]
4	0	1.00	0.96	0.98	0.98	[15899 727]
	1	0.97	1.00	0.98		[8 21737]
5	0	1.00	0.96	0.98	0.98	[15882 744]
	1	0.97	1.00	0.98		[5 21739]

Table B.225: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.96	0.98	0.98	[13923 632]
1	0.97	1.00	0.98		[11 19291]

Table B.226: Performance of *svm* classifier with the cic-ids2017_DoS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.96	0.98	0.98	[15903 724]
	1	0.97	1.00	0.98		[14 21730]
2	0	1.00	0.96	0.98	0.98	[15881 745]
	1	0.97	1.00	0.98		[16 21729]
3	0	1.00	0.96	0.98	0.98	[15972 654]
	1	0.97	1.00	0.98		[16 21729]
4	0	1.00	0.96	0.98	0.98	[15884 742]
	1	0.97	1.00	0.98		[8 21737]
5	0	1.00	0.95	0.98	0.98	[15872 754]
	1	0.97	1.00	0.98		[7 21737]

Table B.227: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_d_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.96	0.98	0.98	[13920 635]
1	0.97	1.00	0.98		[9 19293]

Table B.228: Performance of *svm* classifier with the cic-ids2017_DoS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.96	0.98	0.98	[15902 725]
	1	0.97	1.00	0.98		[14 21730]
2	0	1.00	0.96	0.98	0.98	[15883 743]
	1	0.97	1.00	0.98		[17 21728]
3	0	1.00	0.96	0.98	0.98	[15981 645]
	1	0.97	1.00	0.99		[16 21729]
4	0	1.00	0.96	0.98	0.98	[15884 742]
	1	0.97	1.00	0.98		[8 21737]
5	0	1.00	0.95	0.98	0.98	[15870 756]
	1	0.97	1.00	0.98		[6 21738]

Table B.229: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_d_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.96	0.98	0.98	[13932 623]
1	0.97	1.00	0.98		[9 19293]

Table B.230: Performance of *svm* classifier with the cic-ids2017_DoS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.96	0.98	0.98	[15901 726]
	1	0.97	1.00	0.98		[13 21731]
2	0	1.00	0.96	0.98	0.98	[15882 744]
	1	0.97	1.00	0.98		[15 21730]
3	0	1.00	0.96	0.98	0.98	[15961 665]
	1	0.97	1.00	0.98		[18 21727]
4	0	1.00	0.95	0.98	0.98	[15873 753]
	1	0.97	1.00	0.98		[4 21741]
5	0	1.00	0.95	0.98	0.98	[15872 754]
	1	0.97	1.00	0.98		[5 21739]

Table B.231: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.96	0.98	0.98	[13914 641]
1	0.97	1.00	0.98		[7 19295]

Table B.232: Performance of *svm* classifier with the cic-ids2017_DoS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.95	0.97	0.98	[15713 914]
	1	0.96	1.00	0.98		[9 21735]
2	0	1.00	0.94	0.97	0.98	[15682 944]
	1	0.96	1.00	0.98		[8 21737]
3	0	1.00	0.95	0.97	0.98	[15764 862]
	1	0.96	1.00	0.98		[9 21736]
4	0	1.00	0.94	0.97	0.98	[15675 951]
	1	0.96	1.00	0.98		[3 21742]
5	0	1.00	0.94	0.97	0.98	[15692 934]
	1	0.96	1.00	0.98		[4 21740]

Table B.233: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_e_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.94	0.97	0.98	[13735 820]
1	0.96	1.00	0.98		[7 19295]

Table B.234: Performance of *svm* classifier with the cic-ids2017_DoS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.95	0.97	0.98	[15713 914]
	1	0.96	1.00	0.98		[9 21735]
2	0	1.00	0.94	0.97	0.98	[15684 942]
	1	0.96	1.00	0.98		[8 21737]
3	0	1.00	0.95	0.97	0.98	[15764 862]
	1	0.96	1.00	0.98		[9 21736]
4	0	1.00	0.94	0.97	0.98	[15674 952]
	1	0.96	1.00	0.98		[3 21742]
5	0	1.00	0.94	0.97	0.98	[15694 932]
	1	0.96	1.00	0.98		[4 21740]

Table B.235: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_e_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.94	0.97	0.98	[13735 820]
1	0.96	1.00	0.98		[6 19296]

Table B.236: Performance of *svm* classifier with the cic-ids2017_DoS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.95	0.97	0.98	[15715 912]
	1	0.96	1.00	0.98		[9 21735]
2	0	1.00	0.94	0.97	0.98	[15682 944]
	1	0.96	1.00	0.98		[8 21737]
3	0	1.00	0.95	0.97	0.98	[15765 861]
	1	0.96	1.00	0.98		[9 21736]
4	0	1.00	0.94	0.97	0.98	[15671 955]
	1	0.96	1.00	0.98		[3 21742]
5	0	1.00	0.94	0.97	0.98	[15694 932]
	1	0.96	1.00	0.98		[4 21740]

Table B.237: Performance of *svm* classifier with *Hold Out* section of the cic-ids2017_DoS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.94	0.97	0.98	[13734 821]
1	0.96	1.00	0.98		[6 19296]

Results for the *PS* attack

Table B.238: Performance of *svm* classifier with the PS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21589 59]
	1	1.00	1.00	1.00		[16 26973]
2	0	1.00	1.00	1.00	1.00	[21620 27]
	1	1.00	1.00	1.00		[25 26964]
3	0	1.00	1.00	1.00	1.00	[21619 28]
	1	1.00	1.00	1.00		[24 26965]
4	0	1.00	1.00	1.00	1.00	[21591 56]
	1	1.00	1.00	1.00		[18 26971]
5	0	1.00	1.00	1.00	1.00	[21618 30]
	1	1.00	1.00	1.00		[13 26975]

Table B.239: Performance of *svm* classifier with *Hold Out* section of the PS_a_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19017 38]
1	1.00	1.00	1.00		[10 23850]

Table B.240: Performance of *svm* classifier with the PS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21592 56]
	1	1.00	1.00	1.00		[13 26976]
2	0	1.00	1.00	1.00	1.00	[21622 25]
	1	1.00	1.00	1.00		[25 26964]
3	0	1.00	1.00	1.00	1.00	[21616 31]
	1	1.00	1.00	1.00		[15 26974]
4	0	1.00	1.00	1.00	1.00	[21615 32]
	1	1.00	1.00	1.00		[18 26971]
5	0	1.00	1.00	1.00	1.00	[21626 22]
	1	1.00	1.00	1.00		[13 26975]

Table B.241: Performance of *svm* classifier with *Hold Out* section of the PS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19033 22]
1	1.00	1.00	1.00		[10 23850]

Table B.242: Performance of *svm* classifier with the PS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21597 51]
	1	1.00	1.00	1.00		[13 26976]
2	0	1.00	1.00	1.00	1.00	[21621 26]
	1	1.00	1.00	1.00		[25 26964]
3	0	1.00	1.00	1.00	1.00	[21618 29]
	1	1.00	1.00	1.00		[11 26978]
4	0	1.00	1.00	1.00	1.00	[21616 31]
	1	1.00	1.00	1.00		[16 26973]
5	0	1.00	1.00	1.00	1.00	[21628 20]
	1	1.00	1.00	1.00		[13 26975]

Table B.243: Performance of *svm* classifier with *Hold Out* section of the PS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19034 21]
1	1.00	1.00	1.00		[9 23851]

Table B.244: Performance of *svm* classifier with the PS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21594 54]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21580 67]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	1.00	1.00	1.00	[21583 64]
	1	1.00	1.00	1.00		[23 26966]
4	0	1.00	1.00	1.00	1.00	[21610 37]
	1	1.00	1.00	1.00		[25 26964]
5	0	1.00	1.00	1.00	1.00	[21629 19]
	1	1.00	1.00	1.00		[25 26963]

Table B.245: Performance of *svm* classifier with *Hold Out* section of the PS_b_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19036 19]
1	1.00	1.00	1.00		[14 23846]

Table B.246: Performance of *svm* classifier with the PS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21594 54]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21594 53]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	1.00	1.00	1.00	[21581 66]
	1	1.00	1.00	1.00		[26 26963]
4	0	1.00	1.00	1.00	1.00	[21609 38]
	1	1.00	1.00	1.00		[23 26966]
5	0	1.00	1.00	1.00	1.00	[21628 20]
	1	1.00	1.00	1.00		[25 26963]

