UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

Machine Learning for Attack Tracking in Cyber-Physical Honeynets

Inês Morais Martins

Mestrado em Segurança Informática

Dissertação orientada por:
Prof. Doutor Alan Oliveira de Sá
Prof. Doutor Pedro Miguel Frazão Fernandes Ferreira

2025

Resumo

Proteger infraestrutura crítica contra ameaças cibernéticas é cada vez mais essencial no contexto de sistemas ciberfísicos. Para isso, o uso de honeynets para atrair atacantes e estudar as técnicas usadas por estes dentro de um ambiente controlado é crucial. Este trabalho explora a utilização de técnicas de Machine Learning para realizar deteção e classificação de ataques, e também explorar uma técnica para a correlação de ataques, com base nos datasets X-IIoTID e o SCVIC-APT-2021.

Para as várias tarefas de classificação, nomeadamente classificação binária (ataque vs. não ataque) e a classificação multiclasse (com 10 e 17 classes), foram avaliados vários modelos: XGBoost (XGB), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Logistic Regression (LR) e Naive Bayes (NB). Os modelos foram testados utilizando diferentes conjuntos de variáveis: 30 principais componentes do PCA, todos os componentes do PCA e todas as variáveis originais do dataset.

Os resultados obtidos demonstram que o uso do conjunto completo de variáveis originais proporciona o melhor desempenho nas várias tarefas de classificação, indicando que manter a totalidade da informação disponível favorece os modelos, enquanto a redução dimensional diminui a complexidade computacional mas causa perda de informação que se reflete negativamente na precisão da classificação.

Os modelos de ensemble, Random Forest e XGBoost, destacaram-se pelo seu desempenho superior, conseguindo capturar padrões complexos e lidar eficientemente com elevados volumes de variáveis. Estes modelos obtiveram as melhores métricas em todas as tarefas avaliadas, atingindo valores de accuracy superiores a 99% em vários cenários, especialmente quando empregaram todas as variáveis originais.

DT e KNN também apresentam bons resultados, mas acabavam por ficar aquém dos modelos de ensemble. DT apresenta uma ligeira quebra nos resultados da classificação das 17 classes, com dificuldades notórias na identificação de ataques como Fuzzing, TCP Relay, Reverse Shell e Command and Control. KNN, por sua vez, registou resultados ainda menos satisfatórios nestes ataques, ficando consistentemente abaixo do desempenho do DT.

LR e SVM apresentam resultados moderados, mas apresentam quebras significativas na classificação de certos tipos de ataque. Na tarefa com 10 classes verificaram-se dificuldades na identificação dos ataques como Exploitation e Command and Control. Na classificação de 17 classes, SVM falha completamente na classificação do ataque Fuzzing, independentemente do conjunto de variáveis usado, e LR apresentou igualmente fraco desempenho para esta classe. Ambos os modelos eviden-

ciaram dificuldades adicionais na classificação dos ataques TCP Relay, Reverse Shell, Command and Control e Discovering Resources.

NB é constantemente o pior modelo em todas as tarefas de classificação, independentemente do conjunto de variáveis usado, sendo que isto pode ser explicado pela premissa que este modelo faz de independência das variáveis do dataset.

Apesar da eficácia, este trabalho identificou desafios consideráveis no tratamento de dados, o forte desequilíbrio entre classes e a ausência de valores concentrada em colunas específicas. Ataques como MITM e Fake Notification têm representações tão reduzidas que foram removidos da análise. Outros ataques como Fuzzing, Reverse Shell, Command and Control e Ransomware também apresentam baixa frequência, o que afeta a capacidade dos modelos de generalizar e reconhecer padrões consistentes dessas classes.

Para além da tarefa de classificação, este trabalho propõe uma abordagem para a correlação de ataques aplicada ao dataset SCVIC-APT-2021. Esta abordagem começa com a clusterização dos ataques, assumindo que os ataques já estão classificados, usando K-Means com K=4, correspondente ao número de APTs presentes no dataset. Para o processo de clustering, foram considerados diferentes conjuntos de variáveis: o conjunto completo de variáveis originais, uma seleção manual e uma seleção automática baseada no algoritmo evolutivo NSGA-II.

A seleção de variáveis usando NSGA-II destacou-se por promover o equilíbrio entre a separação entre clusters e coesão interna, facilitando a identificação de grupos de ataques associados a campanhas de APT. Em contraste, o uso do conjunto completo das variáveis originais introduziu maior ruído nos resultados, e a seleção manual permite uma representação intermédia, com resultados mais consistentes do que as variáveis originais, mas menos eficazes que a seleção baseada em NSGA-II. Apesar das variáveis selecionadas usando NSGA-II apresentarem uma melhoria em relação aos outros cenários, não permite identificar um cluster dominante por APT.

Relativamente aos resultados obtidos com a seleção de variáveis via NSGA-II, Data Exfiltration é o ataque que obtém os melhores resultados na atribuição das APTs a diferentes clusters, apresentando clusters dominantes para as várias APTs, mas ainda apresenta divisão nos pontos da APT1 e APT2 entre diferentes clusters. Initial Compromise e Pivoting apresentam resultados satisfatórios, sendo que Initial Compromise apresenta clusters dominantes para as diferentes APTs, mas tem um cluster que contém pontos de todas as APTs; Pivoting apresenta clusters dominantes para duas das APTs, e as outras duas apresentam o mesmo cluster dominante. Por outro lado, Reconnaissance e Lateral Movement apresentam os piores resultados, sendo que Reconnaissance apresenta o mesmo cluster dominante para todas as APTs, e Lateral Movement tem duas APTs com pontos distribuídos por todos os clusters, não sendo capaz de apresentar um cluster dominante por APT.

De forma a validar a hipótese proposta de que centróides de clusters que sejam da mesma APT tendem a encontrar-se mais próximos entre si, e tendo em conta os resultados subóptimos obtidos no processo de clustering, para determinar os centróides dos clusters foi usada a média das variáveis numéricas, e a moda das variáveis categóricas.

Neste trabalho, inicialmente avaliou-se a coesão entre os centróides correspondentes à mesma fase da APT. Verificou-se que as fases de ataque Lateral Movement e Pivoting apresentam centróides com distâncias reduzidas entre si, indicando o uso de táticas semelhantes nas várias APTs. Initial Compromise apresenta distâncias relativamente próximas, indicando um grau de similaridade, mas não apresenta tanta coesão como Lateral Movement e Pivoting. Por outro lado, Reconnaissance e Data Exfiltration apresentam divergência com maiores valores de distâncias entre os centróides, indicando assim o uso de táticas distintas nas várias APTs.

Em relação à coesão dos centróides pertencentes à mesma APT, foi observado que Initial Compromise é a fase de ataque mais distante das outras fases, indicando um grau de separação entre Initial Compromise e as restantes fases. Em contraste, Pivoting é a fase que mais próxima se encontra das demais, indicando um grau de similaridade com as outras fases. Reconnaissance, Lateral Movement e Data Exfiltration apresentam distâncias intermédias entre as suas fases de ataque, evidenciando um grau moderado de coesão entre as fases de ataque.

Contudo, a fiabilidade dos resultados obtidos é limitada pelas características do dataset utilizado. Verifica-se um desequilíbrio na distribuição das APTs e das fases de ataque, com a APT1 e a fase Pivoting significativamente sobrerrepresentadas. Além disso, a ausência de detalhe sobre as técnicas específicas utilizadas e a curta duração temporal dos ataques dificultam uma análise aprofundada. Estes fatores restringem a generalização dos resultados e reforçam a necessidade de estudos futuros com datasets mais ricos, equilibrados e representativos.

Palavras-chave: Dataset, Ataque, Machine Learning, Classificação, Cluster

Abstract

Protecting cyber-physical systems against sophisticated attacks is essential. To that effect, honeynets can be deployed to lure attacker and study their techniques. This research delves into applying machine learning techniques for attack classification and tracking in cyber-physical honeynets, leveraging datasets such as X-IIoTID and SCVIC-APT-2021. The study evaluates a broad range of machine learning models, with ensemble-based methods like Random Forest (RF) and XGBoost (XGB) demonstrating superior performance due to their robustness and ability to model complex, high-dimensional relationships. XGB, in particular, provided a strong balance between accuracy and practicality, delivering consistent results across diverse attack types in both 10-class and 17-class multi-class classification tasks.

Furthermore, the study investigates the potential for APT tracking and correlation by applying clustering algorithms to the SCVIC-APT-2021 dataset. By analyzing distances between cluster centroids, the research aims to group attacks that belong to the same APT campaign, thereby enabling early-stage threat attribution.

The ability to classify attacks accurately and correlate them to broader threat patterns empowers security teams with predictive capabilities, allowing for proactive defense strategies and better response coordination. Overall, this thesis provides a comprehensive analysis of OT-focused cybersecurity datasets, presents effective ML-based intrusion detection approaches, and introduces a novel direction for APT correlation, contributing valuable insights for enhancing cyber resilience in industrial environments.

Keywords: Dataset, Attack, Machine Learning, Classification, Cluster

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contemporary Cyber-Physical Systems (CPS), such as Smart Systems, Industrial Control Systems (ICS), and water distribution systems, as well as critical infrastructures, include embedded computing elements, computer networks, and physical components that work together to monitor, control, and interact with their surrounding environment. This involves integrating Information Technology (IT) and Operational Technology (OT) environments, thereby broadening the attack surface.

As these systems become increasingly interconnected, they also become more vulnerable to multistage attacks, which are sophisticated and complex attacks launched in multiple stages. In these attacks, a series of carefully planned and coordinated tactics is used. The goal is to maintain long-term access to the target system. The process and techniques used vary based on the targets and specific goals. In CPS, multistage attacks can exploit both IT and OT vulnerabilities to gain control over physical processes, potentially causing widespread disruption, sabotage, or even damage to critical infrastructure. An example of a multistage attack is Advanced Persistent Threats (APTs), which is composed of several phases:

1. **Reconnaissance and Weaponization:** In this phase, information is gathered about the technical environment and the key personnel in the organization, and the tools and a plan for exploiting the vulnerabilities found are created.

2. **Delivery**: where the exploits are delivered to the target.

3. **Initial Intrusion**: Attacker establishes a foothold in the target, resulting typically in the installation of a backdoor.

4. **Command and Control (C&C)**: in which the attacker establishes a mechanism to take control of compromised devices, further exploiting the network.

5. **Lateral Movement (LM)**: where attackers move through the network to expand their control of the target.

6. **Data Exfiltration**: where attackers steal sensitive information to gain strategic benefits.

This makes APTs particularly dangerous to CPS environments because the attacker can maintain ongoing control over critical infrastructure without immediate detection, causing extensive damage or disruptions before the attack is discovered.

A well-known example of such a stealthy, multistage attack is the 2010 Stuxnet cyberattack against an ICS, which exploited several zero-day vulnerabilities targeting specific programmable logic controllers (PLCs) used to automate physical processes. The malware infected at least 14 industrial sites based in Iran, including a uranium enrichment plant, and disrupted the Iranian nuclear enrichment infrastructure [54].

Another example is BlackEnergy (2015), which was used in an attack against the power grid in Ukraine. The attack resulted in more than 225,000 customers being without power for over 6 hours. The tool was delivered using spear-phishing emails and weaponized Word documents containing a Trojan horse. After this, the attacker collected and used information about the ICS environment to disconnect substations from the grid [54].

As such, to reinforce the security of these systems, Honeynets can be deployed as a simulated network environment to lure and detect malicious activity. This allows insight into tactics and techniques used by attackers in a controlled environment. When applied to CPS, honeynets, besides simulating digital and network systems, also include elements that simulate the physical world, such as sensors, controllers, and actuators. To that effect, protocols that control physical processes, such as Modbus, DNP3, and Profibus, must be emulated within a cyber-physical (CP) honeynet.

Within this intricate landscape, the integration of mechanisms to detect, monitor, and analyze threats, with the ability to create alerts in real-time within CP honeynets, such as Intrusion Detection Systems (IDS), is essential. Integrating IDS within the CP Honeynet enables proactive threat mitigation, minimizing damage and disruption caused by cyberattacks.

IDS can be signature and anomaly-based. A signature-based IDS determines the existence of attacks in incoming traffic by comparing it to a specific pattern (signature). An anomaly-based IDS, in turn, detects abnormal behavior based on typical/expected user behavior. Anomaly-based methods can be statistical, knowledge, or machine learning (ML) based. While signature-based IDS is highly effective in identifying known threats with specific signatures, it struggles to detect new and emerging threats that don't match existing patterns. On the other hand, anomaly-based IDSs are more effective in identifying unknown threats but tend to create more false positives.

As a subcategory of signature-based IDS, ML is vital in identifying and learning patterns within large amounts of data. In this sense, the use of ML algorithms appears to be a suitable approach for analyzing and detecting attacks within CP Honeynets.

Modern computer networks generate vast amounts of data, which machine learning algorithms can efficiently analyze to identify threats and enhance detection accuracy by recognizing patterns that humans might overlook. These algorithms adapt to evolving threats by learning from new data, making them more effective at detecting previously unknown attacks and enabling real-time threat detection and response – crucial in CP honeynets, where delays can lead to serious

disruptions.

However, the efficacy of ML techniques is dependent upon the availability and quality of existing datasets. The utilization of these techniques necessitates robust datasets that encapsulate the diversity of potential cyber threats in OT environments.

1.1 Goals

With this in mind, the main goal of this project is to develop an IDS for CPS, capable of classifying a wide range of attacks and providing information about attack chains. This goal can be divided into the following sub-objectives:

1. Develop ML-based algorithms to accurately and efficiently detect and classify cyberattacks in CPS, in general, and in CP honeynets specifically. The approach involves the development of an ensemble of ML models trained on existing (*e.g.* X-IIoTID) datasets. Ensemble Learning allows the combination of multiple machine learning models to achieve superior performance compared to individual models. It offers a compelling solution for the nuanced demands of attack classification and tracking in CPS environments. By integrating diverse models, each with its strengths and specialization, ensemble methods enhance the robustness and accuracy of threat detection.

2. Test methodology to correlate attacks belonging to an attack sequence, making use of an approach based on clustering algorithms to aid in the detection of coordinated or evolving threats. Clustering allows the grouping of data points based on shared similarities. Measuring the distance between the centroids of these points will allow us to assess the degree of similarity between the clusters.

1.2 Contributions

The main contributions of this work are:

- Dataset Overview: Presents a comprehensive study of existing datasets for OT environments, analyzes the quality and diversity of the data present in the dataset, considering the balance of classes, and the representation of different attack vectors. Additionally, it includes an overview of ML techniques applied to some of this dataset, analyzing their effectiveness in detecting cyberattacks. The results of this study on the existing datasets for OT environments are published in [55].

- ML-IDS for classification of attacks: Development and implementation of ML-based IDS designed for the classification of attacks in CP Honeynets. Here, several ML algorithms, including an ensemble strategy, were tested and evaluated in their ability to identify and correctly classify the different attacks.

- Analysis of techniques for correlation of attacks within attack chains: Study a possible technique for the correlation of attacks within orchestrated attack sequences. This involves identifying and linking multiple related attack events to map a coherent sequence of actions taken by an attacker. For this purpose, a clustering algorithm is tested to obtain the best possible attack correlation inference based on the distance between different clustered attack data. This study also evaluates whether the selection of different feature sets can improve the quality of the clustering results.

1.3 Document Structure

The rest of this work is organized as follows:

1. Chapter 2 contains state-of-the-art regarding the existing OT, and attack-chain datasets, along with the ML algorithms applied to these datasets, and techniques applied to attack correlation.

2. Chapter 3 describes the methods used to develop and evaluate both the ML models for the classification of attacks and the technique for the attack correlation.