Table B.247: Performance of *svm* classifier with *Hold Out* section of the PS_b_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19036 19]
1	1.00	1.00	1.00		[14 23846]

Table B.248: Performance of *svm* classifier with the PS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21596 52]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21594 53]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	1.00	1.00	1.00	[21583 64]
	1	1.00	1.00	1.00		[20 26969]
4	0	1.00	1.00	1.00	1.00	[21570 77]
	1	1.00	1.00	1.00		[23 26966]
5	0	1.00	1.00	1.00	1.00	[21629 19]
	1	1.00	1.00	1.00		[25 26963]

Table B.249: Performance of *svm* classifier with *Hold Out* section of the PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19036 19]
1	1.00	1.00	1.00		[14 23846]

Table B.250: Performance of *svm* classifier with the PS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21625 23]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21627 20]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	1.00	1.00	1.00	[21617 30]
	1	1.00	1.00	1.00		[20 26969]
4	0	1.00	1.00	1.00	1.00	[21614 33]
	1	1.00	1.00	1.00		[23 26966]
5	0	1.00	1.00	1.00	1.00	[21631 17]
	1	1.00	1.00	1.00		[25 26963]

Table B.251: Performance of *svm* classifier with *Hold Out* section of the PS_c_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19035 20]
1	1.00	1.00	1.00		[14 23846]

Table B.252: Performance of *svm* classifier with the PS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21620 28]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21627 20]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	1.00	1.00	1.00	[21616 31]
	1	1.00	1.00	1.00		[20 26969]
4	0	1.00	1.00	1.00	1.00	[21614 33]
	1	1.00	1.00	1.00		[23 26966]
5	0	1.00	1.00	1.00	1.00	[21545 103]
	1	1.00	1.00	1.00		[25 26963]

Table B.253: Performance of *svm* classifier with *Hold Out* section of the PS_c_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19036 19]
1	1.00	1.00	1.00		[14 23846]

Table B.254: Performance of *svm* classifier with the PS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21625 23]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21627 20]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	0.99	1.00	1.00	[21499 148]
	1	0.99	1.00	1.00		[20 26969]
4	0	1.00	0.99	1.00	1.00	[21497 150]
	1	0.99	1.00	1.00		[23 26966]
5	0	1.00	0.99	1.00	1.00	[21493 155]
	1	0.99	1.00	1.00		[25 26963]

Table B.255: Performance of *svm* classifier with *Hold Out* section of the PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19035 20]
1	1.00	1.00	1.00		[15 23845]

Table B.256: Performance of *svm* classifier with the PS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21588 60]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21589 58]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	0.99	1.00	1.00	[21472 175]
	1	0.99	1.00	1.00		[20 26969]
4	0	1.00	1.00	1.00	1.00	[21578 69]
	1	1.00	1.00	1.00		[23 26966]
5	0	1.00	0.99	1.00	1.00	[21463 185]
	1	0.99	1.00	1.00		[25 26963]

Table B.257: Performance of *svm* classifier with *Hold Out* section of the PS_d_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19003 52]
1	1.00	1.00	1.00		[14 23846]

Table B.258: Performance of *svm* classifier with the PS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21588 60]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	0.99	1.00	1.00	[21499 148]
	1	0.99	1.00	1.00		[35 26954]
3	0	1.00	1.00	1.00	1.00	[21579 68]
	1	1.00	1.00	1.00		[20 26969]
4	0	1.00	0.99	1.00	1.00	[21479 168]
	1	0.99	1.00	1.00		[23 26966]
5	0	1.00	0.99	0.99	0.99	[21327 321]
	1	0.99	1.00	0.99		[25 26963]

Table B.259: Performance of *svm* classifier with *Hold Out* section of the PS_d_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19003 52]
1	1.00	1.00	1.00		[14 23846]

Table B.260: Performance of *svm* classifier with the PS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21588 60]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21589 58]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	0.99	1.00	1.00	[21461 186]
	1	0.99	1.00	1.00		[20 26969]
4	0	1.00	0.99	1.00	1.00	[21462 185]
	1	0.99	1.00	1.00		[23 26966]
5	0	1.00	0.99	0.99	1.00	[21454 194]
	1	0.99	1.00	1.00		[25 26963]

Table B.261: Performance of *svm* classifier with *Hold Out* section of the PS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19003 52]
1	1.00	1.00	1.00		[14 23846]

Table B.262: Performance of *svm* classifier with the PS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.99	1.00	1.00	1.00	[21628 20]
	1	1.00	0.99	1.00		[155 26834]
2	0	0.99	1.00	1.00	1.00	[21618 29]
	1	1.00	0.99	1.00		[175 26814]
3	0	0.99	1.00	0.99	1.00	[21608 39]
	1	1.00	0.99	1.00		[180 26809]
4	0	1.00	0.99	0.99	0.99	[21374 273]
	1	0.99	1.00	0.99		[45 26944]
5	0	0.99	1.00	1.00	1.00	[21616 32]
	1	1.00	0.99	1.00		[181 26807]

Table B.263: Performance of *svm* classifier with *Hold Out* section of the PS_e_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.99	1.00	1.00	1.00	[19026 29]
1	1.00	0.99	1.00		[135 23725]

Table B.264: Performance of *svm* classifier with the PS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.99	1.00	1.00	1.00	[21628 20]
	1	1.00	0.99	1.00		[155 26834]
2	0	1.00	1.00	1.00	1.00	[21618 29]
	1	1.00	1.00	1.00		[103 26886]
3	0	1.00	1.00	1.00	1.00	[21605 42]
	1	1.00	1.00	1.00		[101 26888]
4	0	0.99	1.00	1.00	1.00	[21608 39]
	1	1.00	0.99	1.00		[177 26812]
5	0	0.99	1.00	1.00	1.00	[21616 32]
	1	1.00	0.99	1.00		[181 26807]

Table B.265: Performance of *svm* classifier with *Hold Out* section of the PS_e_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19026 29]
1	1.00	1.00	1.00		[67 23793]

Table B.266: Performance of *svm* classifier with the PS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21554 94]
	1	1.00	1.00	1.00		[76 26913]
2	0	1.00	0.99	0.99	0.99	[21394 253]
	1	0.99	1.00	0.99		[62 26927]
3	0	1.00	1.00	1.00	1.00	[21605 42]
	1	1.00	1.00	1.00		[101 26888]
4	0	1.00	1.00	1.00	1.00	[21604 43]
	1	1.00	1.00	1.00		[89 26900]
5	0	1.00	0.98	0.99	0.99	[21316 332]
	1	0.99	1.00	0.99		[45 26943]

Table B.267: Performance of *svm* classifier with *Hold Out* section of the PS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19025 30]
1	1.00	1.00	1.00		[67 23793]

B.3.3 Results from the collected dataset(s)

Results for the *DoS* attack

Table B.268: Performance of *svm* classifier with the DoS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3041 4]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.269: Performance of *svm* classifier with *Hold Out* section of the DoS_a_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.270: Performance of *svm* classifier with the DoS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3034 11]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3032 14]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.271: Performance of *svm* classifier with *Hold Out* section of the DoS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.272: Performance of *svm* classifier with the DoS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3033 12]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3045 1]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.273: Performance of *svm* classifier with *Hold Out* section of the DoS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.274: Performance of *svm* classifier with the DoS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	0.99	1.00	1.00	[3020 25]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.275: Performance of *svm* classifier with *Hold Out* section of the DoS_b_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.276: Performance of *svm* classifier with the DoS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3042 3]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.277: Performance of *svm* classifier with *Hold Out* section of the DoS_b_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.278: Performance of *svm* classifier with the DoS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	0.99	1.00	1.00	[3020 25]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.279: Performance of *svm* classifier with *Hold Out* section of the DoS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.280: Performance of *svm* classifier with the DoS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.94	0.98	0.96	0.99	[2987 58]
	1	1.00	0.99	1.00		[183 25485]
2	0	0.88	1.00	0.93	0.98	[3033 12]
	1	1.00	0.98	0.99		[423 25245]
3	0	0.99	0.03	0.05	0.90	[84 2961]
	1	0.90	1.00	0.95		[1 25667]
4	0	1.00	0.02	0.05	0.90	[74 2972]
	1	0.90	1.00	0.95		[0 25667]
5	0	0.10	0.99	0.19	0.10	[3001 44]
	1	0.00	0.00	0.00		[25667 0]

Table B.281: Performance of *svm* classifier with *Hold Out* section of the DoS_c_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.94	0.98	0.96	0.99	[2627 58]
1	1.00	0.99	1.00		[154 22496]

Table B.282: Performance of *svm* classifier with the DoS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.91	0.23	0.37	0.92	[698 2347]
	1	0.92	1.00	0.95		[66 25602]
2	0	0.11	0.99	0.19	0.11	[3021 24]
	1	0.00	0.00	0.00		[25668 0]
3	0	0.68	0.12	0.20	0.90	[365 2680]
	1	0.90	0.99	0.95		[175 25493]
4	0	1.00	0.02	0.05	0.90	[73 2973]
	1	0.90	1.00	0.95		[0 25667]
5	0	0.10	0.99	0.19	0.10	[3003 42]
	1	0.00	0.00	0.00		[25667 0]

Table B.283: Performance of *svm* classifier with *Hold Out* section of the DoS_c_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.91	0.22	0.35	0.91	[578 2107]
1	0.91	1.00	0.95		[60 22590]

Table B.284: Performance of *svm* classifier with the DoS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.94	0.98	0.96	0.99	[2987 58]
	1	1.00	0.99	1.00		[183 25485]
2	0	0.11	0.99	0.19	0.11	[3017 28]
	1	0.00	0.00	0.00		[25668 0]
3	0	1.00	0.03	0.05	0.90	[82 2963]
	1	0.90	1.00	0.95		[0 25668]
4	0	1.00	0.02	0.05	0.90	[73 2973]
	1	0.90	1.00	0.95		[0 25667]
5	0	1.00	0.10	0.19	0.90	[312 2733]
	1	0.90	1.00	0.95		[0 25667]

Table B.285: Performance of *svm* classifier with *Hold Out* section of the DoS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	0.10	0.19	0.90	[275 2410]
1	0.90	1.00	0.95		[0 22650]

Table B.286: Performance of *svm* classifier with the DoS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.02	0.04	0.90	[60 2985]
	1	0.90	1.00	0.95		[0 25668]
2	0	1.00	0.01	0.03	0.90	[45 3000]
	1	0.90	1.00	0.94		[0 25668]
3	0	0.11	0.99	0.19	0.13	[3010 35]
	1	0.95	0.02	0.05		[25039 629]
4	0	0.11	0.99	0.19	0.11	[3018 28]
	1	0.00	0.00	0.00		[25667 0]
5	0	0.10	0.99	0.19	0.10	[3009 36]
	1	0.00	0.00	0.00		[25667 0]

Table B.287: Performance of *svm* classifier with *Hold Out* section of the DoS_d_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.11	0.99	0.19	0.13	[2652 33]
1	0.94	0.02	0.05		[22087 563]

Table B.288: Performance of *svm* classifier with the DoS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.10	0.98	0.19	0.10	[2978 67]
	1	0.00	0.00	0.00		[25668 0]
2	0	1.00	0.01	0.02	0.89	[25 3020]
	1	0.89	1.00	0.94		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3043 2]
	1	1.00	1.00	1.00		[3 25665]
4	0	1.00	0.01	0.02	0.89	[30 3016]
	1	0.89	1.00	0.94		[0 25667]
5	0	1.00	0.02	0.04	0.90	[62 2983]
	1	0.90	1.00	0.95		[0 25667]

Table B.289: Performance of *svm* classifier with *Hold Out* section of the DoS_d_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[3 22647]

Table B.290: Performance of *svm* classifier with the DoS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.02	0.04	0.90	[60 2985]
	1	0.90	1.00	0.95		[0 25668]
2	0	1.00	0.01	0.01	0.89	[23 3022]
	1	0.89	1.00	0.94		[0 25668]
3	0	0.11	0.99	0.19	0.11	[3010 35]
	1	0.83	0.01	0.01		[25492 176]
4	0	0.11	0.99	0.19	0.11	[3018 28]
	1	0.00	0.00	0.00		[25667 0]
5	0	1.00	0.02	0.04	0.90	[56 2989]
	1	0.90	1.00	0.94		[0 25667]

Table B.291: Performance of *svm* classifier with *Hold Out* section of the DoS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.10	0.99	0.19	0.10	[2653 32]
1	0.00	0.00	0.00		[22650 0]

Table B.292: Performance of *svm* classifier with the DoS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.10	0.98	0.19	0.10	[2985 60]
	1	0.00	0.00	0.00		[25668 0]
2	0	0.10	0.98	0.19	0.10	[2998 47]
	1	0.00	0.00	0.00		[25668 0]
3	0	0.10	0.97	0.19	0.10	[2967 78]
	1	0.00	0.00	0.00		[25668 0]
4	0	0.11	1.00	0.19	0.11	[3038 8]
	1	0.47	0.00	0.00		[25660 7]
5	0	0.11	0.99	0.19	0.11	[3023 22]
	1	0.15	0.00	0.00		[25663 4]

Table B.293: Performance of *svm* classifier with *Hold Out* section of the DoS_e_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.11	1.00	0.19	0.11	[2680 5]
1	0.55	0.00	0.00		[22644 6]

Table B.294: Performance of *svm* classifier with the DoS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.10	0.98	0.19	0.10	[2995 50]
	1	0.00	0.00	0.00		[25668 0]
2	0	0.11	1.00	0.19	0.11	[3042 3]
	1	0.62	0.00	0.00		[25663 5]
3	0	1.00	0.03	0.05	0.90	[77 2968]
	1	0.90	1.00	0.95		[0 25668]
4	0	0.11	1.00	0.19	0.11	[3041 5]
	1	0.58	0.00	0.00		[25660 7]
5	0	1.00	0.02	0.04	0.90	[62 2983]
	1	0.90	1.00	0.95		[0 25667]

Table B.295: Performance of *svm* classifier with *Hold Out* section of the DoS_e_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.11	1.00	0.19	0.11	[2683 2]
1	0.75	0.00	0.00		[22644 6]

Table B.296: Performance of *svm* classifier with the DoS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2985 60] [25668 0]
2	0 1	0.11 0.83	1.00 0.00	0.19 0.00	0.11	[3044 1] [25663 5]
3	0 1	0.10 0.00	0.97 0.00	0.19 0.00	0.10	[2967 78] [25668 0]
4	0 1	0.11 0.58	1.00 0.00	0.19 0.00	0.11	[3041 5] [25660 7]
5	0 1	0.10 0.00	0.98 0.00	0.19 0.00	0.10	[2979 66] [25667 0]

Table B.297: Performance of *svm* classifier with *Hold Out* section of the DoS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0 1	0.11 0.86	1.00 0.00	0.19 0.00	0.11	[2684 1] [22644 6]

Results for the *PS* attack

Table B.298: Performance of *svm* classifier with the PS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1410 1] [290 17008]
2	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1410 1] [280 17018]
3	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1410 1] [293 17005]
4	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1409 1] [289 17009]
5	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1410 0] [287 17011]

Table B.299: Performance of *svm* classifier with *Hold Out* section of the PS_a_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.99	[1197 0] [247 15064]

Table B.300: Performance of *svm* classifier with the PS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.83	1.00	0.91	0.98	[1409 2]
	1	1.00	0.98	0.99		[290 17008]
2	0	0.83	1.00	0.91	0.98	[1410 1]
	1	1.00	0.98	0.99		[280 17018]
3	0	0.83	1.00	0.91	0.98	[1410 1]
	1	1.00	0.98	0.99		[293 17005]
4	0	0.83	1.00	0.91	0.98	[1408 2]
	1	1.00	0.98	0.99		[288 17010]
5	0	0.83	1.00	0.91	0.98	[1410 0]
	1	1.00	0.98	0.99		[288 17010]

Table B.301: Performance of *svm* classifier with *Hold Out* section of the PS_a_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.83	1.00	0.91	0.99	[1197 0]
1	1.00	0.98	0.99		[247 15064]

Table B.302: Performance of *svm* classifier with the PS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.83	1.00	0.91	0.98	[1407 4]
	1	1.00	0.98	0.99		[290 17008]
2	0	0.83	1.00	0.91	0.98	[1409 2]
	1	1.00	0.98	0.99		[280 17018]
3	0	0.83	1.00	0.91	0.98	[1409 2]
	1	1.00	0.98	0.99		[293 17005]
4	0	0.83	1.00	0.91	0.98	[1406 4]
	1	1.00	0.98	0.99		[288 17010]
5	0	0.83	1.00	0.91	0.98	[1409 1]
	1	1.00	0.98	0.99		[287 17011]