3. Chapter 4 includes a description of the datasets used in this work, as well as possible shortcomings of the datasets, and how the data were pre-processed.

4. Chapter 5 encompasses the results of the experiments conducted in the selected datasets, according to the methodology presented for the classification and correlation of attacks.

5. Chapter 6 consists of the final thoughts taken from the work done and possible future work.

Chapter 2

Related Work

This chapter presents the state-of-the-art, starting with the existing datasets in Section 2.1, beginning by presenting single attacks datasets in Section 2.1.1, followed by attack chain datasets 2.1.2. The next section 2.2 includes works of Machine Learning applied to the single attack datasets. In the last section 2.3 techniques for the correlation of attacks are presented.

2.1 Datasets

Datasets play a pivotal role in advancing the field of cybersecurity by providing realistic scenarios for analysis, testing, and development of security solutions. The convergence of IT and OT, particularly in critical infrastructure and industrial environments, has heightened the importance of datasets that accurately reflect the complexities and challenges of both domains.

IT datasets are crafted to simulate diverse cyber threats and attack scenarios encountered in traditional computer networks, being that IT datasets are more widely available, namely [64], and [53] that contain both conventional attacks and APT attacks. On the other hand, OT datasets are specifically designed to address the unique security concerns of ICS and critical infrastructures.

The creation of datasets for ICS environments has particular challenges. A crucial aspect is distinguishing between anomalies triggered by malicious attacks and those arising from benign operational behavior or equipment failures. The presence of noise and measurement errors can obscure genuine anomalies, leading to false positives or false negatives in IDS. These inaccuracies have the potential to hinder the system's ability to distinguish between normal operational variations and genuine security events, thereby compromising its effectiveness. Moreover, the wide variety of devices and protocols utilized in OT environments adds another layer of complexity. Each device and protocol may exhibit unique behaviors and patterns, making it challenging to establish universal anomaly detection criteria. As a result, the datasets must account for this diversity.

Another concern when producing datasets for OT environments regards privacy. The data generated by OT systems is highly sensitive, often containing proprietary information, operational parameters, and intricate system configurations. Unauthorized access to this data could lead to severe consequences. Therefore, organizations operating critical infrastructures are often reluctant

to share detailed OT datasets for research purposes due to the inherent risks associated with data exposure. As a result, the predominant approach in the field of OT data collection involves utilizing datasets derived from simulated environments [76] that do not include sensitive information.

To better understand the characteristics of the different available datasets in both IT and OT, this Section organizes the discussion in two parts: Section 2.1.1 describes datasets composed of attacks executed individually (i.e., not in an attack sequence); Section 2.1.2, in turn, describes datasets containing attacks executed in chains.

2.1.1 Single attack datasets

The discussion of datasets containing different attacks executed individually (i.e., not in attack chains) can start by the one made available in KDD-CUP-99[77]. It contains traffic from two weeks including four types of attacks: Denial-of-Service (DoS); Unauthorized access from a remote machine (R2L); unauthorized access to a local superuser (U2R); and probing. The training set presents 24 types of attacks, while the test data only presents fourteen. In the same direction, NSL-KDD [89] presents an improvement on the KDD-CUP dataset, where the duplicated values are removed, and the proportion of attacks presented in the test set is increased.

Another dataset example is the UNSW-NB15 [59], a synthetic dataset where pcap files are generated using tcpdump. The features were extracted using Bro-IDS (Zeek) and Argus, and algorithms in c#. The dataset contains 49 features, which are basic, content, and time features that correspond to gathered information from data packets: General purpose, connection, and labeled features. Regarding the labels of the entries in the dataset, they are 0 for normal and 1 for attack records, and also have a label regarding the type of attack; these can be Normal, Fuzzers, Analysis, Backdoors, Denial of Service (DoS), Exploits, Generic, Reconnaissance, Shellcode, and Worms.

In [12], the the ToN_IoT dataset combines information from pcap files, Zeek logs, sensor data, and OS logs. The data is labeled with normal and attack traffic, and which attack is being performed: Scanning Attack, Denial of Service, Distributed Denial of Services, Ransomware, Backdoor, Injection, Cross Site Scripting, Password Attack, and Man-in-the-Middle (MiTM). The features present in the dataset are Connection activity, Statistical activity, DNS, SSL, HHTP activity, and violation activity from the Zeek alerts.

SWAT [30] is another dataset provided by iTrust, the Centre for Research in Cyber Security at the Singapore University that uses a water treatment testbed, representing a small-scale version of a realistic modern Cyber-Physical system, integrating digital and physical elements to control and monitoring systems. The dataset contains 11 days of continuous operation, seven under normal operation, and four days under attacks; the data collected is network traffic and values from the 51 sensors and actuators labeled as normal and abnormal. The dataset contains 41 attacks of False Data Injection.

WUSTL-IIOT-2018 [78] is a dataset used for SCADA cybersecurity research that has presented five attacks: Port Scanner, Address scan, Device Identification, Device Identification (Aggressive Mode), and Exploit. The dataset shows 93.93% of regular traffic and 6.07% of attack

traffic. Its subsequent version, WUSTL-IIOT-2021 [91], is a dataset focused on Industrial Internet of Things (IIoT), with 53 hours of data collected. The dataset presents typical and attack scenarios, with 92.72% being typical scenarios. The attacks presented include command Injection Traffic, DoS, Reconnaissance, and Backdoor Traffic.

Electra[67] is an ICS dataset generated from the network traffic of an electric traction substation running in a usual scenario and under attack. The scenario of creation of the dataset as PLCs and a SCADA system that S7Comm and Modbus control. The dataset has presented Reconnaissance, False data injection, and Replay attacks.

Edge-IIoTset [25] is a cyber security dataset for IoT and IIoT. The IoT data is generated from various IoT devices and low-cost digital sensors for sensing temperature and humidity. The dataset presents 14 attacks, divided into 5 categories: DoS/Distributed Denial of Service (DDoS), Information gathering, MiTM, Injection, and Malware. Relative to DoS, it performed TCP SYN Flood, UDP flood, HTTP flood, and ICMP flood. Regarding information gathering, Port Scanning, OS Fingerprinting, and Vulnerability Scanning were performed. The MiTM Attacks class used DNS Spoofing and ARP Spoofing. In injection, cross-site scripting, SQL Injection, and Uploading. Finally, malware attacks use backdoor attacks, password cracking, and ransomware.

In the creation of the dataset ICS-Flow [21], a bottle-filling factory simulation includes ICS that controls the equipment within the factory, such as pipes, valves, a conveyor belt, and a water tank. Four attacks were implemented: reconnaissance attacks (IP-scanning and Port-scanning), Replay attacks, DDoS attacks, and MitM attacks (false data injection).

The dataset WDT Dataset [56] was acquired from a hardware-in-the-loop Water Distribution Testbed. This dataset contains both physical and network data. The acquisition of data is made into two different datasets: a physical one, which contains measures from sensors, valves, and pumps taken from the PLCs, and a network one, which contains the traffic from the SCADA network. The physical attacks present are water leaks by opening manual valves and sensors and pumps breakdown (blocking sensors and pumps). The cyber attacks are Man-In-the-Middle, more specifically ARP poisoning, which is an attack against the MODBUS communication protocol, Denial of Services, namely TCP and ICMP flood and Land attacks, and finally, Scanning attacks composed of SYN, FIN, Null, and XMAS scans. The network traffic was captured using Wireshark.

Another relevant dataset is the X-IIoTID [2], an up-to-date connectivity-agnostic and device-agnostic dataset created to detect intrusions in complex IIoT networks. There are nine types of intrusions in the X-IIoTID dataset: Reconnaissance, Command and Control(C&C), Weaponization, Exploitation, Tampering, Lateral Movement (LM), Exfiltration, Crypto Ransomware, RansomwareDoS (RDoS). These nine types of intrusions also contain various types within themselves, thus leading to 18 kinds of intrusions within the dataset. Zeek was used to extract information about network connections. After this, Batch and Python scripts were developed to parse collected data into Excel files. The features contain network and host-extracted features and OSSEC and Zeek alert logs.

Table 2.1 has an overview of the datasets discussed in this section. The table comprises the

Table 2.1: Datasets Overview

Dataset	IT	OT	Features			Categories of Attacks
			Network	Connection	Logs	
KDD-CUP-99[77]	✓			✓		4
NSL-KDD [89]	✓					4
UNSW-NB15 [59]	✓		✓	✓		9
ToN_IoT [12]	✓	✓		✓	✓	9
SWAT [30]		✓	✓			4
WUSTL-IIOT-2018 [78]		✓	✓			2
WUSTL-IIOT-2021 [91]		✓	✓	✓		4
Electra [67]		✓		✓		3
Edge-IIoTSet [25]	✓	✓	✓		✓	5
ICS-Flow [21]	✓	✓	✓			4
WDT [56]	✓	✓	✓			3
X-IIoTID [2]	✓	✓	✓	✓	✓	9

following information: It indicates if the dataset contains attacks in IT and/or OT; if the features present in the dataset include network features (such as protocol), connection features (such as connection duration), log features (such as process execution), and finally how many categories of attack exist.

Complementarily, Table 1 of Appendix .1 presents an overview of the number of attacks contained in various OT datasets. There, it is possible to observe that the existing datasets are, in general, imbalanced. It is important to note that the datasets WUSTL-IIOT-2018 and WUSTL-IIOT-2021 are not present in the table because, despite containing multiple categories of attacks, the datasets only allow for binary classification. The imbalanced nature of the datasets raises problems, as they may struggle to generalize across different attack categories due to the skewed representations.

Among these datasets, X-IIOTID arises as the one with the most extensive array of attacks, highlighting a diverse spectrum that surpasses other datasets in richness and variety. The frequency of attacks such as MitM, False Data Injection, and DoS is noticeable in several datasets, reflecting the prevalence of these threat vectors in OT environments. However, it is important to note the imbalanced nature of X-IIOTID, where certain attacks, particularly MitM and Fake Notification, are underrepresented.

2.1.2 Attack chains datasets

Attack chain datasets are few and mostly focused on IT systems. Despite their focus on IT environments, the datasets do not cover all phases of attack chains, and do not capture the complexity and sophistication of these cyberattacks, as discussed below.

The 1998 DARPA dataset [19] was developed to simulate the network traffic of an entire Air Force base. This simulation started by recording actual traffic from a local Air Force base to understand the types of services and traffic level. Scripted actors were designed to emulate different

roles, such as managers, secretaries, programmers, and system administrators, each performing their specific work tasks. The attacks present in the dataset ranged from spies who aimed to gather information and plant backdoors, to novice hackers who broke in and left, and malicious insiders. The attacks included surveillance/probing, denial-of-service, remote-to-root, and user-to-root.

The 2012 ISCX dataset [74] incorporated a wide range of sophisticated attack scenarios designed to mimic real-world malicious behavior. These included reconnaissance, vulnerability scanning, system exploitation, and post-exploitation techniques such as maintaining access and covering tracks. In this dataset, attackers would exploit known vulnerabilities in the Windows and Linux environments using tools like Metasploit and Slowloris to simulate remote access trojans (RATs), DoS attacks, and botnet-based distributed DoS attacks. It is important to note that this dataset contains only labels regarding the existence or non-existence of attacks.

More recent datasets such as the DAPT 2020 [61] controlled environments such as virtual machines (VMs) hosting vulnerable services and enterprise cloud network components were used to simulate APTs. The data collection process spanned five days, following a structured attack timeline. Initially, normal user traffic was generated to establish a baseline. Over the next few days, an internal Red Team simulated realistic APT attack phases, including reconnaissance, foothold establishment, lateral movement, and data exfiltration. The Red Team aimed to mimic real-world APT attacks while remaining stealthy to avoid detection by security tools, while the data was captured in the form of network traffic and host logs.

The Unraveled dataset [64] is a semi-synthetic dataset capturing an APT alongside 2 less skilled attacks. The dataset covers a six-week period, with Week 1 capturing normal employee behavior, followed by 5 weeks of attack data. Amateur Attacker falls bait for an SSH Honeypot deployed by the defender, Skilled Hackers attempt to penetrate the network through phishing but attempt gets blocked by the defender's security system, and APT Attackers for 5 weeks conduct an attack and successfully exfiltrating sensitive data. The APT of the attack includes reconnaissance (gathering network and employee information), establishing a foothold (deploying malware through targeted websites), lateral movement (gaining access to critical systems), data exfiltration (stealing sensitive information), and cover-up (erasing traces of the attack. The dataset is available in [63] here can be verified that the labels regarding to which group of attack, the attacks belong to namely the Skilled Attacker labels are missing.

The SCVIC-APT-2021 dataset [53] was produced using a multi-domain environment with two domains and four sub-networks to enable the simulation of Advanced Persistent Threat (APT) attacks, including lateral movement and pivoting. The setup uses Kali Linux to design attacks on machines, starting with initial victims in Domain 1. This domain consists of four machines, including a Domain Controller (DC) and three regular end devices. A VPN connects Domain 1 to Domain 2, which contains its own DC and a PC that serves as the ultimate target for credential extraction. The APT attacks in the dataset include: reconnaissance, initial compromise, lateral movement, pivoting, and data exfiltration.

The only public available dataset that contains APT attacks in CPS environments is CICAPT-

IIOT [29], these is semi-synthetic, designed to mimic APT behaviors while incorporating a variety of tools and devices that reflect the complexity of IIoT systems. The attack phases included in the dataset simulate a typical APT campaign, from data collection and exfiltration to maintaining persistence, evading defenses, and lateral movement across network components. The attack is carried out in a 'low and slow' manner, with actions executed in random intervals to reflect the stealthy and persistent nature of APTs. A distinctive characteristic of the dataset is its imbalance, with 99.5% of the data representing normal behavior and a small fraction representing malicious activity, reflecting real-world APT scenarios. This imbalance poses challenges for machine learning models, as oversampling techniques might distort the model's ability to detect APTs in real-world scenarios, where attacks are rare compared to everyday system behavior.

In Table 2.2 an overview of the existing datasets is presented. Here, the existing chain attack datasets, along with the number of instances of APTs and the number of phases of APTs present in the dataset, can be seen. For the existing datasets, SCVIC-APT-2021 emerges as the one with the highest number of instances of APTs and the second most complete in terms of phases of APTs present in the dataset.

Table 2.2: Datasets containing APTs Overview

Dataset	N of APTs	Phases of APT
CICAPT-IIOT [29]	1	6
SCVIC-APT-2021 [53]	4	5
Unraveled [64]	1	6
DAPT-2020 [61]	1	5
ISCX [74]	-	3
DARPA [19]	-	3

2.2 Machine Learning in single attack datasets

Several studies demonstrated the potential of applying ML algorithms to detect and classify single attacks present in available datasets. This section discusses the state-of-the-art of these ML-based approaches.

In [38] the authors present a detection model for DDoS attacks using the NSL-KDD and KDD-CUP-99 datasets. For feature selection, the correlation between all 41 features is measured, and the models used for the classification were K-Nearest Neighbors (KNN) and Naive Bayes (NB). The KNN model presents an accuracy of 98.51, and the NB model has an accuracy of 93.95. Other work using the KDD-Cup-99 dataset researchers [75] suggest employing a stacked non-symmetric deep auto-encoder to extract features coupled with a Random Forest (RF) during the classification phase. The results show that the model used in the 5-class can offer an average accuracy of 97.85%. The results of the "R2L" and "U2L" attacks are the lowest in accuracy. Regarding the NSL-KDD dataset, the 5-class classification model offered a total accuracy rate of 85.42%, "R2L" and "U2R" are again the lowest accuracy classification.