Table B.303: Performance of *svm* classifier with *Hold Out* section of the PS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.83	1.00	0.91	0.98	[1195 2]
1	1.00	0.98	0.99		[247 15064]

Table B.304: Performance of *svm* classifier with the PS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1408 3] [290 17008]
2	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1409 2] [283 17015]
3	0 1	0.83 1.00	1.00 0.98	0.90 0.99	0.98	[1407 4] [294 17004]
4	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1406 4] [286 17012]
5	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1409 1] [291 17007]

Table B.305: Performance of *svm* classifier with *Hold Out* section of the PS_b_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0 1	0.83 1.00	1.00 0.98	0.90 0.99	0.98	[1195 2] [250 15061]

Table B.306: Performance of *svm* classifier with the PS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1408 3] [290 17008]
2	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1409 2] [280 17018]
3	0 1	0.83 1.00	1.00 0.98	0.90 0.99	0.98	[1408 3] [295 17003]
4	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1406 4] [289 17009]
5	0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1409 1] [291 17007]

Table B.307: Performance of *svm* classifier with *Hold Out* section of the PS_b_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0 1	0.83 1.00	1.00 0.98	0.91 0.99	0.98	[1195 2] [247 15064]

Table B.308: Performance of *svm* classifier with the PS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.83	1.00	0.91	0.98	[1407 4]
	1	1.00	0.98	0.99		[290 17008]
2	0	0.83	1.00	0.91	0.98	[1407 4]
	1	1.00	0.98	0.99		[280 17018]
3	0	0.83	1.00	0.90	0.98	[1406 5]
	1	1.00	0.98	0.99		[293 17005]
4	0	0.83	1.00	0.91	0.98	[1403 7]
	1	1.00	0.98	0.99		[287 17011]
5	0	0.83	1.00	0.91	0.98	[1405 5]
	1	1.00	0.98	0.99		[287 17011]

Table B.309: Performance of *svm* classifier with *Hold Out* section of the PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.83	0.99	0.90	0.98	[1190 7]
1	1.00	0.98	0.99		[247 15064]

Table B.310: Performance of *svm* classifier with the PS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.78	1.00	0.87	0.98	[1407 4]
	1	1.00	0.98	0.99		[405 16893]
2	0	0.77	1.00	0.87	0.98	[1407 4]
	1	1.00	0.98	0.99		[423 16875]
3	0	0.76	1.00	0.86	0.98	[1405 6]
	1	1.00	0.97	0.99		[434 16864]
4	0	0.77	1.00	0.87	0.98	[1403 7]
	1	1.00	0.98	0.99		[409 16889]
5	0	0.77	1.00	0.87	0.98	[1405 5]
	1	1.00	0.98	0.99		[410 16888]

Table B.311: Performance of *svm* classifier with *Hold Out* section of the PS_c_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.77	0.99	0.87	0.98	[1190 7]
1	1.00	0.98	0.99		[363 14948]

Table B.312: Performance of *svm* classifier with the PS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.78	1.00	0.87	0.98	[1407 4]
	1	1.00	0.98	0.99		[405 16893]
2	0	0.77	1.00	0.87	0.98	[1407 4]
	1	1.00	0.98	0.99		[423 16875]
3	0	0.76	1.00	0.86	0.98	[1404 7]
	1	1.00	0.97	0.99		[434 16864]
4	0	0.78	1.00	0.87	0.98	[1403 7]
	1	1.00	0.98	0.99		[406 16892]
5	0	0.77	1.00	0.87	0.98	[1405 5]
	1	1.00	0.98	0.99		[409 16889]

Table B.313: Performance of *svm* classifier with *Hold Out* section of the PS_c_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.77	0.99	0.87	0.98	[1190 7]
1	1.00	0.98	0.99		[363 14948]

Table B.314: Performance of *svm* classifier with the PS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.83	0.98	0.90	0.98	[1379 32]
	1	1.00	0.98	0.99		[290 17008]
2	0	0.83	0.98	0.90	0.98	[1387 24]
	1	1.00	0.98	0.99		[280 17018]
3	0	0.82	0.97	0.89	0.98	[1373 38]
	1	1.00	0.98	0.99		[293 17005]
4	0	0.83	0.98	0.90	0.98	[1375 35]
	1	1.00	0.98	0.99		[285 17013]
5	0	0.83	0.98	0.90	0.98	[1378 32]
	1	1.00	0.98	0.99		[287 17011]

Table B.315: Performance of *svm* classifier with *Hold Out* section of the PS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.82	0.97	0.89	0.98	[1162 35]
1	1.00	0.98	0.99		[247 15064]

Table B.316: Performance of *svm* classifier with the PS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.78	1.00	0.87	0.98	[1407 4]
	1	1.00	0.98	0.99		[405 16893]
2	0	0.77	1.00	0.87	0.98	[1407 4]
	1	1.00	0.98	0.99		[419 16879]
3	0	0.77	1.00	0.87	0.98	[1405 6]
	1	1.00	0.98	0.99		[430 16868]
4	0	0.78	1.00	0.87	0.98	[1403 7]
	1	1.00	0.98	0.99		[406 16892]
5	0	0.77	1.00	0.87	0.98	[1405 5]
	1	1.00	0.98	0.99		[409 16889]

Table B.317: Performance of *svm* classifier with *Hold Out* section of the PS_d_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.77	0.99	0.87	0.98	[1190 7]
1	1.00	0.98	0.99		[361 14950]

Table B.318: Performance of *svm* classifier with the PS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.78	1.00	0.87	0.98	[1407 4]
	1	1.00	0.98	0.99		[405 16893]
2	0	0.77	1.00	0.87	0.98	[1407 4]
	1	1.00	0.98	0.99		[419 16879]
3	0	0.77	1.00	0.87	0.98	[1405 6]
	1	1.00	0.98	0.99		[430 16868]
4	0	0.78	1.00	0.87	0.98	[1403 7]
	1	1.00	0.98	0.99		[406 16892]
5	0	0.77	1.00	0.87	0.98	[1405 5]
	1	1.00	0.98	0.99		[409 16889]

Table B.319: Performance of *svm* classifier with *Hold Out* section of the PS_d_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.77	0.99	0.87	0.98	[1190 7]
1	1.00	0.98	0.99		[361 14950]

Table B.320: Performance of *svm* classifier with the PS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.83	0.98	0.89	0.98	[1377 34]
	1	1.00	0.98	0.99		[290 17008]
2	0	0.83	0.98	0.90	0.98	[1383 28]
	1	1.00	0.98	0.99		[280 17018]
3	0	0.82	0.97	0.89	0.98	[1371 40]
	1	1.00	0.98	0.99		[293 17005]
4	0	0.83	0.98	0.90	0.98	[1375 35]
	1	1.00	0.98	0.99		[285 17013]
5	0	0.83	0.98	0.90	0.98	[1377 33]
	1	1.00	0.98	0.99		[287 17011]

Table B.321: Performance of *svm* classifier with *Hold Out* section of the PS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.82	0.97	0.89	0.98	[1162 35]
1	1.00	0.98	0.99		[247 15064]

Table B.322: Performance of *svm* classifier with the PS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.83	0.98	0.90	0.98	[1389 22]
	1	1.00	0.98	0.99		[290 17008]
2	0	0.83	0.99	0.90	0.98	[1390 21]
	1	1.00	0.98	0.99		[280 17018]
3	0	0.83	0.98	0.90	0.98	[1383 28]
	1	1.00	0.98	0.99		[293 17005]
4	0	0.83	0.98	0.90	0.98	[1383 27]
	1	1.00	0.98	0.99		[285 17013]
5	0	0.83	0.99	0.90	0.98	[1389 21]
	1	1.00	0.98	0.99		[287 17011]

Table B.323: Performance of *svm* classifier with *Hold Out* section of the PS_e_i dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.83	0.98	0.90	0.98	[1173 24]
1	1.00	0.98	0.99		[247 15064]

Table B.324: Performance of *svm* classifier with the PS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.83	0.98	0.90	0.98	[1389 22]
	1	1.00	0.98	0.99		[290 17008]
2	0	0.83	0.99	0.90	0.98	[1390 21]
	1	1.00	0.98	0.99		[280 17018]
3	0	0.83	0.98	0.90	0.98	[1383 28]
	1	1.00	0.98	0.99		[293 17005]
4	0	0.83	0.98	0.90	0.98	[1383 27]
	1	1.00	0.98	0.99		[285 17013]
5	0	0.83	0.99	0.90	0.98	[1389 21]
	1	1.00	0.98	0.99		[287 17011]