Regarding the SWAT dataset in [9], the usage of supervised Support Vector Machine (SVM), RF, KNN, unsupervised OCSVM (One-Class SVM), and Auto Encoder (AE) methods were analyzed. The models' accuracy using the dataset containing all sensor data was as follows: SVM: 0.9963, RF: 1, KNN:1, OCSVM: 0.8465, and AE: 0.8539. The results from the dataset that contains the sensor that is under attack are SVM: 0.9242, RF:0.9371, KNN:0.9316, OCSVM:0.8424, and AE:0.8539. In [44], the authors introduce a methodology centered around 1D Convolutional Neural Networks (1D-CNN). The F1-score of the ensemble of four layers in the 1D-CNN model was 0.9206, with a precision of 1 and a recall of 0.8529.

In another work, [24], a genetic algorithm-based feature selection method for ICS is proposed. This method integrates a feature ranking fusion mechanism in the genetic algorithm for eliminating redundant features, while also improving the global merit-seeking speed using the growing tree clustering idea. The models evaluated were RF, LR, DT, NN, NB, with NN and LR demonstrating superior performance.

In [87], A Dense Neural Network (DNN)-based model to detect the anomalies in the SWAT dataset, the model included a Long Short-Term Memory (LSTM) layer and 100 intermediate layers. The LSTM layer predicted the actuator's position based on its historical position. Each hidden layer was fully connected to an output layer with a bi-linear function. The cost function was defined by the cross-entropy of the actual and predicted probability distributions. The model detected 13 of the 36 total scenarios: 0.98295 for precision, 0.67847 for recall, and 0.80281 for F1 score. Still using the SWAT dataset, a 1D CNN model was constructed. It could detect 31 attacks, achieving 0.912 for precision, 0.861 for recall, and 0.886 for F1 score.

In [68], the Electra dataset is evaluated using the OCSVM, Isolation Forest, SVM, RF, and Neural Network models; the authors start by using Principal Component Analysis (PCA) and T-distributed stochastic neighbor embedding (T-SNE) and subsampling techniques due to the imbalance nature of the dataset. The SVM model achieves the highest performance, with precision and recall rates of 97.56% and 100%, respectively. Following closely is the RF model, demonstrating precision and recall rates of 98.77% and 98.71%. On the other hand, the DNN, a supervised model, yields comparatively lower results, with precision at 96.92% and perfect recall (100%). In semi-supervised models, OCSVM stands out with superior performance, boasting precision and recall rates of 98.62% and 98.56%. Meanwhile, the Isolation Forest exhibits a flawless recall (100%) but a slightly diminished precision compared to the other models under consideration, registering at 87.39%.

The Edge-IIoT dataset was evaluated in [25] using Decision Tree (DT), RF, SVM, KNN, and DNN for binary classification (exists attack or bot) and multiclass classification. In the binary classification, the highest accuracy, 99.99, was achieved by RF, SVM, KNN, and DNN, while the DT classifier achieved an accuracy of 99.98. For the 6-Class classification, DNN outperformed other classifiers with the highest accuracy of 96.01. On the contrary, the lowest accuracy was observed in the DT classifier at 77.90, followed by RF at 82.90, SVM at 85.62, and KNN at 83.39. In the case of the 15-Class classification, the DNN classifier again demonstrated superior

performance with the highest accuracy of 94.67. Conversely, the DT classifier had the lowest accuracy at 67.11, trailed by RF with 80.83, SVM with 77.61, and KNN with 79.18. In [23], a model based on a hybrid CNN-LSTM model exhibited an average accuracy of 97.85 in discerning benign from malicious traffic. In the 15-class classification, it demonstrated an average accuracy of 97.14. In [41], another CNN-LSTM is proposed both for binary and 7-class classification and compared with LSTM, Logistic Regression (LR), and NB. All models present 100 accuracy and precision in the binary classification; for the 7-class classification, the accuracy is 98.69.

In the original paper[21], the missing values are replaced with 0 to evaluate the dataset ICS-FLOW for preprocessing, and the PCA and the T-SNE are used for feature selection. The authors used the following models for evaluation: DT, RF, and Artificial Neural Networks (ANN), both for binary and multiclass classification. All algorithms identify attack flows with an accuracy greater than 99.4%, and the RF technique outperforms the others with an accuracy of 99.5% and a higher F1-score.

In [36] the integration of a denoising autoencoder (DAE) with LSTM units is studied as a robust security solution for intrusion detection in ICS networks. The evaluation is done using the ICS-Flow dataset. This reveals the DAE-LSTM model's superiority over LSTM, achieving high accuracy (99.6%), precision (98.2%), and recall (95.2%), in the task of binary classification.

Regarding the work already carried out using the X-IIOTID dataset, in the original paper [2] to evaluate the dataset, the authors apply PCA and T-SNE for feature selection. The models chosen were DT, NB, KNN, SVM, LR, and DNN, gated recurrent unit (GRU). The DT model performs better for all classifications: binary, 10-class, and 19-class classifications. Achieving an accuracy of 99.54%, 99.49%, and 99.45% respectively.

In [5] a binary and multiclass classification using deep learning architecture of CNN, LSTM, and CNN + LSTM generated from a hybrid combination of these. The accuracy results from the X-IIoTID dataset are 99.10, LSTM 99.05, and CNN+LSTM 99.84. In [3], clustering algorithms such as K-meloids and K-means are used for binary and multiclass classification. The results for K-Meloids in the binary classification were 99.79 accuracy, the 9-class classification has an accuracy of 100, and the 13-class classification has 99.85 for accuracy. For the K-means, the accuracy in binary classification is 99,76; in the 9-class, it has an accuracy of 97,069, being that the class with lower accuracy is lateral movement; in the 13-class classification, the overall accuracy is 96.38. The classification task (binary and multiclass) with the different datasets has been approached using several methodologies with promising results across different studies. Considering these findings, the possibility of using an ensemble approach that combines the strengths of different models from these studies can be explored. This approach, adopted in this thesis, can potentially lead to a more robust and efficient classification.

2.3 Correlation of attacks

The work in [83] uses a three-phase detection model leveraging semantic recognition and a state-based framework to detect APTs. The detection model focuses on identifying invariant behaviors

common to APT attacks, such as code deployment, sensitive data collection, and communication with Command and Control (C&C) servers. At the core of the framework is the aggregation of contextual information into states, which enables efficient detection without the need to store extensive historical data. This is achieved by capturing high-level semantics of data flows, control flows, and process behaviors, which are used to recognize suspicious actions. To support real-time detection and attack reconstruction, the framework stores system event logs as provenance graphs in memory. These graphs enable the tracking of activities and the reconstruction of attack chains when necessary. Predefined rules are used to monitor state transitions, allowing the system to track suspicious events.

The proposed model in [90] leverages a semi-supervised learning approach combined with complex network characteristics to detect APT activities. Once again a finite state machines was employed to model the state transitions of nodes during the attack process, allowing for the identification of suspicious hosts at different stages of the APT lifecycle. The framework leverages network traffic data, including DNS logs and network flows, to extract features such as vertex degrees and clustering coefficients, which are indicative of APT behavior. A semi-supervised learning approach is employed, utilizing labeled data from the Red Team dataset and unlabeled network traffic to train the model. The Shared Nearest Neighbor (SNN) clustering algorithm is used to group hosts based on their network behavior, with clusters assigned to different stages of the APT lifecycle using the state machine modeling.

In [32] is used Hidden Markov Models (HMM) to analyze the APT's, involving several steps. A Off-line Training Module is used to train the HMM parameters based on historical attack data. This module processes the pcap traces and extracts the relevant features needed to define the states, transition probabilities, and observation probabilities of the HMM. The resulting HMM configuration files, which include probability matrices and the number of states, are then used in the Prediction Module. This module applies HMM algorithms to real-time alert sequences from IDSs, enabling the system to infer the current state of an attack and calculate its probability. Further more the Viterbi algorithm, which identifies the most likely sequence of states given the observed alerts, is used. this allows the system to not only detect ongoing attacks but also predict their future steps.

In [51], HMM and a two-layer LSTM are evaluated using the NSL-KDD dataset and a Cowrie honeypot dataset presented by the authors. For the NSL-KDD dataset, the attack type label is simplified and clustered to serve as observation and hidden sequences for the HMM. This approach aligns with the synchronicity assumption of HMMs, where each observation corresponds to a hidden state at a given time. Similarly, for the Cowrie honeypot dataset, attackers are distinguished by their source IPs, and attack patterns are constructed using simplified features such as attack type and current state. The HMM is initialized with categorical states, starting in the "Normal" state, and the maximum likelihood principle is applied to determine the next phase. This method is particularly effective for modeling discrete, observable attack outputs. In contrast, the LSTM model is designed to capture more complex temporal relationships in time-series data,

DeepAG [50] is a framework designed to detect APT sequences and predict attack paths by leveraging system logs and advanced machine learning techniques. It utilizes transformer models to process semantic information from logs, enabling the detection of APT sequences through high-dimensional semantic vectors processed in parallel. For attack path prediction, DeepAG employs a bi-directional LSTM network: the Forward LSTM processes a log index sequence to generate predictions and probabilities, which serve as the starting point for multiple branches. Each branch is then reversed and fed into the Backward LSTM, and the final prediction is determined by averaging the likelihoods from both forward and backward directions. This bi-directional approach mitigates the bias of single-direction models, improving prediction reliability. Additionally, DeepAG constructs attack graphs to model non-linear dependencies among attack sequences, providing a comprehensive visual representation of potential attack strategies.

A real-time alert correlation and prediction framework composed of two models, one online and the other offline [70]. The offline phase, the framework preprocesses and aggregates low-level alerts into meta-alerts, reducing data volume while preserving critical security information. By analyzing historical data, causal relationships between meta-alerts are identified and used to construct the Bayesian Attack Graph, where nodes represent meta-alerts and edges represent causal dependencies, with conditional probability tables quantifying the likelihood of transitions between states. In the online phase, the framework leverages the BAG to predict the most probable next steps of an attacker in real-time. When new alerts are observed, their probabilities are updated and propagated through the BAG using forward and backward propagation techniques, allowing the system to dynamically adjust its predictions based on the latest evidence.

Chapter 3

Methodology

This chapter outlines the methodology used in this project to develop and evaluate ML models for IDSs in CPSs. The chapter is organized as follows: in section 3.1, an overview of the architecture is presented, along with a brief explanation of the models used; in section 3.2, a description of the experiences conducted is presented.

3.1 Architecture

An overall overview of the architecture is presented in Fig. 3.1. The architecture of this project comprises several modules. The first module is the classification, which receives the preprocessed Flow information. This block is composed of an ML-based attack classifier that labels the collected network data, detecting and classifying potential attacks.

The label from the classification serves as the input for the following block: Attack Tracking. This block is composed of an Attack Correlation and an Attack Prediction module. The Attack Correlation determines if a correlation exists between the labeled attacks received as an input stream. If a correlation exists, this information is passed to the Attack Prediction sub-model, which determines the attacker's next step with a certain probability.

In this work, only the first module, classification, is addressed, along with an analysis of techniques for attack correlation. For this purpose, two distinct datasets are used. A brief description of the techniques in both these modules will be presented in subsection 3.1.1 for classification, and for correlation in subsection 3.1.2.

3.1.1 Classification

To implement the classification module in our architecture, we explored various machine learning models, each with unique strengths and characteristics well-suited for intrusion detection. Next, a description of the models employed is provided, including XGB, RF, DT, KNN, SVM, LR, and NB. It is essential to note that the classification task is undertaken in relation to both binary and multi-class (10-class and 17-class) classification, and that this classification is performed non-hierarchically.

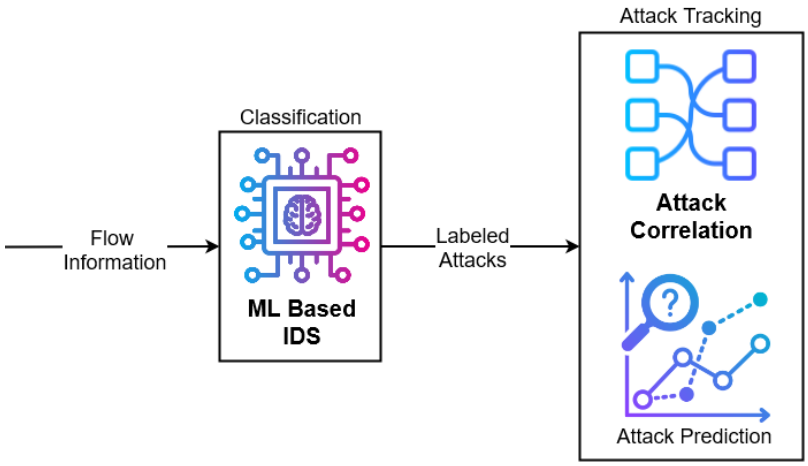


Figure 3.1: Overview of architecture

XGBoost [16] is an ensemble algorithm built upon weak learners, usually DT, where each learner corrects the errors previously made. The algorithm starts with a shallow tree and then calculates the error of each data point. A new tree is built that will focus on the misclassifications. The tree is added to the overall model, predictions are updated, and the process is repeated. The final prediction is the sum of the learner's predictions.

Random Forest (RF) [11] is a bagging (Bootstrap Aggregating), where each decision tree is trained on a different subset of the data, drawn randomly with replacement from the original dataset. This randomness helps to reduce variance, making the model less prone to overfitting compared to individual decision trees. Furthermore, a random subset of features is selected at each split in a tree, ensuring that the trees in the forest are more diverse. Once the ensemble of trees is built, the RF algorithm combines their predictions to make a final decision. For classification tasks, RF employs a majority voting system: each tree in the forest casts a vote for the class it predicts, and the class with the most votes is selected as the final prediction.

Decision Tree (DT) [11] is a supervised algorithm employed in regression and classification. This algorithm comprises decision nodes, where each represents a test on a specific feature of the input data. From each outcome of the decision node, a branch exists, and this process continues until a leaf node containing the class label is reached. When partitioning the data based on features, a purity criterion (Gini impurity or Entropy) is used to maximize the points from a single class in each node. DT provides a clear and intuitive way to visualize the data, making it easy to understand its prediction. However, this model tends to overfit, especially when allowed to grow deep without running, becoming overly complex and fitted to the training data. [57]

K-Nearest Neighbors (KNN) [11] is a supervised ML algorithm for classification and regression. This algorithm relies on the distance to measure the similarity between instances in the feature space. For new data points, calculate their distance to all the existing data points. The final

class is determined based on the k nearest neighbors and their class classification. KNN can be computationally expensive, especially in large datasets,[4], as it requires calculating distances to all data points. The model is also sensitive to the choice of K.[10]

Support Vector Machine (SVM) [79] is a supervised machine learning algorithm used for classification (SVC) and regression(SRV). SVM aims to find a hyperplane that maximally separates the different classes, the decision boundary, ensuring that the points of each class are as far as possible. The support vector is the class sample closest to the decision boundary, these vectors are essential in defining the position and orientation of the hyperplane. An unseen point is classified based on the distance to the decision boundary.

Logistic Regression (LR) [11] is a classification algorithm that models the relation between a dependent variable and one or more independent variables. The algorithm calculates the probability that a given instance belongs to the positive class, employing a logistic function. A point in the dataset is determined to belong to the positive class if the predicted probability is above or below a threshold. LR is especially useful for binary classifications, as the output is the probabilities that can be considered as the likelihood of class membership, making it highly interpretable. It is also computationally efficient, making it suitable for large datasets. However, LR uses linear decision boundaries, meaning that it is unsuitable for more complex patterns in the data,[31], and is also sensitive to outliers[42].