Table B.325: Performance of *svm* classifier with *Hold Out* section of the PS_e_ii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.83	0.98	0.90	0.98	[1173 24]
1	1.00	0.98	0.99		[247 15064]

Table B.326: Performance of *svm* classifier with the PS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.83	0.97	0.89	0.98	[1371 40]
	1	1.00	0.98	0.99		[290 17008]
2	0	0.83	0.98	0.90	0.98	[1378 33]
	1	1.00	0.98	0.99		[280 17018]
3	0	0.82	0.97	0.89	0.98	[1367 44]
	1	1.00	0.98	0.99		[293 17005]
4	0	0.83	0.97	0.89	0.98	[1365 45]
	1	1.00	0.98	0.99		[285 17013]
5	0	0.83	0.97	0.89	0.98	[1370 40]
	1	1.00	0.98	0.99		[287 17011]

Table B.327: Performance of *svm* classifier with *Hold Out* section of the PS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.82	0.96	0.89	0.98	[1153 44]
1	1.00	0.98	0.99		[247 15064]

B.4 CNN

B.4.1 Results from the experiments with the CIC-IDS2017 dataset

Results for the *DoS* attack

Table B.328: Performance of *CNN* with the cic-ids2017_DoS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16622 5] [23 21721]
2	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16621 5] [27 21718]
3	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16616 10] [25 21720]
4	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16608 18] [18 21727]
5	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16617 9] [17 21727]

Table B.329: Performance of *CNN* with the cic-ids2017_DoS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16609 18] [17 21727]
2	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16620 6] [23 21722]
3	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16618 8] [25 21720]
4	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16607 19] [18 21727]
5	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[16617 9] [20 21724]

Table B.330: Performance of *CNN* with the cic-ids2017_DoS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16611 16]
	1	1.00	1.00	1.00		[22 21722]
2	0	1.00	1.00	1.00	1.00	[16614 12]
	1	1.00	1.00	1.00		[32 21713]
3	0	1.00	1.00	1.00	1.00	[16599 27]
	1	1.00	1.00	1.00		[21 21724]
4	0	1.00	1.00	1.00	1.00	[16598 28]
	1	1.00	1.00	1.00		[17 21728]
5	0	1.00	1.00	1.00	1.00	[16618 8]
	1	1.00	1.00	1.00		[19 21725]

Table B.331: Performance of *CNN* with *Hold Out* section of the cic-ids2017_DoS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14544 11]
1	1.00	1.00	1.00		[28 19274]

Table B.332: Performance of *CNN* with the cic-ids2017_DoS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16616 11]
	1	1.00	1.00	1.00		[23 21721]
2	0	1.00	1.00	1.00	1.00	[16624 2]
	1	1.00	1.00	1.00		[65 21680]
3	0	1.00	1.00	1.00	1.00	[16581 45]
	1	1.00	1.00	1.00		[23 21722]
4	0	1.00	1.00	1.00	1.00	[16591 35]
	1	1.00	1.00	1.00		[18 21727]
5	0	1.00	1.00	1.00	1.00	[16606 20]
	1	1.00	1.00	1.00		[23 21721]

Table B.333: Performance of *CNN* with the cic-ids2017_DoS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16603 24]
	1	1.00	1.00	1.00		[12 21732]
2	0	1.00	1.00	1.00	1.00	[16593 33]
	1	1.00	1.00	1.00		[29 21716]
3	0	1.00	1.00	1.00	1.00	[16616 10]
	1	1.00	1.00	1.00		[29 21716]
4	0	1.00	1.00	1.00	1.00	[16612 14]
	1	1.00	1.00	1.00		[78 21667]
5	0	1.00	1.00	1.00	1.00	[16607 19]
	1	1.00	1.00	1.00		[30 21714]

Table B.334: Performance of *CNN* with the cic-ids2017_DoS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16615 12]
	1	1.00	1.00	1.00		[14 21730]
2	0	1.00	1.00	1.00	1.00	[16595 31]
	1	1.00	1.00	1.00		[28 21717]
3	0	1.00	1.00	1.00	1.00	[16588 38]
	1	1.00	1.00	1.00		[27 21718]
4	0	1.00	1.00	1.00	1.00	[16603 23]
	1	1.00	1.00	1.00		[17 21728]
5	0	1.00	1.00	1.00	1.00	[16591 35]
	1	1.00	1.00	1.00		[17 21727]

Table B.335: Performance of *CNN* with *Hold Out* section of the cic-ids2017_DoS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14536 19]
1	1.00	1.00	1.00		[26 19276]

Table B.336: Performance of *CNN* with the cic-ids2017_DoS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16621 6]
	1	1.00	1.00	1.00		[26 21718]
2	0	1.00	1.00	1.00	1.00	[16611 15]
	1	1.00	1.00	1.00		[32 21713]
3	0	1.00	1.00	1.00	1.00	[16619 7]
	1	1.00	1.00	1.00		[28 21717]
4	0	1.00	1.00	1.00	1.00	[16607 19]
	1	1.00	1.00	1.00		[20 21725]
5	0	1.00	1.00	1.00	1.00	[16606 20]
	1	1.00	1.00	1.00		[21 21723]

Table B.337: Performance of *CNN* with the cic-ids2017_DoS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16620 7]
	1	1.00	1.00	1.00		[27 21717]
2	0	1.00	1.00	1.00	1.00	[16600 26]
	1	1.00	1.00	1.00		[27 21718]
3	0	1.00	1.00	1.00	1.00	[16619 7]
	1	1.00	1.00	1.00		[28 21717]
4	0	1.00	1.00	1.00	1.00	[16619 7]
	1	1.00	1.00	1.00		[22 21723]
5	0	1.00	1.00	1.00	1.00	[16618 8]
	1	1.00	1.00	1.00		[33 21711]

Table B.338: Performance of *CNN* with the cic-ids2017_DoS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.96	0.98	0.98	[15984 643]
	1	0.97	1.00	0.98		[20 21724]
2	0	1.00	1.00	1.00	1.00	[16608 18]
	1	1.00	1.00	1.00		[29 21716]
3	0	1.00	1.00	1.00	1.00	[16609 17]
	1	1.00	1.00	1.00		[28 21717]
4	0	1.00	1.00	1.00	1.00	[16614 12]
	1	1.00	1.00	1.00		[29 21716]
5	0	1.00	1.00	1.00	1.00	[16611 15]
	1	1.00	1.00	1.00		[23 21721]

Table B.339: Performance of *CNN* with *Hold Out* section of the cic-ids2017_DoS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14539 16]
1	1.00	1.00	1.00		[26 19276]

Table B.340: Performance of *CNN* with the cic-ids2017_DoS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.98	1.00	0.99	0.99	[16569 58]
	1	1.00	0.99	0.99		[316 21428]
2	0	1.00	1.00	1.00	1.00	[16601 25]
	1	1.00	1.00	1.00		[39 21706]
3	0	0.99	1.00	0.99	0.99	[16588 38]
	1	1.00	0.99	1.00		[156 21589]
4	0	1.00	1.00	1.00	1.00	[16584 42]
	1	1.00	1.00	1.00		[17 21728]
5	0	1.00	1.00	1.00	1.00	[16603 23]
	1	1.00	1.00	1.00		[21 21723]

Table B.341: Performance of *CNN* with the cic-ids2017_DoS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16612 15]
	1	1.00	1.00	1.00		[54 21690]
2	0	0.98	1.00	0.99	0.99	[16587 39]
	1	1.00	0.98	0.99		[353 21392]
3	0	0.99	1.00	0.99	1.00	[16600 26]
	1	1.00	0.99	1.00		[156 21589]
4	0	0.99	1.00	0.99	1.00	[16570 56]
	1	1.00	0.99	1.00		[123 21622]
5	0	1.00	1.00	1.00	1.00	[16606 20]
	1	1.00	1.00	1.00		[21 21723]

Table B.342: Performance of *CNN* with the cic-ids2017_DoS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.99	0.99	0.99	0.99	[16509 118]
	1	0.99	1.00	0.99		[108 21636]
2	0	1.00	1.00	1.00	1.00	[16583 43]
	1	1.00	1.00	1.00		[48 21697]
3	0	0.99	0.99	0.99	0.99	[16536 90]
	1	1.00	0.99	0.99		[238 21507]
4	0	0.98	1.00	0.99	0.99	[16552 74]
	1	1.00	0.99	0.99		[269 21476]
5	0	0.99	1.00	0.99	0.99	[16587 39]
	1	1.00	0.99	0.99		[239 21505]