Naive Bayes (NB) [11] is a probabilistic classification based on the Bayes theorem, which is an assumption that features are independent of any other feature for a given class. Meaning that the presence or absence of a certain feature does not influence the presence or absence of other features for that class. The classifier calculates the posterior probability of each class based on the instance's features, and the class with the highest probability is selected. NB, is computationally efficient and easy to implement, making it ideal for large-scale problems or as a baseline model, the major weakness is that the assumption of feature independence, rarely holds true in real-world datasets. This can lead to suboptimal performance when features are highly correlated with each other.[72]

3.1.2 Correlation

Regarding the attack correlation, the K-means algorithm was tested for clustering, and Non-dominated Sorting Genetic Algorithm II (NSGA-II) was used for feature selection. The two techniques serve different but complementary roles in analyzing and detecting patterns within the data, with the goal of correlating attacks that belong to the same APT. Below a brief explanation on this techniques is presented.

K-Means: In the algorithm, data points are grouped in such a way that points within the same cluster are more similar to each other than to those in other clusters. The cluster's centroid, rep-

resenting the average position of all points in that group, serves as a reference for assigning data points to clusters. The algorithm starts with K initial centroids, and each point of the data is assigned to the nearest centroid, after this the centroid is recalculated by computing the man of the points in each cluster, and this is repeated until the value of the centroids is stabilized. The main advantages of this algorithm are its simplicity and computational efficiency, additionally using techniques such as k-means++ initialization can improve the algorithm's performance by intelligently choosing initial centroids.[6] Despite this, the algorithm needs the specification of the number of K clusters, k-means are sensitive to outliers as these can distort the cluster centroids, leading to poor clustering.[10]

Non-dominated Sorting Genetic Algorithm II (NSGA-II) [20] is a multi-objective optimization algorithm belonging to the evolutionary algorithm family. NSGA-II tries to find a set of optimal solutions the Pareto front, where no solution can be considered better than another without worsening at least one objective

This algorithm utilizes genetic principles, including selection, crossover, and mutation. It begins by initializing a potential population of solutions, and these solutions are then evaluated according to the defined objectives. These solutions are sorted into nondominated fronts; a solution is nondominated when it presents the best trade-off between all objectives. After sorting the solutions to maintain diversity NSGA-II uses a crowding distance to ensure diversity, this is done by measuring the distance between a solution and its neighbors in each objective, the boundary solutions are given infinite values so that they are preserved in the solution. During the selection process for the next generation, priority is given to the larger crowding distances, ensuring that a wide range of solutions are explored.

NSGA-II can converge the Pareto front while maintaining diversity among the solutions, however, it struggles with high-dimensional objective space, where ensuring diversity becomes more challenging. The pseudo-code of NSGA-II is described in Algorithm 1.

Algorithm 1: Pseudo-code of NSGA-II

```
P ← InitPopulation();
EvaluateFitness(P);
F ← NonDominatedSort(P);
AssignCrowdingDistance(F);
while not TerminationCondition() do
    Q ← Selection(P);
    Q ← CrossoverAndMutation(Q);
    EvaluateFitness(Q);
    R ← P ∪ Q;
    F ← NonDominatedSort(R);
    AssignCrowdingDistance(F);
    P ← SelectNextGeneration(F);
return ExtractParetoFront(P);
```

3.2 Experience Description

A description of the experiences conducted will be given next, starting with the classification task and followed by the correlation. As mentioned before, two different datasets were used for the classification task: the X-IIoTID [2] datasets are used, while for the correlation, the SCVIC-APT-2021 [53] datasets are used.

The choice of X-IIoTID dataset stems from its status as the most comprehensive dataset available for OT environments, as shown in Tables 1 and 2.1. However, it does not include complex attacks like APTs. Therefore, the SCVIC-APT-2021 dataset was selected for analysis of attack correlation techniques. SCVIC-APT-2021 is the most extensive dataset in IT containing APTs, offering the largest variety of APTs and the most complete representation of APT phases, as seen in Table 2.2.

3.2.1 Classification of attacks

This subsection outlines the methodology employed for classifying attacks into binary and multiclass categories (10-class and 17-class). The dataset was processed and classified using various feature sets and machine learning models.

The process begins with thorough data cleaning, where irrelevant and redundant features are removed to enhance data quality. Any missing values are handled systematically by imputing a value for each row to ensure completeness. Categorical variables are converted into numerical representations using one-hot encoding, allowing machine learning algorithms to process them effectively. Additionally, rare instances associated with certain underrepresented classes are excluded to avoid skewing the results. To prepare the data for modeling, an 80/20 stratified split is employed, ensuring a balanced distribution of class labels across both training and testing sets. The numerical data is then normalized to a standard range, which helps improve model performance by ensuring consistent feature scaling.

Three distinct feature sets are used for classification: a reduced set of principal components, the complete set of principal components, and the original feature set without any dimensionality reduction. The use of PCA for dimensionality reduction captures the most significant variance within the dataset, potentially improving model performance by focusing on the most informative features [2]. The classification is also performed on the full set of PCA components to evaluate the impact of dimensionality reduction on accuracy. The original feature set serves as a baseline for comparing performance across the reduced and full feature spaces.

A variety of machine learning models are employed for this classification task, namely XGB, RF, NB, DT, LR, SVM, and KNN [2], all of which are described in section 3.1.1. Hyperparameter tuning is conducted using grid search to optimize model performance. The search space is presented in Table 3.1, and k-fold cross-validation is used to ensure the models generalize well to unseen data. This cross-validation process reduces the risk of overfitting by testing the models on multiple subsets of the data.

Table 3.1: Hyper-parameters search space

Algorithm	Parameter	Values
DT	max_depth	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,12, 14, 15, 20, 30, 40, 50
	criterion	gini, entropy
KNN	k_range	[1, 19]
	weight_options	uniform, distance
LinearSVC	C	0.1, 1, 10, 100, 1000
	loss	hinge, squared_hinge
LR	C	1, 10, 100, 100
XGBoost	max depth	3, 4, 5, 7, 10, 12, 15
	learning rate	0.01, 0.1, 1
	gamma	0, 0.25, 0.5, 0.75, 1
	number of estimators	50 ,100, 300, 500
RF	max depth	4, 5, 6, 7, 8, 10, 12, 14, 16, 18, 20, 30, 40
	criterion	gini, entropy
	number of estimators	25, 50, 100, 150, 200, 300, 400, 500

Evaluation of Classification

The performance of each model and feature set combination is evaluated using a comprehensive set of metrics, including accuracy, precision, recall, F1-score, MCC, and confusion matrices. These metrics provide a well-rounded assessment of each model's ability to classify attacks accurately and effectively, ensuring that the models are not only precise but also capable of handling diverse and imbalanced data scenarios. This systematic methodology ensures the robustness of the classification system, optimizing it for both binary and multiclass classification tasks.

In understanding these metrics, it is essential to define the basic terms that underpin them:

- **True Positive (TP)**: Number of correctly predicted positive examples.
- **True Negative (TN)**: Number of correctly predicted negative examples.
- **False Positive (FP)**: Number of incorrectly predicted positive examples.
- **False Negative (FN)**: Number of incorrectly predicted negative examples.

Accuracy measures the proportion of correctly predicted observations (both positive and negative) out of the total observations. It is computed according to Equation 3.1:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Precision measures how many of the predicted positive cases are actually positive, as specified in Equation 3.2:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

Recall measures how many actual positive cases were correctly predicted, according to Equation 3.3:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

F1-Score is the harmonic mean of Precision and Recall, and it provides a single score that balances the two, as defined in Equation 3.4:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

Matthews's Correlation Coefficient (MCC) is a measure of the quality of binary classifications, considering all four confusion matrix categories (TP, TN, FP, FN). It returns a value between -1 and $+1$, where $+1$ indicates a perfect classification, 0 indicates no better than random classification, and -1 indicates total disagreement between predictions and actual values.

$$\text{MCC} = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.5}$$

3.2.2 Correlation of attacks

In this section, we explore the hypothesis that the clustered data from different attack stages belonging to the same APT group have centroids that are closer to each other, when compared to attacks from different APT groups. This involves applying K-means clustering to subsets of the dataset, each corresponding to a specific attack type, using various feature sets. This separation allowed for focused clustering within each attack type. The aim is to assess whether clustering patterns support the hypothesis of closer proximity among the attack stages originating from the same APT group.

The hypothesis suggests that attacks belonging to the same APT group will exhibit similar characteristics, which should manifest in the form of cluster centroids being spatially closer compared to those from different APT groups. To test this, the dataset was first divided into subdatasets for each type of attack, allowing focused analysis within each subset.

Clustering was performed using the K-means algorithm, with K corresponding to the number of distinct APT groups in the dataset. Additionally, it was evaluated how much the selection of specific features from the attack data could improve the clustering process results. Three feature selection strategies were employed to test the hypothesis: using all available features, a set of features selected based on domain knowledge, and features selected through GA optimization.

In the first scenario, K-means clustering was conducted using the complete set of features to provide a baseline. This approach gives a comprehensive view of the dataset but may include irrelevant or noisy data, potentially diluting clustering accuracy.

The second scenario involved using a carefully selected subset of features. These features were selected based on those that would enable tracking the route of the attacker and describing

the flow of the attacks.

The third approach employed NSGA-II to select optimal features by leveraging multi-objective optimization, and for that, the following approach was implemented. The features were converted into numerical values, where each value is a specific feature; for instance, 0 corresponds to the first feature, 1 corresponds to the second feature, and so on. Instead of the traditional binary approach, features are 0 (feature not present) and 1 (feature present). This choice was made to avoid the crossover between chains of 0's, which could result in a less effective exploration of the feature subset.

A mutation rate of 0.25 was used. This rate controls how frequently genes in the population are mutated to explore new potential solutions. For the crossover mechanism, a single-point crossover was employed. In this method, a random point was selected in the parent individuals, and the genes (feature numbers) were exchanged beyond this point. This helps in combining promising feature subsets from two parents to create new, potentially better-performing offspring.

The evaluation function was created, where, for each APT, it first calculates the distribution of attack points across several clusters, using the K-Means clustering algorithm. This involves determining which data points belong to which cluster. The result is transformed into percentages, representing the proportion of points from an APT that are located in each cluster. This distribution is stored for each attack type, effectively capturing how each APT spreads across the available clusters.

Next, all possible permutations of assigning APTs to clusters are tested, exploring different ways to attribute each attack type to a specific cluster. For each permutation, it calculates the sum of percentages that represent the alignment of APTs to clusters. The goal is to find the optimal assignment, the permutation of APT-to-cluster mappings that maximizes the total sum of these percentages.

This optimal assignment is considered the best match, where the APT types are most accurately represented by specific clusters. After determining the best permutation, the algorithm calculates the objective values for each cluster by subtracting the percentage of attack points in the assigned cluster from 1, reflecting the inverse of the percentage alignment to each APT. Algorithm 2, describes the fitness computation of the possible solutions, when selecting relevant attack features using NSGA-II.

The evaluation of these clustering scenarios began by correctly assigning the subsets of attacks to their respective APT groups. From this process, the most successful scenario was selected. Following this, centroid proximity analysis was performed, measuring the distances between the centroids of clusters within the same APT group and comparing them to the distances between clusters from different APT groups. This methodology aims to validate the hypothesis and identify the most effective feature sets for clustering.

The termination criteria were set to stop when no significant changes were detected in the EvaluateFitness function related to the best combination of features over the last x generations, indicating that the algorithm had most likely converged, meaning the population had stabilized,

and further evolution would be unlikely to yield better results. By focusing on the fitness of the best feature combination, the criteria ensure that the algorithm efficiently identifies a near-optimal solution without unnecessary computational overhead.

The lack of significant fitness improvements over multiple generations serves as a clear signal that the algorithm has exhausted its potential for further optimization. By defining x as a predetermined threshold, the criteria provide a structured and measurable endpoint, ensuring the algorithm concludes once it has sufficiently explored the solution space.

Algorithm 2: EvaluateFitness

Input: Input: APT data, Number of clusters k
Step 1: Distribution of attack points across clusters
foreach *APT in APT_List:* **do**
    Cluster_Assignments = KMeans(APT data, k)
    foreach *Cluster c in Clusters:* **do**
        Perc[APT][c] = #(Points in cluster c) / Total points for APT
    end
    Store Perc[APT]
end
Step 2: Permutation testing
foreach *Permutation P of possible APT-to-cluster assignments* **do**
    Alignment_Score = 0
    foreach *APT in APT_List:* **do**
        Assigned_Cluster = P[APT] Alignment_Score += Perc[APT][Assigned_Cluster]
    end
    if *Alignment_Score $>$ Best_Score* **then**
        Best_Score = Alignment_Score
        Best_Permutation = P
    end
end
Step 3: Objective value calculation
foreach *APT in APT_List* **do**
    Assigned_Cluster = Best_Permutation[APT]
    Objective_Value[APT] = 1 - Perc[APT][Assigned_Cluster]
end
Return Best_Permutation, Objective_Values

Chapter 4

Datasets

In this chapter, the details of the Datasets used in this work are presented, namely: X-IIoTID (in Section 4.1) and SCVI-APT-2021 (in Section 4.2). This includes a brief presentation of the datasets, the features present in each dataset, the challenges posed by the datasets, and the approach taken for data preprocessing.

4.1 X-IIoTID

X-IIoTID is a comprehensive, connectivity-agnostic, and device-agnostic intrusion dataset developed using the Brown-IIoTbed testbed, designed to represent real-world IIoT environments. Brown-IIoTbed integrates both legacy industrial devices like PLCs and modern IoT technologies. The testbed spans three key IIoT system tiers: the edge tier (including physical devices and edge computing), the platform tier (featuring cloud storage and analytics), and the enterprise tier (for service and application devices). The testbed also includes security measures, such as open-source IDS (OSSEC) for intrusion detection, and firewall protections at the edge and on attacker machines, including Kali Linux, both inside and outside the network, for attack simulation.

4.1.1 Features

This dataset contains 820834 records and 68 features, including network connection-related features extracted from Zeek logs using Batch and Python scripts. The data is then further processed using an Excel file, which incorporates Zeek and OSSEC alert logs. The host features are related to edge gateway resources, such as CPU utilization, memory load, I/O activity, system load, processes, and context switches. The target variables class1, class2, and class3 represent a binary classification indicating whether an attack exists or not, class2 indicates the class of the attack, and class3 is the subclass of attacks. In Appendix .2, a brief description of features found in the X-IIoTID dataset is given.

4.1.2 Missing Values

As can be observed in Fig. 4.1, the missing values are concentrated in the features Duration, Src bytes, Des bytes, Src pkts, Src ip bytes, Des pkts, Des ip bytes, and in the features generated from

these features, namely total bytes, total packets, packet rate, byte rate, Scr pkts ratio, Des pkts ratio, Scr bytes ratio, and Des bytes ratio.

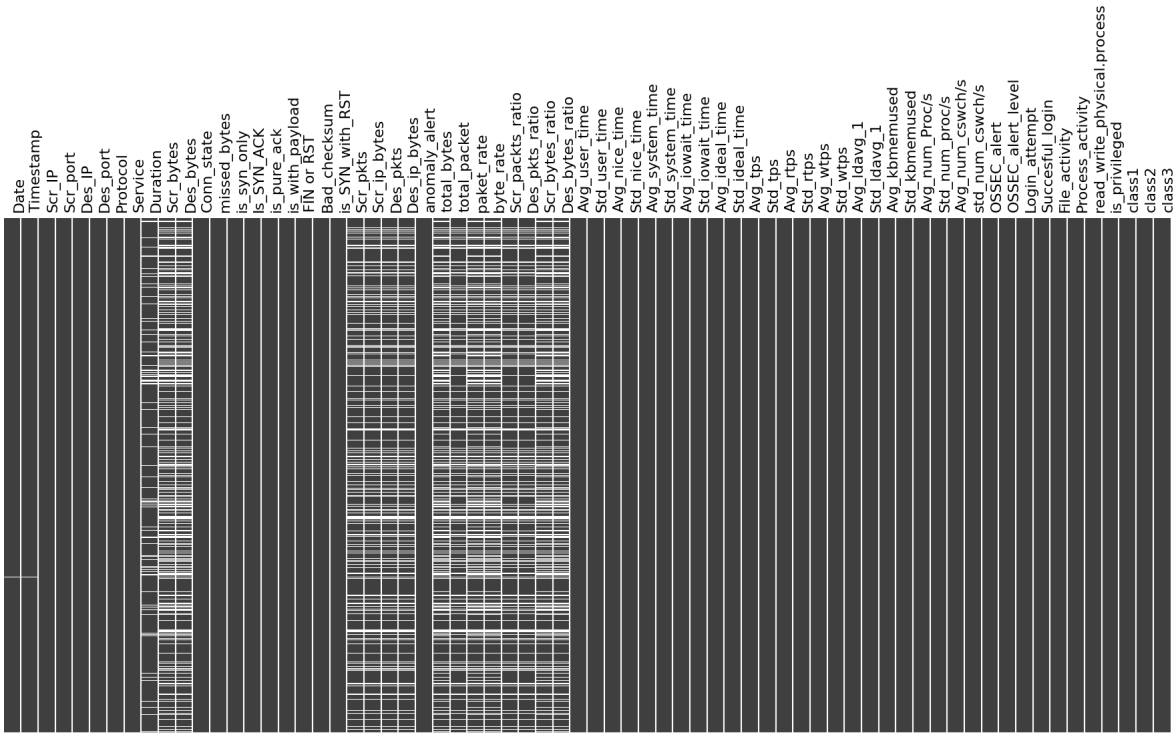


Figure 4.1: X-IIoTID missing values

4.1.3 Challenges

One of the challenges with this dataset is the class imbalance, especially regarding the subclass of attacks, as can be observed in 1. Attacks such as MitM (117) and Fake Notification (28) have very little representation in the dataset, and as such, they were removed. Then we have attacks such as Fuzzing (1313), Reverse Shell(1016), Ransomware (458), C&C (2863), and TCP Relay (2119) that, despite having more representation than the previously mentioned attacks, still have significantly less representation than RDOS(141261), Generic Scanning (52277), and Scanning Vulnerability (52852). This imbalance presents a challenge for machine learning models, as they tend to be biased towards the more frequent classes, impacting the model's ability to generalize across underrepresented attacks.

Another challenge associated with this dataset is the presence of missing values, as previously mentioned. As shown in Fig. 4.1, a considerable number of missing values are present, and they tend to be concentrated within certain columns. This distribution of missing values requires careful preprocessing to avoid skewing the model's performance. Handling these is crucial for ensuring the integrity of the dataset and the reliability of results.

4.1.4 Data Preprocessing

Regarding our work with this dataset, we started by cleaning and transforming the data into a suitable format for training the models. This process initiates with strategically removing specific columns, including 'Date', 'Scr_IP', 'Des_IP', 'Scr_port', 'Des_port', and 'Timestamp'. The missing values are systematically handled by replacing them with the mean of the respective column. Categorical features transformed using the get_dummies method. Furthermore, due to the limited number of instances, rows associated with the class 3 classification of MitM and Fake Notification are excluded from the dataset.

After this, the target classes are isolated and stored in variables before being removed from the main dataframe. The resulting dataset comprises 738,577 rows and 80 columns. To facilitate robust model training, the data undergoes Stratified Splitting based on class3, allocating 80% for training and 20% for testing. Following this split, numerical data normalization is performed, constraining values to an interval of [-1, 1] using the StandardScaler() function. The subsequent step involves applying PCA for feature selection and comparing these results with PCA applied to all components and to all features without PCA.

4.2 SCVI-APT-2021

In the following section, the SCVI-APT-2021 is presented, including a description of the features present in the dataset, as well as a representation of the attacks that exist within it. This dataset was created in a multi-domain environment designed to simulate APT attacks. The environment includes two domains and four sub-networks, enabling scenarios such as lateral movement and pivoting. Attack simulations are carried out using Kali, which targets machines in Domain 1, allowing for initial access to victims. Domain 1 comprises four machines, including a Domain Controller (DC) and three regular end devices. The two domains are connected via a virtual private network (VPN), with Domain 2 containing another DC and a regular PC, the ultimate target for credential extraction by the attacker.

4.2.1 Features

This dataset comprises 259120 entries and 84 columns, including both numerical and categorical features, as well as the target class, which categorizes the attacks present in the dataset. A brief description of the features present in SCVI-APT-2021 is provided in Appendix .3.

4.2.2 Attacks

This dataset contains APTs that encompass multiple stages of sophisticated cyberattacks, including initial compromise, reconnaissance, lateral movement, data exfiltration, and pivoting. Each stage represents a critical phase in the APT lifecycle. Table 4.1 provides a representation of the techniques used in each attack.

Table 4.1: APT Attack techniques

APT Phase	Technique
Initial Compromise	VSFTPD*
Reconnaissance	Active Scanning
	Gather Host Information
	Gather Network Information
Lateral Movement	Pass the Hash/Ticket
	Remote Desktop Protocol
	VMI
Pivoting	AutoRoute
	Socks4a
	Proxy Chain
Data Exfiltration	DNS Tunnelling
	C2 Tunnelling
	Encode and Encrypt

Table 4.2 provides an overview of the dataset's distribution across various APTs and other types of attacks. It can be observed that there is an imbalance in the distribution of points across APTs and their associated attack stages. APT1 emerges as the most represented, with 1911 entries, which is more than twice the number of entries in APT3 (1056) and three times the points of APT2 (623 points) and APT4 (693 points). This disparity highlights a skewed representation, where APT1 dominates the dataset.

Within the attack stages, Initial Compromise is the stage least represented in the dataset. In contrast, Pivoting is the most represented attack stage. For instance, APT1 has only 19 entries in the Initial Compromise stage, which is a stark contrast to the Pivoting stage, where APT1 alone has 862 entries. The same pattern is observed in the other APTs. Additionally, while Data Exfiltration is an attack stage that is underrepresented across various APTs, with 10 entries in APT2 and 12 in APT3, it is notable that this trend does not apply to APT1, which exhibits a more balanced representation.

Fig. 4.2 allows to observe how the different APTs are distributed over the timeline. It highlights key information, such as the source and destination IP addresses of the attack, as well as the type of attack being performed. It's also evident from the figure that the 4 APTs occur within the same day, spanning a period of 13 hours. Notably, the APTs are consistently separated over a few hours. Additionally, the figure shows that the source and destination IPs, as well as the techniques employed, remain largely the same across the various APTs. This extends to the order of the techniques utilized during the attacks across the various APTs.

In Fig. 4.3, a visual representation of how the several APTs attacks were executed within the network is presented. It is possible to observe that APT1 and APT4 have an identical attack chain throughout the network, following the same sequence of actions as they navigate through it, while APT2 and APT3 exhibit some minor variations in their attack chains.

Table 4.2: Attack Distribution in each APT

APT	Phases of APT	N of points	Total of points
1	Initial Compromise	19	1911
	Reconnaissance	322	
	Pivoting	862	
	Lateral Movement	241	
	Data Exfiltration	467	
2	Initial Compromise	17	623
	Reconnaissance	123	
	Pivoting	324	
	Lateral Movement	149	
	Data Exfiltration	10	
3	Initial Compromise	19	1056
	Reconnaissance	229	
	Pivoting	583	
	Lateral Movement	213	
	Data Exfiltration	12	
4	Initial Compromise	18	693
	Reconnaissance	159	
	Pivoting	353	
	Lateral Movement	125	
	Data Exfiltration	38	

4.2.3 Challenges

The dataset presents several challenges due to the imbalance in the distribution of attacks across the different APTs. Some APTs have a significantly higher number of attack instances compared to others. For instance, APT1 has a disproportionately large number of entries when compared to the other APTs. The same applies to the different stages of attacks, both within the dataset and the APTs they are associated with. Phases, such as Initial Compromise, are underrepresented, while others, like Pivoting, may have a higher frequency of entries.

Another limitation of the dataset is the lack of detailed information about the specific techniques used in the attacks. Although the dataset captures various stages of APTs, it does not provide granular details on the exact attack techniques used in each dataset entry. This missing information makes it harder to conduct an in-depth analysis of how attackers exploited the vulnerabilities in the network presented.

Additionally, the APTs in this dataset exhibit similar behaviors across different attack sequences and occur within a small network environment. This limits the variety of attack strategies and tactics seen, as attackers in larger, more complex environments would likely employ more diverse approaches to evade detection. The small network size can also restrict the type and complexity of attacks that would manifest differently in a larger-scale network.

In terms of time distribution, while the dataset is well-distributed along the timeline, the timeline itself is relatively short. Multiple APTs occur within the same hours, compressing the attack activities into a condensed period. This may not fully reflect the longer-term persistence typical

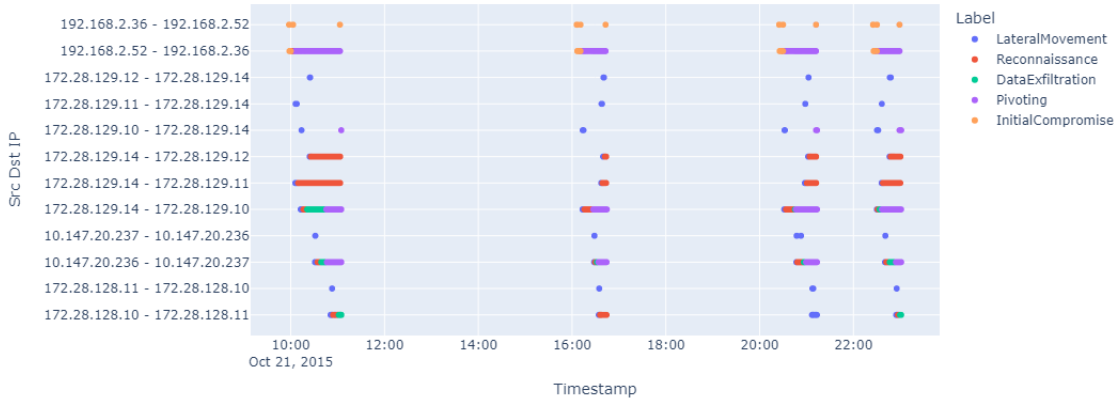


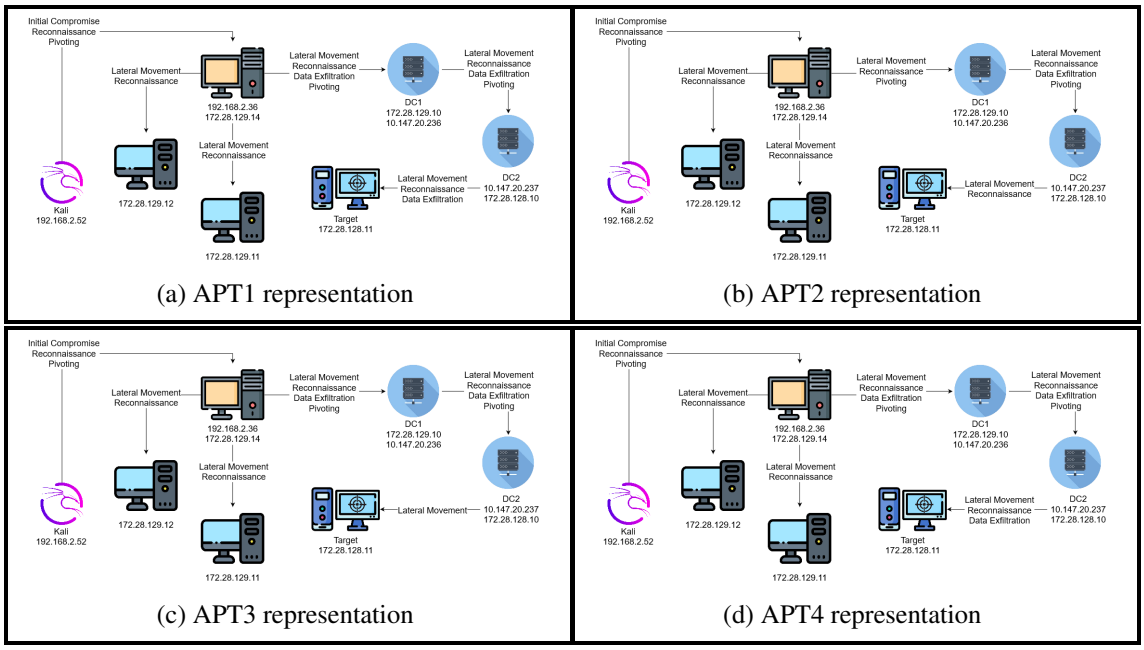
Figure 4.2: APTs distribution over time

of real-world APT campaigns, which often unfold over weeks or months. The short timeline may thus pose limitations on how effectively time-based analysis or event correlation can be conducted.

4.2.4 Data Prepocessing

The process begins by aligning the APTs to a common timeline, since they are initially distributed across distinct timeframes. The attack labels are stored in a separate variable for later use. Columns such as ”Flow ID” and ”Label” are removed from the dataset, and any rows containing null values are dropped.

Categorical features, including ”Protocol”, ”Src IP”, ”Dst IP”, ”Src Port”, ”Dst Port”, and ”Real Timestamp”, are transformed using one-hot encoding via the get_dummies() function, and numerical features are normalized using the StandardScaler() method. The dataset is then divided based on the previously saved attack labels, and each subset is clustered using the K-means algorithm with K = 4.



(a) APT1 representation

(b) APT2 representation

(c) APT3 representation

(d) APT4 representation

Figure 4.3: Representation of APTs in the network

Chapter 5

Results and Discussion

This chapter presents the results of the experiences conducted regarding the classification and correlation. In section 5.1, the attack classification results from the selected models across several scenarios after using Grid Search and being trained with X_train and tested with X_test are presented. The evaluation utilized the X-IIOTID dataset and included accuracy, precision, recall, F1-scores, both for the models and for each class, as well as the MCC and the Confusion Matrix for some of the selected models. Section 5.2 presents the results from the clustering using the different scenarios from the SCVI-APT-2021 dataset, and finally, presents an analysis of the distance between the centroids computed as an attempt to infer the possible correlation between different attack stages belonging to the same APT.

5.1 Attack Classification

This section highlights the results obtained in the classification task, specifically in the different scenarios described previously, using various feature sets: PCA with 30 components, All Components, and All Features. First, the results regarding the binary classification are presented in Subsection 5.1.1, followed by the multiclass classification, 10-class classification in Subsection 5.1.2, and finally the 17-class classification in subsection 5.1.3. In the last subsection 5.1.4, a brief discussion of the classification results is presented, along with a comparison with the original results from[2]. Note that, as shown in Fig. 5.1, the PCA analysis has an explained variance of 98.17 with 30 components.

5.1.1 Binary Classification

Table 5.1 presents the results for the algorithms tested in the context of binary classification, using three scenarios: PCA with 30 components, all PCA components, and all features. These models were trained using the hyperparameters resulting from the grid search with cv = 7 presented in Appendix .6 Table 4.