Table B.343: Performance of *CNN* with *Hold Out* section of the cic-ids2017_DoS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14542 13]
1	1.00	1.00	1.00		[26 19276]

Table B.344: Performance of *CNN* with the cic-ids2017_DoS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.99	1.00	0.99	0.99	[16591 36]
	1	1.00	0.99	0.99		[238 21506]
2	0	1.00	1.00	1.00	1.00	[16603 23]
	1	1.00	1.00	1.00		[72 21673]
3	0	0.99	0.99	0.99	0.99	[16518 108]
	1	1.00	0.99	0.99		[212 21533]
4	0	0.98	1.00	0.99	0.99	[16593 33]
	1	1.00	0.99	0.99		[254 21491]
5	0	1.00	1.00	1.00	1.00	[16583 43]
	1	1.00	1.00	1.00		[26 21718]

Table B.345: Performance of *CNN* with the cic-ids2017_DoS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.98	1.00	0.99	0.99	[16592 35]
	1	1.00	0.99	0.99		[269 21475]
2	0	1.00	1.00	1.00	1.00	[16575 51]
	1	1.00	1.00	1.00		[38 21707]
3	0	0.98	1.00	0.99	0.99	[16592 34]
	1	1.00	0.98	0.99		[388 21357]
4	0	0.99	1.00	0.99	0.99	[16589 37]
	1	1.00	0.99	0.99		[190 21555]
5	0	0.98	1.00	0.99	0.99	[16577 49]
	1	1.00	0.99	0.99		[294 21450]

Table B.346: Performance of *CNN* with the cic-ids2017_DoS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[16587 40]
	1	1.00	1.00	1.00		[26 21718]
2	0	0.98	0.99	0.99	0.99	[16461 165]
	1	0.99	0.99	0.99		[326 21419]
3	0	0.99	0.99	0.99	0.99	[16542 84]
	1	1.00	0.99	0.99		[239 21506]
4	0	1.00	1.00	1.00	1.00	[16585 41]
	1	1.00	1.00	1.00		[18 21727]
5	0	0.99	0.99	0.99	0.99	[16451 175]
	1	0.99	0.99	0.99		[139 21605]

Table B.347: Performance of *CNN* with *Hold Out* section of the cic-ids2017_DoS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[14518 37]
1	1.00	1.00	1.00		[32 19270]

Results for the *PS* attack

Table B.348: Performance of *CNN* with the cic-ids2017_PS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21647 1] [19 26970]
2	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21641 6] [27 26962]
3	0 1	1.00 0.99	0.99 1.00	0.99 0.99	0.99	[21347 300] [10 26979]
4	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21632 15] [18 26971]
5	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21639 9] [19 26969]

Table B.349: Performance of *CNN* with the cic-ids2017_PS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21614 34] [13 26976]
2	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21631 16] [26 26963]
3	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21627 20] [14 26975]
4	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21623 24] [18 26971]
5	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21626 22] [19 26969]

Table B.350: Performance of *CNN* with the cic-ids2017_PS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21642 6] [13 26976]
2	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21630 17] [34 26955]
3	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21636 11] [16 26973]
4	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21610 37] [18 26971]
5	0 1	1.00 1.00	1.00 1.00	1.00 1.00	1.00	[21647 1] [20 26968]

Table B.351: Performance of *CNN* with *Hold Out* section of the cic-ids2017_PS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19048 7]
1	1.00	1.00	1.00		[11 23849]

Table B.352: Performance of *CNN* with the cic-ids2017_PS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21629 19]
	1	1.00	1.00	1.00		[13 26976]
2	0	1.00	1.00	1.00	1.00	[21633 14]
	1	1.00	1.00	1.00		[27 26962]
3	0	1.00	1.00	1.00	1.00	[21622 25]
	1	1.00	1.00	1.00		[15 26974]
4	0	1.00	1.00	1.00	1.00	[21629 18]
	1	1.00	1.00	1.00		[16 26973]
5	0	1.00	1.00	1.00	1.00	[21648 0]
	1	1.00	1.00	1.00		[25 26963]

Table B.353: Performance of *CNN* with the cic-ids2017_PS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21623 25]
	1	1.00	1.00	1.00		[11 26978]
2	0	1.00	1.00	1.00	1.00	[21635 12]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	1.00	1.00	1.00	[21610 37]
	1	1.00	1.00	1.00		[11 26978]
4	0	1.00	1.00	1.00	1.00	[21624 23]
	1	1.00	1.00	1.00		[19 26970]
5	0	1.00	1.00	1.00	1.00	[21637 11]
	1	1.00	1.00	1.00		[34 26954]

Table B.354: Performance of *CNN* with the cic-ids2017_PS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21628 20]
	1	1.00	1.00	1.00		[20 26969]
2	0	1.00	1.00	1.00	1.00	[21635 12]
	1	1.00	1.00	1.00		[45 26944]
3	0	1.00	1.00	1.00	1.00	[21621 26]
	1	1.00	1.00	1.00		[15 26974]
4	0	1.00	1.00	1.00	1.00	[21623 24]
	1	1.00	1.00	1.00		[19 26970]
5	0	1.00	1.00	1.00	1.00	[21625 23]
	1	1.00	1.00	1.00		[23 26965]

Table B.355: Performance of *CNN* with *Hold Out* section of the cic-ids2017_PS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19034 21]
1	1.00	1.00	1.00		[11 23849]

Table B.356: Performance of *CNN* with the cic-ids2017_PS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21620 28]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21631 16]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	1.00	1.00	1.00	[21611 36]
	1	1.00	1.00	1.00		[17 26972]
4	0	1.00	1.00	1.00	1.00	[21614 33]
	1	1.00	1.00	1.00		[21 26968]
5	0	1.00	1.00	1.00	1.00	[21632 16]
	1	1.00	1.00	1.00		[25 26963]

Table B.357: Performance of *CNN* with the cic-ids2017_PS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21628 20]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21632 15]
	1	1.00	1.00	1.00		[39 26950]
3	0	1.00	1.00	1.00	1.00	[21618 29]
	1	1.00	1.00	1.00		[20 26969]
4	0	1.00	1.00	1.00	1.00	[21621 26]
	1	1.00	1.00	1.00		[29 26960]
5	0	1.00	1.00	1.00	1.00	[21632 16]
	1	1.00	1.00	1.00		[29 26959]

Table B.358: Performance of *CNN* with the cic-ids2017_PS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21629 19]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21631 16]
	1	1.00	1.00	1.00		[40 26949]
3	0	1.00	1.00	1.00	1.00	[21620 27]
	1	1.00	1.00	1.00		[20 26969]
4	0	1.00	1.00	1.00	1.00	[21616 31]
	1	1.00	1.00	1.00		[29 26960]
5	0	1.00	1.00	1.00	1.00	[21633 15]
	1	1.00	1.00	1.00		[25 26963]

Table B.359: Performance of *CNN* with *Hold Out* section of the cic-ids2017_PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19041 14]
1	1.00	1.00	1.00		[14 23846]

Table B.360: Performance of *CNN* with the cic-ids2017_PS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.99	1.00	1.00	1.00	[21648 0]
	1	1.00	1.00	1.00		[134 26855]
2	0	1.00	1.00	1.00	1.00	[21611 36]
	1	1.00	1.00	1.00		[36 26953]
3	0	1.00	1.00	1.00	1.00	[21598 49]
	1	1.00	1.00	1.00		[21 26968]
4	0	1.00	1.00	1.00	1.00	[21597 50]
	1	1.00	1.00	1.00		[32 26957]
5	0	0.99	1.00	0.99	1.00	[21648 0]
	1	1.00	0.99	1.00		[233 26755]

Table B.361: Performance of *CNN* with the cic-ids2017_PS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21613 35]
	1	1.00	1.00	1.00		[21 26968]
2	0	1.00	1.00	1.00	1.00	[21614 33]
	1	1.00	1.00	1.00		[35 26954]
3	0	1.00	1.00	1.00	1.00	[21615 32]
	1	1.00	1.00	1.00		[20 26969]
4	0	1.00	1.00	1.00	1.00	[21593 54]
	1	1.00	1.00	1.00		[23 26966]
5	0	1.00	1.00	1.00	1.00	[21607 41]
	1	1.00	1.00	1.00		[25 26963]