In the binary classification task, the RF model stands out with the highest accuracy across all scenarios studied, with the scenario with all features obtaining the highest accuracy, precision, recall, and F1-score of 99.72. It also has the highest MCC of 99.45%, indicating a strong correlation

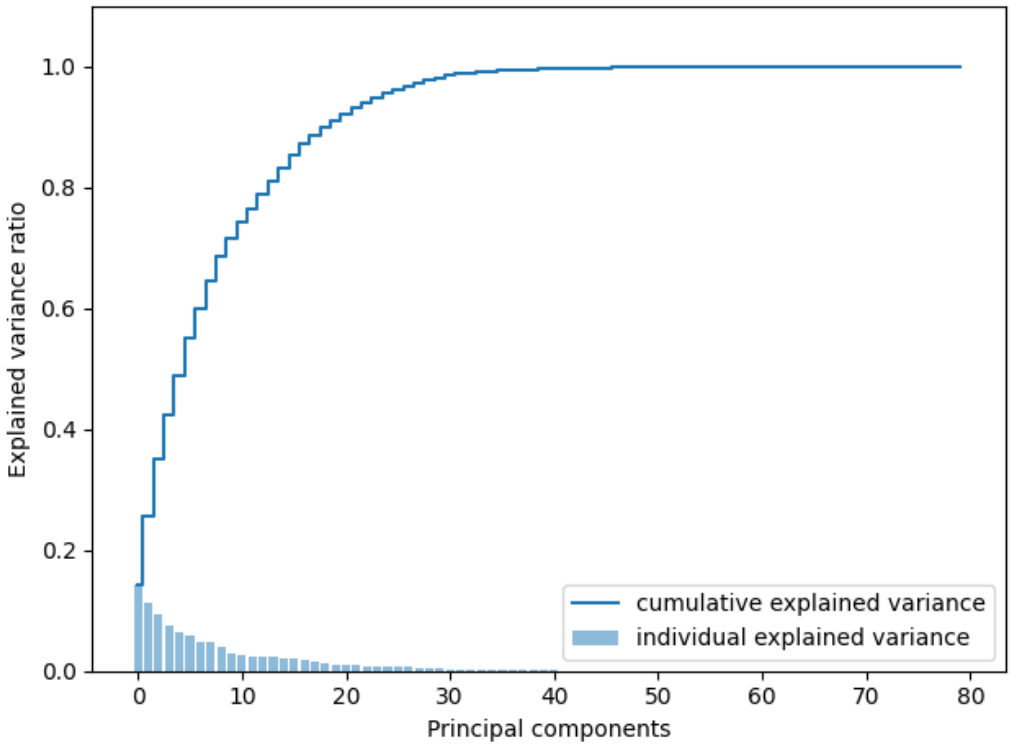


Figure 5.1: Results from PCA

Table 5.1: Results from binary Classification

Classification	Model	Accuracy	Precision	Recall	F1-Score	MCC
PCA 30-Components	SVM	91.50	91.51	91.50	91.50	83.13
	DT	96.88	96.89	96.88	96.88	93.77
	LR	91.68	91.72	91.68	91.67	83.47
	KNN	96.60	96.60	96.60	96.60	93.20
	NB	80.39	80.83	80.39	80.35	61.25
	XGB	97.09	97.14	97.10	97.09	94.23
	RF	**98.13**	**98.16**	**98.13**	**98.13**	**96.28**
PCA All-Components	SVM	97.96	98.00	97.96	97.96	95.95
	DT	99.00	99.01	99.01	99.01	98.01
	LR	97.79	97.82	97.79	97.79	95.60
	KNN	98.36	98.36	98.36	98.36	96.72
	NB	90.33	90.34	90.34	90.34	80.67
	XGB	99.22	99.23	99.23	99.23	98.45
	RF	**99.42**	**99.43**	**99.41**	**99.42**	**98.84**
All-Features	SVM	97.96	98.00	97.96	97.96	95.95
	DT	99.63	99.63	99.63	99.63	99.25
	LR	97.76	97.79	97.76	97.76	95.55
	KNN	98.36	98.36	98.36	98.36	96.71
	NB	88.97	89.70	88.97	88.94	78.70
	XGB	99.71	99.71	99.71	99.71	99.42
	RF	**99.72**	**99.72**	**99.72**	**99.72**	**99.45**

between predicted and actual classes. This demonstrates that RF is highly effective in capturing the nuances of the data, especially when all features are considered. XGB is the second-best model across the various scenarios, showing strong performance with accuracy scores of 97.09%, 99.22%, and 99.71% for the 30-Component, All-Components, and All-Features scenarios, respectively. Its performance closely follows RF, particularly in the All-Features scenario, where it reaches 99.71% in accuracy, precision, recall, and F1-score, with an MCC of 99.42%. This suggests that both RF and XGB are highly robust models for binary classification, particularly when a more comprehensive feature set is employed.

DT performs quite well, especially in the All-Features scenario, where it achieves 99.63% accuracy and an MCC of 99.25%. However, it shows a noticeable drop in performance when fewer components are used (30 Components), with an accuracy of 96.88%. DT models are prone to overfitting, which could explain the performance variance across scenarios. KNN performs consistently across all scenarios with slight variations. It achieves 98.36% accuracy in both the All-Components and All-Features scenarios, while its performance in the 30-Components scenario is slightly lower at 96.60%. KNN tends to perform well in high-dimensional spaces but may struggle with noisy or less relevant features, which may explain why it does not outperform RF or XGB.

LR performs moderately well, with its best accuracy at 97.79% in the All-Components scenario. However, its performance drops in the 30-Components scenario, achieving only 91.68% accuracy. LR struggles with complex data relationships and feature interactions, which limit its effectiveness in this task compared to more advanced models like RF and XGB. SVM shows a reasonable performance across the scenarios, achieving the best accuracy of 97.96% in both the All-Components and All-Features scenarios. It shows a more significant drop in performance in the 30-Component scenario, with an accuracy of 91.50%. SVMs are powerful classifiers, but they can be sensitive to the choice of kernel and the scale of the data.

NB is the weakest model overall. It achieves only 88.97% accuracy in the All-Features scenario and 80.39% accuracy in the 30-Components scenario, which is significantly lower than the other models. NB's assumption of feature independence may not hold well in this dataset, explaining its poor performance compared to more sophisticated models.

Fig. 5.2 presents the confusion matrix (CM) of the best models in each scenario (RF in all the scenarios), along with the overall second best classification model (XGB with All Features). Starting with RF models when trained with 30 PCA Components, it's possible to observe that there is a small number of Normal instances being classified as Attack (0.6761%), on the other hand, there are a significant amount of Attacks being classified as Normal (3.126%), but when Attack is predicted by the models, only 0.7302% are actual Normal traffic.

In the RF model with All Components, the number of Normal instances being classified as Attacks is smaller than in the previously presented model (0.2139%), the same is observed for the number of Attacks being classified as Normal Traffic (0.9644%), and the number of Attacks predicted by the model that are Normal (0.2271%).

The same behavior is observed with the RF models trained with all Features present, having

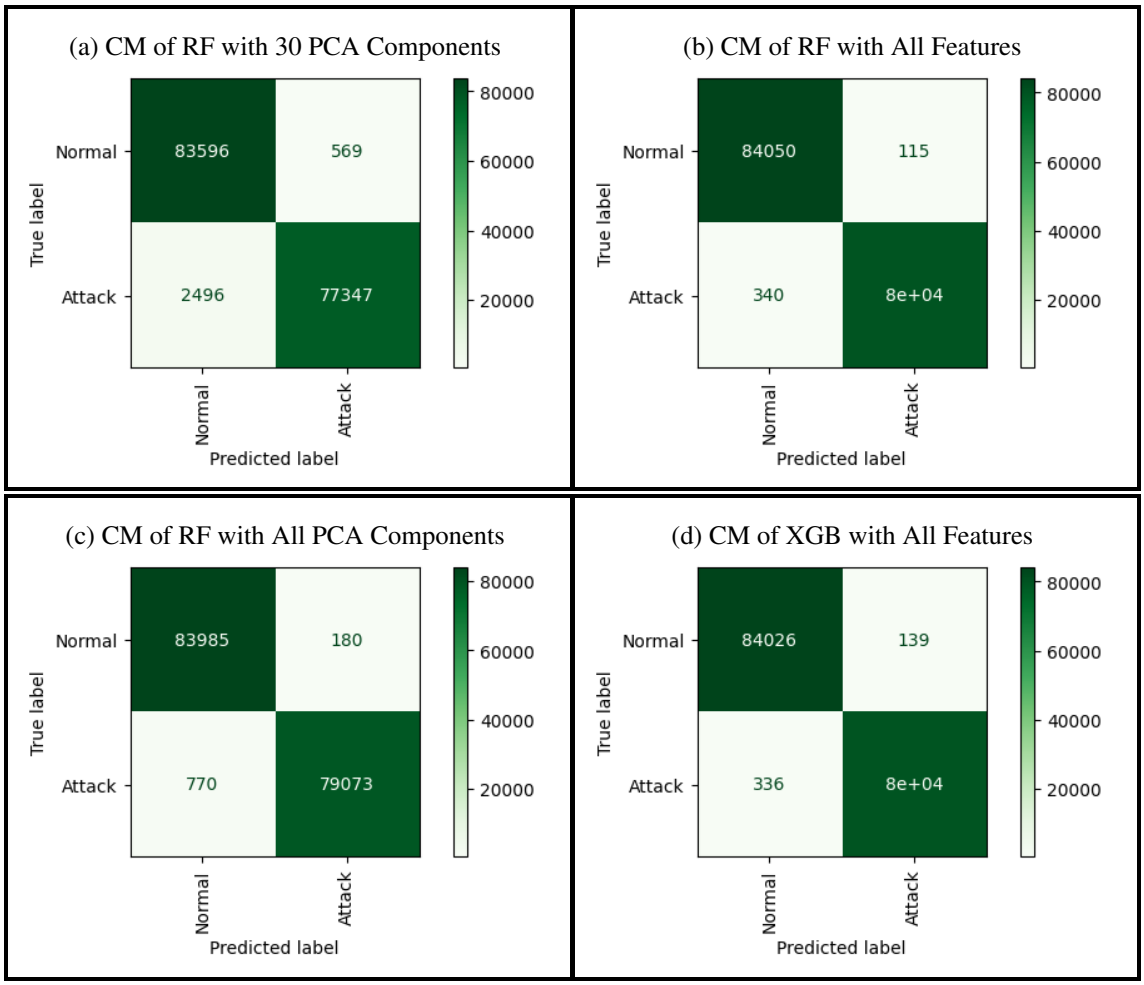


Figure 5.2: Confusion Matrix of some Algorithms

an even lower rate of Normal instances misclassified as Attacks (0.1366%). Similarly, the proportion of Attacks misclassified as Normal traffic is reduced to 0.4258%, and only 0.1444% of the instances predicted as Attacks are actually Normal.

Regarding the XGB model with all Features, the number of misclassified instances of Normal Traffic as Attack is higher than that of the RF model with All Features (0.1652%), but still lower than the other two models. On the other hand, the proportion of Attacks misclassified as Normal (0.4208%) is lower than all other models, with only a few instances being incorrectly classified when compared to RF with All Features. Additionally, the number of instances classified as Normal when their actual Attacks (0.1745%) is smaller than RF in the same scenario, but still lower than the other two models.

Overall, in the 30-Components scenario, the models perform well but not as strongly as in the other scenarios. RF is the top performer here with 98.13% accuracy and an MCC of 96.28%, followed by XGB with 97.09% accuracy. This scenario represents a dimensionality-reduction approach, which can help reduce computational costs but may sacrifice some classification power, as seen by the relatively lower performance across models. With all PCA components included,

model performance improves significantly. RF achieves 99.42% accuracy, and XGB performs almost as well with 99.22% accuracy. This scenario suggests that using all principal components retains more critical information for classification, allowing models to perform better than in the 30-Components scenario. Using All Features is the best scenario overall, as most models reach their highest performance in this setting. RF achieves the top score with 99.72% accuracy and an MCC of 99.45%, followed closely by XGB with 99.71% accuracy and an MCC of 99.42%. The complete feature set provides the models with the maximum amount of information, which seems to help the more sophisticated models (RF, XGB) make highly accurate predictions.

In terms of misclassification behavior, the confusion matrix analysis revealed that the RF model, when trained with all features, minimized the rate of Normal instances misclassified as Attacks (0.1366%) and Attacks misclassified as Normal (0.4258%). XGB, while slightly higher in terms of misclassifying Normal as Attacks (0.1652%), still showed strong performance and had a lower proportion of Attacks misclassified as Normal (0.4208%) compared to RF. These findings underline the capacity of RF and XGB to effectively distinguish between the two classes, exceptionally when trained on the complete feature set.

5.1.2 10-class Classification

Table 5.2 presents results for 10-class classification in three scenarios: 30-Components (PCA with 30 components), All-Components (PCA with all components), and All-Features (no PCA, all original features used), for various machine learning models. The metrics reported include accuracy, precision, recall, F1-score, and MCC, which provide a comprehensive evaluation of model performance. These models were trained using the hyperparameters resulting from the grid search with cv = 7 presented in Appendix .6 Table 5.

Starting with the 30-Components scenario, RF is the top-performing model across all metrics, indicating exceptional performance in accurately classifying the 10 classes. The high MCC value suggests a strong correlation between the predicted and actual classifications, highlighting RF's robustness and reliability in this multi-class scenario. XGB closely follows, performing at a high level across all metrics. Both DT and KNN are slightly behind RF and XGB but remain solid alternatives, with accuracy levels of 96.54% for DT and 96.45% for KNN. In contrast, LR and SVM show moderate performance, with accuracies of 93.13% and 92.38%, respectively. While these models perform reasonably well, they are outclassed by tree-based and ensemble models. However, NB performs poorly, achieving only 65.36% accuracy and an MCC of 57.19%, making it unsuitable for this task. Interestingly, NB shows a relatively high precision of 76.43%, which suggests that when the model does predict a class, it tends to be correct. However, its poor recall (65.37%) and overall performance indicate that it struggles with correctly identifying all relevant instances, leading to imbalanced and unreliable results. Overall, RF and XGB stand out as the most accurate and reliable models for this 10-class classification, while DT and KNN offer good alternatives when simpler or more interpretable models are required.

Regarding All-Components, RF achieves an accuracy of 99.34% and scores high across preci-

Table 5.2: Results from 9-class Classification

Classification	Model	Accuracy	Precision	Recall	F1-Score	MCC
30-Components	SVM	92.38	92.39	92.38	92.08	88.62
	DT	96.54	96.54	96.55	96.54	94.88
	LR	93.13	93.07	93.14	92.97	89.74
	KNN	96.45	96.27	96.45	96.22	94.72
	NB	65.36	76.43	65.37	66.56	57.19
	XGB	97.66	97.66	97.66	97.61	96.53
	RF	**97.97**	**97.98**	**97.97**	**97.85**	**96.99**
All-Components	SVM	97.72	97.76	97.72	97.70	96.63
	DT	98.91	98.91	98.91	98.91	98.38
	LR	97.84	97.86	97.84	97.80	96.79
	KNN	98.12	98.11	98.12	98.10	97.21
	NB	75.87	85.01	75.87	76.95	70.76
	XGB	99.29	99.29	99.29	99.28	98.94
	RF	**99.34**	**99.34**	**99.34**	**99.34**	**99.02**
All-Features	SVM	97.84	97.88	97.84	97.82	96.81
	DT	99.60	99.60	99.60	99.60	99.41
	LR	97.54	97.55	97.53	97.49	96.34
	KNN	98.14	98.13	98.14	98.12	97.24
	NB	55.40	86.50	55.40	59.60	54.25
	XGB	**99.75**	**99.75**	**99.75**	**99.75**	**99.63**
	RF	99.72	99.72	99.72	99.72	99.58

sion, recall, F1-score, and MCC, with the latter at 99.02%. XGB follows closely with an accuracy of 99.29% and an MCC of 98.94%. These models demonstrate exceptional performance, effectively handling the complexity of the 10-class problem and providing reliable and robust classification. DT achieves an accuracy of 98.91% and an MCC of 98.38%, while KNN reaches an accuracy of 98.12% and an MCC of 97.21%. Both models show high precision and recall, making them effective alternatives for the classification task. However, their slightly lower scores compared to RF and XGB. SVM and LR provide good, but not outstanding, results. SVM has an accuracy of 97.72% and an MCC of 96.63%, while LR achieves an accuracy of 97.84% with an MCC of 96.79%. These models perform well overall but are outperformed by the more advanced models. NB significantly underperforms compared to the other models. With an accuracy of just 75.87% and an MCC of 70.76%, NB struggles with classification across the 10 classes. Although NB shows a relatively high precision of 85.01%, its low recall and overall performance indicate substantial limitations.

When analyzing the All Features scenario, XGB stands out as the top performer, with all metrics at 99.75% except for MCC, which is slightly lower at 99.63%. This indicates its superior ability to handle complex patterns and interactions in the data. RF follows closely behind XGB, with metrics just marginally lower but still exceptionally high, emphasizing its strength as an ensemble model. DT delivers near-perfect results, with all metrics scoring at 99.60%, except for MCC (99.41%), which demonstrates its strength in handling multi-class data. KNN shows com-

petitive performance with scores around 98%, indicating its effectiveness in capturing local data structure in the 10-class scenario. SVM achieves a strong performance, with accuracy, precision, recall, and F1-score all around 97.80%, and an MCC of 96.81%, indicating a good balance and reliability in its predictions. LR also performs well, with metrics slightly lower than those of SVM, reflecting its robustness in linear relationships but a slightly lower ability to capture more complex patterns. NB performs poorly in this context with an accuracy of 55.40%, which, despite a high precision of 86.50%, suggests significant imbalances or incorrect assumptions about feature independence.

The F1-score for the classification of each attack in the 10-class using the 30 PCA Components is presented in Table 5.3. These figures provide a visual comparison of how each model performs across different attack classes, further highlighting the strengths and weaknesses observed in the detailed metrics.

Table 5.3: F1-Scores for each attack using 30 PCA Components

	DT	KNN	SVM	LR	NB	RF	XGB
Normal	96.95	96.72	93.65	94.62	65.86	97.98	97.91
Reconnaissance	95.79	96.84	81.67	85.27	55.47	97.32	95.69
Weaponization	98.35	96.43	94.96	92.04	60.59	99.27	99.28
Exploitation	63.9	61.49	30.77	78.11	4.52	63.3	79.42
Lateral Movement	85.76	86.94	76.54	81.51	17.87	92.17	92.6
Comand and Control	69.63	71.59	69.25	55.51	14.53	72.18	78.44
Exfiltration	96.44	93.93	95.21	90.72	50.33	97.65	98.32
Tampering	53.27	21.36	80.21	76.65	5.77	52.08	76.03
Crypto-ransomware	83.98	91.21	73.12	70.59	13.9	97.75	97.21
RDOS	99.94	99.93	99.82	99.87	98.81	99.97	99.97

It is evident that RF and XGB demonstrate superior performance across most classes. RF achieves near-perfect F1-scores in classes such as RDOS (99.97), Weaponization (99.27), and Exfiltration (97.65). XGB performs similarly well, with its strongest results also in RDOS (99.97), Exfiltration (98.32), and Weaponization (99.28). However, both RF and XGB face challenges in specific categories. RF achieves a more moderate score in Tampering (52.08), struggles with Exploitation (63.30), and Command and Control (72.18). XGB shows better but still moderate performance in Tampering (76.03), in Command and Control (78.44), and Exploitation (79.42).

DT and KNN models exhibit more variability in their performance. DT performs well in classes like Weaponization (98.35) and Reconnaissance (95.79) but struggles in others, such as Tampering (53.27), and obtains moderate results in Exploitation (63.90) and Command and Control (69.63). KNN mirrors this pattern, with strong results in Weaponization (96.42), Exfiltration (93.93), but weaker performance in Tampering (21.36), and moderate in Exploitation (61.49) and Command and Control (71.59).

SVM shows mixed results, performing adequately in Exfiltration (95.21%) but poorly in Exploitation (30.77%), and achieving moderate results in Command and Control (69.25%) and Crypto-ransomware (73.12%). LR has its strengths and weaknesses as well. It excels in Weaponiza-

tion (92.03%) and Exfiltration (90.72%). However, it fails in other categories, such as Command and Control (55.51%), and achieves moderate results in Crypto-ransomware (70.59%) and Tampering (76.65%).

NB consistently underperforms across almost all classes, with especially poor results in Exploitation (4.52), Tampering (5.77), and Crypto-ransomware (13.90). Despite this, NB managed to achieve a relatively strong score in RDOS (98.81). This imbalance in the model's performance aligns with its moderately high precision, indicating that while NB may be accurate in its predictions for specific classes, it struggles significantly with others, leading to uneven classification results.

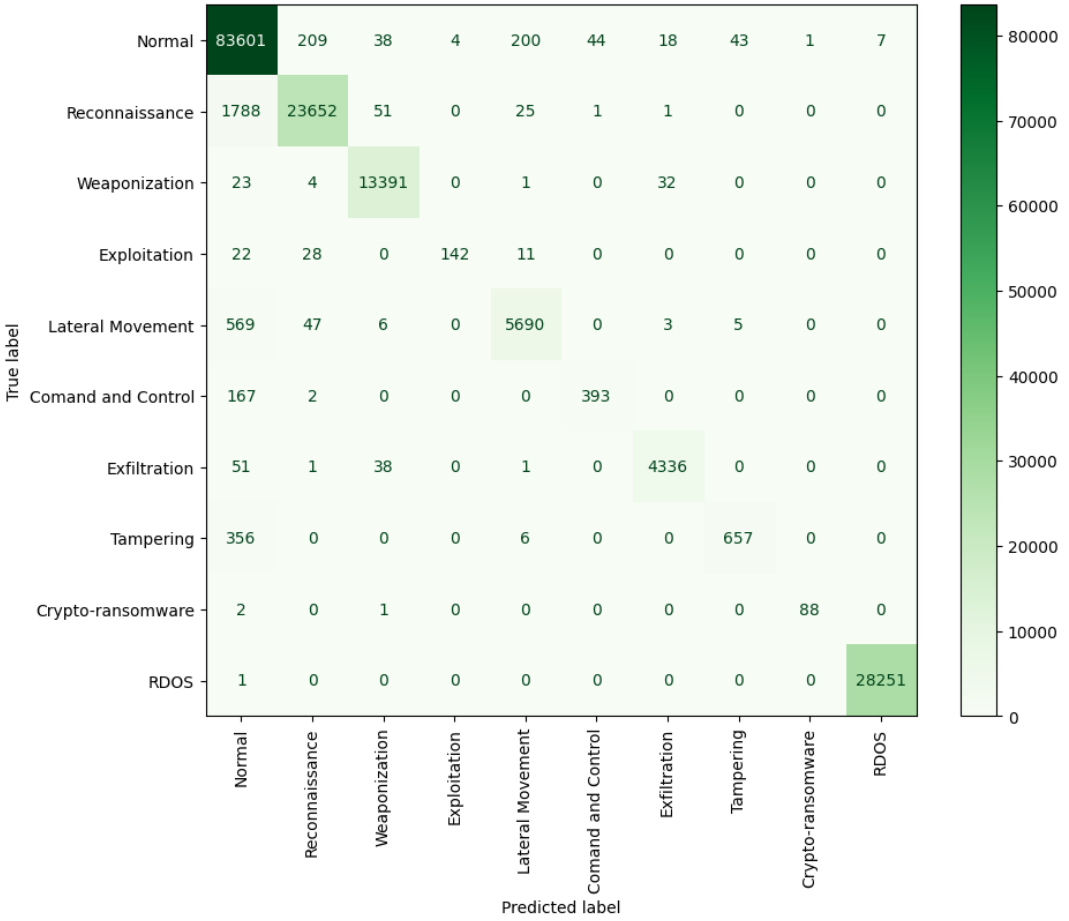


Figure 5.3: Confusion Matrix from XGB using 30 PCA Components

Looking at the confusion matrix for the XGB model in Figure 5.3, it is clear that a significant portion of misclassified attacks are incorrectly identified as Normal traffic. For example, Reconnaissance has 1,788 instances (7.01%) misclassified as Normal, while 78 instances (0.30%) are incorrectly classified as other attack types. Additionally, Normal traffic itself has 564 instances (0.67%) misclassified as attacks, highlighting a tendency for the model to confuse benign traffic with specific attacks.

Focusing on attacks with lower F1-scores, we observe that Exploitation has 22 instances

(10.84%) misclassified as Normal, 28 instances (13.79%) classified as Reconnaissance, and 11 instances (5.42%) classified as Lateral Movement. Command and Control is heavily misclassified, with 167 instances (29.72%) classified as Normal. Similarly, Tampering shows a pattern of misclassification, with 356 instances (34.94%) being incorrectly labeled as Normal.

Overall, XGB and RF demonstrate consistently high performance across most classes, while DT and KNN also show strong results in several categories. SVM and LR perform reasonably but face limitations. NB generally underperforms, particularly in complex or less frequent attack classes. RDOS has the best score among all classes (99.97 with XGB and RF). The classes with the worst scores are Exploitation, Tampering, and Command and Control. Having 79.42 with XGB, 80.21 with SVM, and 78.44 with XGB, respectively. Most classes achieved the best results with XGB and RF, except for those mentioned earlier.

In regard to the attack classification when using All Components, the F1-score for the classification of each attack is presented in Table 5.4.

Table 5.4: F1-Scores for each attack using All PCA Components

	DT	KNN	LR	SVM	NB	RF	XGB
Normal	99.07	98.43	98.2	98.24	62.25	99.44	99.4
Reconnaissance	97.79	97.63	93.82	94.18	56.91	98.71	98.51
Weaponization	99.79	98.23	97.09	98.87	67.79	99.93	99.91
Exploitation	92.7	76.88	78.98	81.84	23.61	95.7	95.43
Lateral Movement	97.77	95.95	96.91	96.17	74.41	98.46	98.61
Comand and Control	80.37	79.18	77.36	81.67	23.59	84.21	87.18
Exfiltration	99.21	96.12	96.25	98.81	56.18	99.67	99.65
Tampering	96.95	89.16	95.94	96.08	74.65	98.08	98.77
Crypto-ransomware	96.7	97.3	92.66	88.1	43.56	98.32	97.78
RDOS	99.99	99.94	99.89	99.92	98.47	99.99	99.99

In the 10-class classification scenario, with all PCA Components, XGB and RF models again have the highest F1-score across almost all attacks. XGB presents a drop in performance regarding attack Command and Control (87.18) and a slight drop when classifying the Exploitation (95.43) class. The same behavior is observed in the RF classifier, which exhibits a drop in performance when evaluating Command and Control (84.21) and Exploitation (95.7). When comparing the results obtained with all PCA components to those with 30 PCA components, the two models (XGB and RF), which already performed best in the reduced scenario, show improved F1-scores across all attack types. This improvement is particularly noticeable in the detection of Tampering, Exploitation, and Command and Control attacks, despite these attacks still having the lowest scores overall.

On the other hand, NB once again performs poorly in attack classification, presenting its worst results in Command and Control (23.59) and Exploitation (23.61), while maintaining good results for RDOS (98.47). NB performance also increased when compared to the previous model trained with 3O PCA Components.

DT exhibits the same pattern as the previous models, where the classes Command and Con-

trol (80.37) and Exploitation (92.7) present the lowest Scores of the model. DT displays good performance, but it is less consistent than XGB and RF, having lower scores than the first two models.

Regarding KNN, LR, and SVM, these models exhibit a more mixed performance. Although these models perform well for many attack types, they consistently struggle with detecting Command and Control and Exploitation attacks, which yield the lowest F1-scores across all three models. In addition to these challenges KNN also has difficulty in detecting Tampering (89.16), LR struggles to correctly classify Crypto-Ransomware (92.66), and Reconnaissance (93.82), SVM faces challenges in detecting Crypto-Ransomware (88.10), Reconnaissance (94.18). These models have lower performance than XGB and RF, both in regard to the attacks mentioned earlier, as well as in other classes.

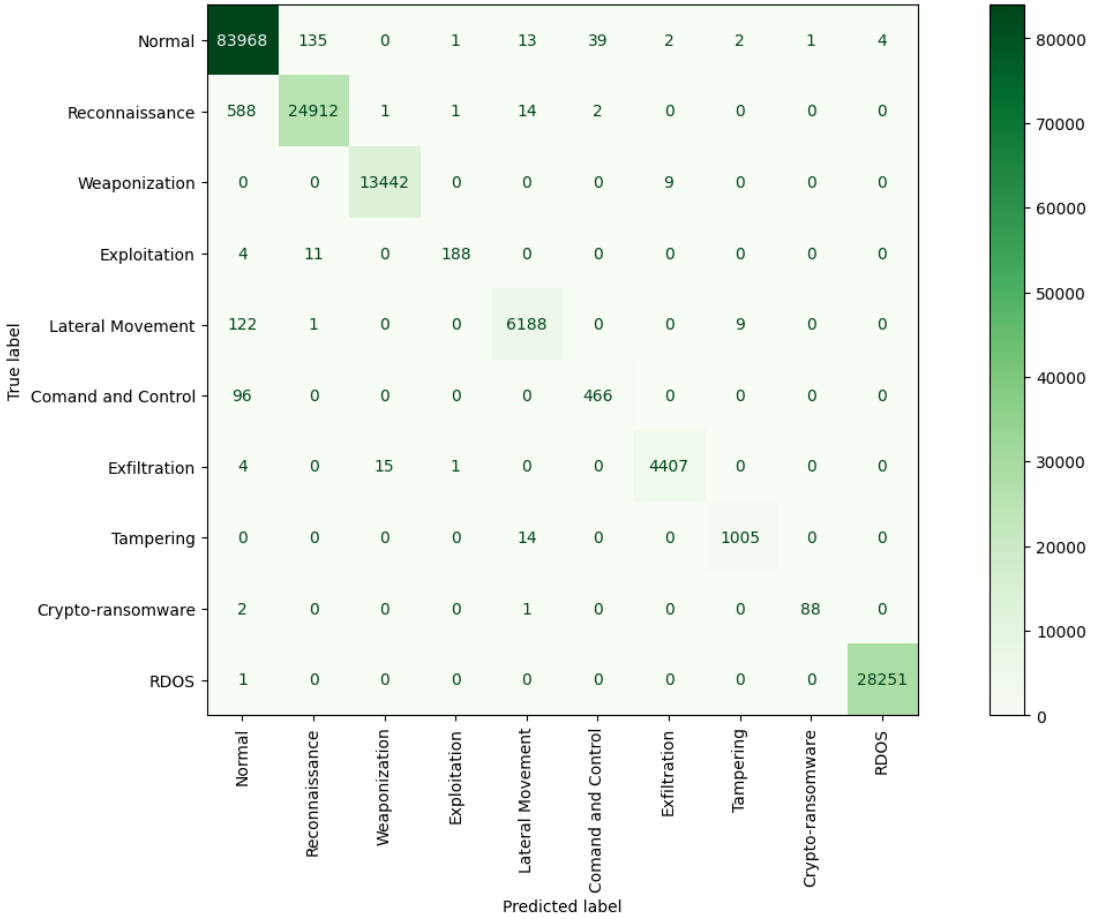


Figure 5.4: Confusion Matrix from XGB using all Components

Looking at the results from the Confusion Matrix regarding the XGB model presented in Fig.5.4, it's possible to observe that, once again, the misclassification in certain attack-related classes predominantly results in them being classified as Normal Traffic. For example, Reconnaissance has 588 instances that are classified as Normal (2.31%), Lateral Movement has 122 instances (1.94%), and Command and Control has 96 cases (17.08%) being mistakenly identified

as Normal. This suggests that the model struggles to distinguish between these types of attacks and legitimate traffic.

When examining Exploitation attacks, it is inferred that the entrances are being classified as Normal and Reconnaissance, the relatively low number of instances for Exploitation in the overall dataset might be contributing to this confusion.