Table B.362: Performance of *CNN* with the cic-ids2017_PS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21626 22]
	1	1.00	1.00	1.00		[23 26966]
2	0	1.00	1.00	1.00	1.00	[21620 27]
	1	1.00	1.00	1.00		[38 26951]
3	0	1.00	1.00	1.00	1.00	[21591 56]
	1	1.00	1.00	1.00		[21 26968]
4	0	1.00	1.00	1.00	1.00	[21613 34]
	1	1.00	1.00	1.00		[37 26952]
5	0	1.00	1.00	1.00	1.00	[21606 42]
	1	1.00	1.00	1.00		[29 26959]

Table B.363: Performance of *CNN* with *Hold Out* section of the cic-ids2017_PS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19028 27]
1	1.00	1.00	1.00		[21 23839]

Table B.364: Performance of *CNN* with the cic-ids2017_PS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21620 28]
	1	1.00	1.00	1.00		[34 26955]
2	0	1.00	1.00	1.00	1.00	[21607 40]
	1	1.00	1.00	1.00		[39 26950]
3	0	1.00	1.00	1.00	1.00	[21598 49]
	1	1.00	1.00	1.00		[32 26957]
4	0	1.00	1.00	1.00	1.00	[21615 32]
	1	1.00	1.00	1.00		[40 26949]
5	0	1.00	1.00	1.00	1.00	[21615 33]
	1	1.00	1.00	1.00		[45 26943]

Table B.365: Performance of *CNN* with the cic-ids2017_PS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21617 31]
	1	1.00	1.00	1.00		[31 26958]
2	0	1.00	1.00	1.00	1.00	[21607 40]
	1	1.00	1.00	1.00		[43 26946]
3	0	1.00	1.00	1.00	1.00	[21605 42]
	1	1.00	1.00	1.00		[37 26952]
4	0	1.00	1.00	1.00	1.00	[21606 41]
	1	1.00	1.00	1.00		[40 26949]
5	0	1.00	1.00	1.00	1.00	[21606 42]
	1	1.00	1.00	1.00		[33 26955]

Table B.366: Performance of *CNN* with the cic-ids2017_PS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[21600 48]
	1	1.00	1.00	1.00		[23 26966]
2	0	1.00	1.00	1.00	1.00	[21609 38]
	1	1.00	1.00	1.00		[45 26944]
3	0	1.00	1.00	1.00	1.00	[21597 50]
	1	1.00	1.00	1.00		[35 26954]
4	0	1.00	1.00	1.00	1.00	[21624 23]
	1	1.00	1.00	1.00		[52 26937]
5	0	1.00	1.00	1.00	1.00	[21599 49]
	1	1.00	1.00	1.00		[30 26958]

Table B.367: Performance of *CNN* with *Hold Out* section of the cic-ids2017_PS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[19030 25]
1	1.00	1.00	1.00		[27 23833]

B.4.2 Results from the experiments with the custom dataset

Results for the *DoS* attack

Table B.368: Performance of *CNN* with the DoS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[1 25667]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.369: Performance of *CNN* with the DoS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[1 25667]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.370: Performance of *CNN* with the DoS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3031 14]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3043 2]
	1	1.00	1.00	1.00		[0 25667]

Table B.371: Performance of *CNN* with *Hold Out* section of the DoS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.372: Performance of *CNN* with the DoS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3043 2]
	1	1.00	1.00	1.00		[0 25667]

Table B.373: Performance of *CNN* with the DoS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.374: Performance of *CNN* with the DoS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3045 1]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3042 3]
	1	1.00	1.00	1.00		[0 25667]

Table B.375: Performance of *CNN* with *Hold Out* section of the DoS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.376: Performance of *CNN* with the DoS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3043 2]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3043 2]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.377: Performance of *CNN* with the DoS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3042 3]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25667]

Table B.378: Performance of *CNN* with the DoS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3041 4]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3040 5]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3044 2]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3040 5]
	1	1.00	1.00	1.00		[0 25667]

Table B.379: Performance of *CNN* with *Hold Out* section of the DoS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.380: Performance of *CNN* with the DoS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3043 2]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3046 0]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.381: Performance of *CNN* with the DoS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3044 2]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3042 3]
	1	1.00	1.00	1.00		[0 25667]

Table B.382: Performance of *CNN* with the DoS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3034 11]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3043 2]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3042 3]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3042 4]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3042 3]
	1	1.00	1.00	1.00		[0 25667]

Table B.383: Performance of *CNN* with *Hold Out* section of the DoS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2685 0]
1	1.00	1.00	1.00		[0 22650]

Table B.384: Performance of *CNN* with the DoS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3042 3]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3044 1]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3040 6]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3045 0]
	1	1.00	1.00	1.00		[0 25667]

Table B.385: Performance of *CNN* with the DoS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	1.00	1.00	1.00	[3041 4]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3041 4]
	1	1.00	1.00	1.00		[0 25668]
3	0	1.00	1.00	1.00	1.00	[3036 9]
	1	1.00	1.00	1.00		[0 25668]
4	0	1.00	1.00	1.00	1.00	[3040 6]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3043 2]
	1	1.00	1.00	1.00		[0 25667]

Table B.386: Performance of *CNN* with the DoS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	1.00	0.99	1.00	1.00	[3029 16]
	1	1.00	1.00	1.00		[0 25668]
2	0	1.00	1.00	1.00	1.00	[3034 11]
	1	1.00	1.00	1.00		[5 25663]
3	0	1.00	0.99	1.00	1.00	[3026 19]
	1	1.00	1.00	1.00		[1 25667]
4	0	1.00	0.99	1.00	1.00	[3030 16]
	1	1.00	1.00	1.00		[0 25667]
5	0	1.00	1.00	1.00	1.00	[3033 12]
	1	1.00	1.00	1.00		[0 25667]

Table B.387: Performance of *CNN* with *Hold Out* section of the DoS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	1.00	1.00	1.00	1.00	[2674 11]
1	1.00	1.00	1.00		[0 22650]

Results for the *PS* attack

Table B.388: Performance of *CNN* with the PS_a_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	0.90 1.00	1.00 0.99	0.95 1.00	0.99	[1411 0] [157 17141]
2	0 1	0.92 1.00	1.00 0.99	0.96 1.00	0.99	[1411 0] [115 17183]
3	0 1	0.89 1.00	1.00 0.99	0.94 1.00	0.99	[1411 0] [171 17127]
4	0 1	0.97 1.00	0.98 1.00	0.97 1.00	1.00	[1387 23] [49 17249]
5	0 1	0.91 1.00	1.00 0.99	0.95 1.00	0.99	[1410 0] [147 17151]

Table B.389: Performance of *CNN* with the PS_a_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	0.93 1.00	1.00 0.99	0.96 1.00	0.99	[1408 3] [105 17193]
2	0 1	0.97 1.00	1.00 1.00	0.99 1.00	1.00	[1411 0] [40 17258]
3	0 1	0.92 1.00	1.00 0.99	0.96 1.00	0.99	[1411 0] [118 17180]
4	0 1	0.96 1.00	0.98 1.00	0.97 1.00	1.00	[1387 23] [51 17247]
5	0 1	0.93 1.00	1.00 0.99	0.96 1.00	0.99	[1410 0] [110 17188]

Table B.390: Performance of *CNN* with the PS_a_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0 1	0.97 1.00	0.99 1.00	0.98 1.00	1.00	[1394 17] [42 17256]
2	0 1	0.97 1.00	0.98 1.00	0.98 1.00	1.00	[1388 23] [40 17258]
3	0 1	0.98 1.00	0.98 1.00	0.98 1.00	1.00	[1388 23] [35 17263]
4	0 1	0.97 1.00	0.98 1.00	0.97 1.00	1.00	[1378 32] [49 17249]
5	0 1	0.97 1.00	0.99 1.00	0.98 1.00	1.00	[1391 19] [38 17260]

Table B.391: Performance of *CNN* with *Hold Out* section of the PS_a_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.97	1.00	0.99	1.00	[1197 0]
1	1.00	1.00	1.00		[34 15277]

Table B.392: Performance of *CNN* with the PS_b_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.93	1.00	0.96	0.99	[1409 2]
	1	1.00	0.99	1.00		[105 17193]
2	0	0.92	1.00	0.96	0.99	[1410 1]
	1	1.00	0.99	1.00		[115 17183]
3	0	0.92	1.00	0.96	0.99	[1409 2]
	1	1.00	0.99	1.00		[117 17181]
4	0	0.92	1.00	0.95	0.99	[1407 3]
	1	1.00	0.99	1.00		[130 17168]
5	0	0.93	1.00	0.96	0.99	[1410 0]
	1	1.00	0.99	1.00		[110 17188]

Table B.393: Performance of *CNN* with the PS_b_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.94	1.00	0.97	0.99	[1406 5]
	1	1.00	0.99	1.00		[89 17209]
2	0	0.92	1.00	0.96	0.99	[1411 0]
	1	1.00	0.99	1.00		[115 17183]
3	0	0.92	1.00	0.96	0.99	[1409 2]
	1	1.00	0.99	1.00		[117 17181]
4	0	0.92	1.00	0.95	0.99	[1407 3]
	1	1.00	0.99	1.00		[130 17168]
5	0	0.93	1.00	0.96	0.99	[1410 0]
	1	1.00	0.99	1.00		[110 17188]

Table B.394: Performance of *CNN* with the PS_b_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.97	0.99	0.98	1.00	[1392 19]
	1	1.00	1.00	1.00		[42 17256]
2	0	0.97	0.98	0.98	1.00	[1387 24]
	1	1.00	1.00	1.00		[40 17258]
3	0	0.98	0.98	0.98	1.00	[1386 25]
	1	1.00	1.00	1.00		[34 17264]
4	0	0.97	0.98	0.97	1.00	[1381 29]
	1	1.00	1.00	1.00		[49 17249]
5	0	0.97	0.99	0.98	1.00	[1389 21]
	1	1.00	1.00	1.00		[38 17260]

Table B.395: Performance of *CNN* with *Hold Out* section of the PS_b_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.97	0.99	0.98	1.00	[1180 17]
1	1.00	1.00	1.00		[34 15277]

Table B.396: Performance of *CNN* with the PS_c_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.86	1.00	0.93	0.99	[1409 2]
	1	1.00	0.99	0.99		[220 17078]
2	0	0.86	1.00	0.92	0.99	[1411 0]
	1	1.00	0.99	0.99		[238 17060]
3	0	0.85	1.00	0.92	0.99	[1411 0]
	1	1.00	0.99	0.99		[254 17044]
4	0	0.85	1.00	0.92	0.99	[1404 6]
	1	1.00	0.99	0.99		[251 17047]
5	0	0.86	1.00	0.92	0.99	[1410 0]
	1	1.00	0.99	0.99		[232 17066]

Table B.397: Performance of *CNN* with the PS_c_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.86	1.00	0.93	0.99	[1409 2]
	1	1.00	0.99	0.99		[220 17078]
2	0	0.85	1.00	0.92	0.99	[1410 1]
	1	1.00	0.99	0.99		[254 17044]
3	0	0.85	1.00	0.92	0.99	[1411 0]
	1	1.00	0.99	0.99		[254 17044]
4	0	0.89	0.98	0.93	0.99	[1387 23]
	1	1.00	0.99	0.99		[170 17128]
5	0	0.86	1.00	0.92	0.99	[1410 0]
	1	1.00	0.99	0.99		[232 17066]

Table B.398: Performance of *CNN* with the PS_c_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.97	0.95	0.96	0.99	[1344 67]
	1	1.00	1.00	1.00		[42 17256]
2	0	0.97	0.96	0.97	0.99	[1355 56]
	1	1.00	1.00	1.00		[40 17258]
3	0	0.98	0.94	0.96	0.99	[1329 82]
	1	1.00	1.00	1.00		[34 17264]
4	0	0.96	0.94	0.95	0.99	[1327 83]
	1	1.00	1.00	1.00		[49 17249]
5	0	0.97	0.95	0.96	0.99	[1343 67]
	1	1.00	1.00	1.00		[38 17260]

Table B.399: Performance of *CNN* with *Hold Out* section of the PS_c_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.97	0.95	0.96	0.99	[1142 55]
1	1.00	1.00	1.00		[34 15277]

Table B.400: Performance of *CNN* with the PS_d_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.86	1.00	0.93	0.99	[1409 2]
	1	1.00	0.99	0.99		[220 17078]
2	0	0.85	1.00	0.92	0.99	[1407 4]
	1	1.00	0.99	0.99		[254 17044]
3	0	0.85	1.00	0.92	0.99	[1411 0]
	1	1.00	0.99	0.99		[254 17044]
4	0	0.85	1.00	0.92	0.99	[1408 2]
	1	1.00	0.99	0.99		[251 17047]
5	0	0.86	1.00	0.92	0.99	[1410 0]
	1	1.00	0.99	0.99		[232 17066]

Table B.401: Performance of *CNN* with the PS_d_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.90	0.99	0.94	0.99	[1397 14]
	1	1.00	0.99	1.00		[157 17141]
2	0	0.85	1.00	0.92	0.99	[1410 1]
	1	1.00	0.99	0.99		[254 17044]
3	0	0.85	1.00	0.92	0.99	[1411 0]
	1	1.00	0.99	0.99		[254 17044]
4	0	0.85	1.00	0.92	0.99	[1408 2]
	1	1.00	0.99	0.99		[251 17047]
5	0	0.86	1.00	0.92	0.99	[1410 0]
	1	1.00	0.99	0.99		[232 17066]

Table B.402: Performance of *CNN* with the PS_d_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.97	0.95	0.96	0.99	[1344 67]
	1	1.00	1.00	1.00		[42 17256]
2	0	0.97	0.96	0.97	0.99	[1355 56]
	1	1.00	1.00	1.00		[40 17258]
3	0	0.98	0.95	0.96	0.99	[1342 69]
	1	1.00	1.00	1.00		[34 17264]
4	0	0.96	0.94	0.95	0.99	[1327 83]
	1	1.00	1.00	1.00		[49 17249]
5	0	0.97	0.95	0.96	0.99	[1343 67]
	1	1.00	1.00	1.00		[38 17260]

Table B.403: Performance of *CNN* with *Hold Out* section of the PS_d_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.85	1.00	0.92	0.99	[1197 0]
1	1.00	0.99	0.99		[207 15104]

Table B.404: Performance of *CNN* with the PS_e_i dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.93	0.99	0.96	0.99	[1392 19]
	1	1.00	0.99	1.00		[105 17193]
2	0	0.93	0.99	0.96	0.99	[1392 19]
	1	1.00	0.99	1.00		[99 17199]
3	0	0.92	0.99	0.95	0.99	[1390 21]
	1	1.00	0.99	1.00		[117 17181]
4	0	0.91	0.99	0.95	0.99	[1390 20]
	1	1.00	0.99	1.00		[130 17168]
5	0	0.97	0.99	0.98	1.00	[1394 16]
	1	1.00	1.00	1.00		[38 17260]

Table B.405: Performance of *CNN* with the PS_e_ii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.93	0.99	0.96	0.99	[1392 19]
	1	1.00	0.99	1.00		[105 17193]
2	0	0.92	0.99	0.95	0.99	[1394 17]
	1	1.00	0.99	1.00		[115 17183]
3	0	0.92	0.99	0.95	0.99	[1390 21]
	1	1.00	0.99	1.00		[117 17181]
4	0	0.91	0.99	0.95	0.99	[1390 20]
	1	1.00	0.99	1.00		[130 17168]
5	0	0.93	0.99	0.96	0.99	[1394 16]
	1	1.00	0.99	1.00		[110 17188]

Table B.406: Performance of *CNN* with the PS_e_iii dataset

K	Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
1	0	0.97	0.94	0.96	0.99	[1331 80]
	1	1.00	1.00	1.00		[42 17256]
2	0	0.97	0.96	0.97	0.99	[1355 56]
	1	1.00	1.00	1.00		[40 17258]
3	0	0.98	0.94	0.96	0.99	[1329 82]
	1	1.00	1.00	1.00		[34 17264]
4	0	0.96	0.94	0.95	0.99	[1327 83]
	1	1.00	1.00	1.00		[49 17249]
5	0	0.97	0.94	0.96	0.99	[1331 79]
	1	1.00	1.00	1.00		[38 17260]

Table B.407: Performance of *CNN* with *Hold Out* section of the PS_e_iii dataset

Class	Precision	Recall	F1-Score	Accuracy	Confusion Matrix
0	0.97	0.99	0.98	1.00	[1181 16]
1	1.00	1.00	1.00		[34 15277]