Overall, it's possible to observe that when using all PCA components, the overall results of all models improve across most attack types. The increase in feature information allows the models to achieve higher F1-scores. Although certain attacks, such as Command and Control and Exploitation, still pose challenges for most models, the overall classification accuracy is significantly better compared to the scenario with 30 PCA components.

The F1-score for the classification of each attack in the 10-class with All Features scenario is presented in Table 5.5. Here can be observed that XGB and RF continue to have the highest scores among the models, with XGB as the standout performer, achieving near-perfect F1-scores across all classes, with metrics like Weaponization (99.98), Lateral Movement (99.15), and achieving the lowest results with the class Command and Control (95.86).

Table 5.5: F1-Scores for each attack using All Features

	DT	KNN	SVM	LR	NB	RF	XGB
Normal	99.63	98.39	98.14	98.24	42.97	99.74	99.77
Reconnaissance	99.19	97.61	93.86	94.19	68.81	99.48	99.52
Weaponization	99.94	98.18	97.35	98.73	77.14	99.98	99.98
Exploitation	94.9	75.9	79.77	78.8	12.93	98.27	98.26
Lateral Movement	98.78	95.78	97.13	96.36	27.16	99.01	99.14
Comand and Control	93.98	76.38	76.74	81.44	3.33	94.3	95.86
Exfiltration	99.76	96.56	96.24	98.32	69.95	99.94	99.92
Tampering	99.76	87.47	96.35	96.95	64.96	99.66	99.51
Crypto-ransomware	98.9	97.24	90.91	87.06	33.46	99.45	99.45
RDOS	100	99.95	99.9	99.92	99.28	100*	100*

* Important to note that the values are 99.998(...)

RF obtains similar results, excelling across all classes, with F1-scores consistently above 99, except for Exploitation (98.27) and Command and Control (94.30). DT also demonstrates exceptional performance, obtaining results above 98 in the majority of attack classes, but shows slightly reduced effectiveness in classes like Exploitation (94.90) and Command and Control (93.98).

KNN performs well in several categories, with its performance dropping in classes with greater complexity or fewer examples, such as Exploitation (75.90), Command and Control (76.38), and Tampering (87.47). This suggests that KNN is effective in capturing local patterns but may struggle in more varied or complex classes. SVM exhibits similar behavior, achieving good results, but it falters in classes such as Command and Control (76.74) and Exploitation (79.77), where its performance is less consistent. LR performs similarly to the previous two models, robustly in most classes, but struggles in other classes, such as Exploitation (78.80), Command and Control (81.44), and Crypto-ransomware (87.06), reflecting its limitations.

NB generally underperforms across the board, with particularly low F1-scores classes such as Command and Control (3.33), Exploitation (12.93), Lateral Movement (27.16), and Crypto-Ransomware (33.46). Its overall performance is weak.

Examining the XGB model confusion matrix in Figure 5.5, we can observe that the pattern from previous scenarios persists, with most misclassifications involving attacks being wrongly classified as Normal Traffic. Notably, Reconnaissance (0.64%), Lateral Movement (1.25%), and Command and Control (5.34%) show the highest rates of misclassification as Normal. Additionally, Normal Traffic itself has 0.14% of instances incorrectly classified as attacks.

One notable observation from these results is the minimal misclassification of RDOS and Crypto-Ransomware, with each having only a single instance misclassified, indicating strong model performance for these attack types.

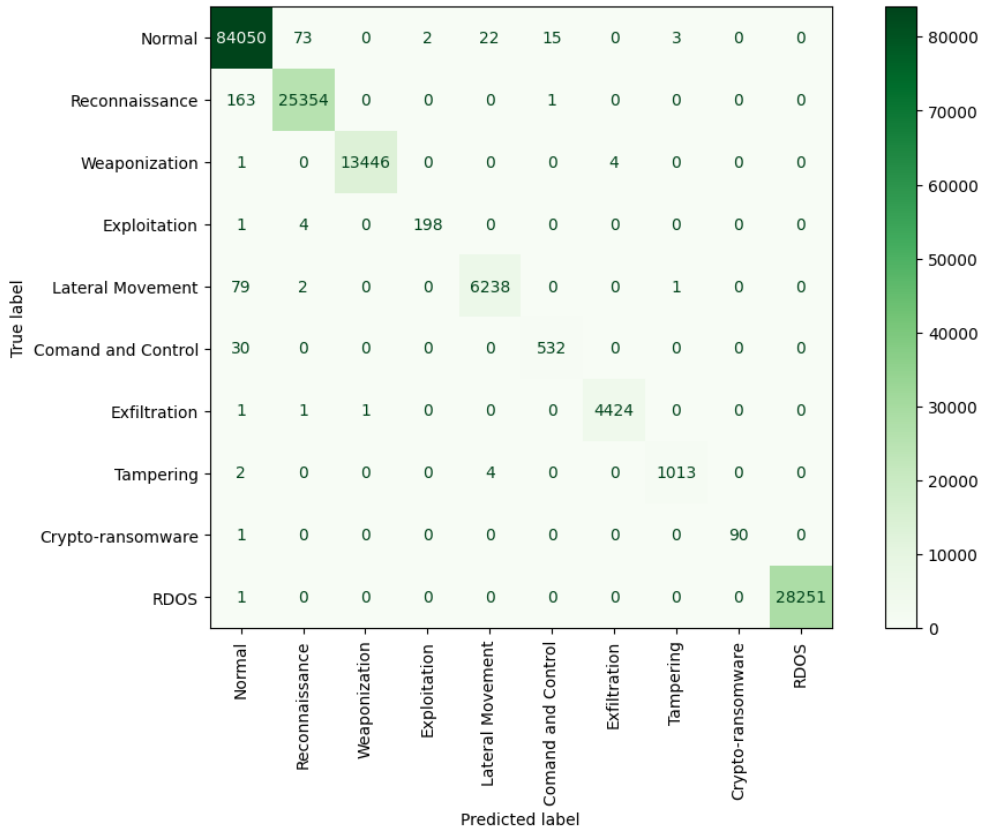


Figure 5.5: Confusion Matrix from XGB using all Features

Overall, the All-Features scenario consistently yields the highest performance across models, when compared to the previous scenarios (30 PCA Components and All PCA Components), particularly for models such as XGB, RF, and DT. This suggests that retaining all features provides the most information for accurate classification. XGB performs best in this scenario, followed closely by RF.

RF is the most consistent top-performing model across all scenarios, showing strong results even with reduced data. It excels particularly when using all features, demonstrating its ability to

handle high-dimensional data and complex patterns effectively. XGB performs exceptionally well, especially in the All-Features scenario, where it achieves the best results. Its boosting mechanism helps it learn more effectively from errors and handle large feature sets with high accuracy. NB is consistently the weakest model in all scenarios, its assumption of feature independence does not hold well for this dataset, leading to poor performance compared to the other models.

In terms of the misclassification, the majority of the traffic being wrongly classified is attacks being classified as Normal (the class with the highest number of instances). Across the multiple scenarios examined, the 30 PCA Components present the highest number of misclassifications, while all features display the smallest number, with all PCA components showing slightly more misclassifications than All Features. The most significant improvement observed, from 30 PCA Components to All PCA Components and All Features, is the correct classification of the attacks: Exploitation, Command and Control, and Tampering.

5.1.3 17-class Classification

The table 5.6 presents the results for 17-class classification using three scenarios previously employed: 30-Components (PCA with 30 components), All-Components (PCA with all components), and All-Features (no PCA, utilizing all features). These results are obtained for various machine learning models. The metrics reported include accuracy, precision, recall, F1-score, and MCC. These models were trained using the hyperparameters resulting from the grid search with cv = 7 presented in Appendix .6 Table 6.

In the 30-component scenario, XGB leads the group, achieving an accuracy of 98.02% and consistently high scores in precision, recall, F1-score, and MCC, with the latter reaching 97.14%. RF is very close behind XGB, with a nearly identical accuracy of 97.96% and slightly lower MCC (97.05%). DT and KNN also perform well, although not at the same level as XGB or RF. DT reaches an accuracy of 96.66%, while KNN is slightly behind with 96.49%, both models showing substantial precision and recall values. Their respective MCC scores of 95.20% for DT and 94. 93% for KNN, although their performance lags slightly behind the ensemble models. SVM and LR fall into a more moderate performance category. SVM achieves an accuracy of 92.84%, with precision and recall scores around 93%, indicating its ability to handle classification reasonably well but not as effectively as the tree-based models. LR performs slightly better with an accuracy of 94.17%, but its MCC of 91.51% shows that it struggles to fully capture the complexity of the 17 classes. NB significantly underperforms compared to the other models, with an accuracy of only 54.00% and a low MCC of 52.36%, indicating its poor ability to classify the 17 classes. Despite a high precision of 82.76%, its recall and F1-scores are notably lower, showing that NB struggles to identify all relevant instances across classes.

In the All-Components scenario, XGB and RF continue to demonstrate exceptional performance, leading the models with the highest scores across all metrics. XGB achieves a stellar accuracy of 99.38%, slightly outperforming Random Forest at 99.34%. These scores are consistent across various metrics, including precision, recall, and F1-score. DT and KNN also perform

Table 5.6: Results from 17-class Classification

Classification	Model	Accuracy	Precision	Recall	F1-Score	MCC
30-Components	SVM	92.84	93.81	92.84	92.94	89.61
	DT	96.66	96.67	96.66	96.66	95.20
	LR	94.17	93.85	94.17	93.76	91.51
	KNN	96.49	96.37	96.49	96.24	94.93
	NB	54.00	82.76	54.00	58.48	52.36
	XGB	**98.02**	**97.98**	**98.02**	**97.91**	**97.14**
	RF	97.96	97.97	97.96	97.77	97.05
All-Components	SVM	97.56	97.45	97.56	97.32	96.45
	DT	98.83	98.84	98.83	98.83	98.31
	LR	97.66	97.60	97.66	97.50	96.62
	KNN	98.11	98.12	98.11	98.07	97.27
	NB	67.35	88.01	67.35	71.39	64.93
	XGB	**99.38**	**99.37**	**99.38**	**99.36**	**99.10**
	RF	99.34	99.33	99.34	99.32	99.05
All-Features	SVM	97.11	97.03	97.11	96.87	95.81
	DT	99.60	99.60	99.59	99.60	99.42
	LR	97.43	97.38	97.43	97.26	96.28
	KNN	98.14	98.13	98.14	98.10	97.32
	NB	49.09	88.57	49.10	48.30	52.45
	XGB	**99.78**	**99.78**	**99.78**	**99.78**	**99.69**
	RF	99.71	99.71	99.71	99.70	99.59

strongly, with DT achieving an accuracy of 98.83% and KNN closely following at 98.11%. Both models maintain high scores in the different metrics. SVM and LR demonstrate solid performance but fall short of the top models. LR achieves an accuracy of 97.66%, and SVM follows closely with 97.56%. While SVM and LR provide reliable classification with commendable scores, their performance is slightly overshadowed by the higher scores of RF and XGB. NB struggles significantly with a much lower accuracy of 67.35%, although it presents an unusually high precision of 88.01%. This discrepancy suggests that NB may be overpredicting certain classes, resulting in high precision in those areas but failing to generalize effectively across the entire dataset. The F1-score of 71.39% and MCC of 64.93 further indicate its limited effectiveness in this multi-class task, making it a less reliable choice compared to other models.

When using All-Features XGB is once again the top performer, with a 99.78% in all metrics, and a MCC of 99.69% showcasing its adaptability and robustness across different classification tasks. RF continues to be the second best performer, with 99.71% in all scores, closely mirroring XGB's results. DT remains consistent with its 10-class performance, maintaining all metrics at around 99.60%, showing its effectiveness in both more straightforward and more complex multi-class tasks. KNN retains its performance, with metrics close to the 10-class results, demonstrating its reliability across different class numbers. SVM again shows solid performance with metrics around 97%, slightly lower than in the 10-class scenario, likely due to the increased complexity of distinguishing more classes. LR continues to perform nicely with metrics around 97.40%,

but slightly lower MCC indicates some struggles with the increased class complexity. NB again underperformed, with an even lower accuracy of 49.09% in the 17-class scenario, indicating its limitations in handling complex and varied data distributions, despite its high precision.

Table 5.7: F1-Score for 17 classes using 30 PCA Components

	DT	KNN	SVM	LR	NB	RF	XGB
Normal	97.08	96.77	95.43	96.09	42.95	98.13	98.2
RDOS	99.95	99.95	99.79	99.9	98.74	99.95	99.96
Scanning vulnerability	98.61	99.3	78.45	86.32	58.02	99.63	99.7
Generic scanning	98.96	99.26	88.85	90.4	76.15	99.46	99.57
Brute Force	99.83	99.64	98.52	97.98	89.5	99.99	99.96
MQTT cloud broker subscription	90.25	89.39	84.38	85.56	53.08	94.67	96
Discovering Resources	87.09	88.61	63.49	68.52	22.87	92.16	83.98
Exfiltration	96.38	94.5	95.98	91.94	74.04	97.59	98.56
Insider Malcious	95.75	88.96	85.35	93.03	72.91	98.08	98.43
Modbus Register Reading	82.61	87.88	0.5	89.81	1.75	89.37	91.37
False Data Injection	56.75	23.73	82.61	75.4	4.63	55.7	76.03
Command and Control	69.97	72.07	71.22	53.06	16.07	69.88	78.15
Dictionary	86.05	85.03	71.73	22.93	6.62	93.6	96.25
TCP Relay	47.43	45.81	23.05	46.71	4.6	50.96	67.51
Fuzzing	36.33	52.11	0	2.88	3.97	48.02	59.22
Reverse shell	61.86	65.36	70.25	77.25	5.63	71.92	84.68
Crypto Ransomware	92.05	95.6	78.85	79.27	19.44	95.4	97.18

Examining the 17-class classification with 30 PCA components, the results of the F1 scores for each class are presented in Table 5.7. It is evident that RF and XGB models consistently deliver high F1-scores across most classes. RF achieves perfect or near-perfect scores for classes like RDOS (99.95), Brute Force (99.99), and Scanning Vulnerability (99.63). Reaches the worst results in classes as TCP Relay (50.96), Fuzzing (48.02), and False Data injection (55.70), and moderate results in Command and Control (69.88) and Reverse Shell (71.92). XGB produces similar results achieving near perfect results in classes as RDOS and Scanning Vulnerability. In the same way as RF, XGB yields the worst results in classes such as TCP Relay (67.51), Fuzzing (59.22), False Data injection (76.03), Command and Control (78.15), and Reverse Shell (84.68), showing better results in these classes.

In contrast, NB generally performs the worst across many classes, with extremely low F1-scores in several categories. For instance, it scores 1.74 for Modbus Register Reading, 4.60 for TCP Relay, 5.63 for Reverse Shell, and 3.97 for Fuzzing, obtaining good results for RDOS (98.78), and moderate results in classes as Generic Scanning (76.15), Brute Force (89.5), Exfiltration (74.04), and Insider Malicious(72.91).

SVM also struggles in specific categories. It completely fails in the Fuzzing class, scoring 0, and scoring very close to 0 in Modbus Register Reading. Performs poorly in TCP Relay with an F1-score of 23.05, Reverse Shell (70.25). LR performs reasonably well in classes like RDOS (99.90) and Brute Force (97.98), but fails in others, such as Fuzzing (2.88) and Dictionary (22.93).

DT and KNN show mixed performance. They perform well in specific categories, such as RDOS and Brute Force, with F1-scores close to or exceeding 99.5%, but falter in others, like False Data Injection, TCP Relay, and Fuzzing, where their scores drop significantly. Achieving moderate results in categories such as Command and Control and Reverse Shell, with values between 60 and 70. DT only achieves 47.43 for TCP Relay, 36.33 for Fuzzing, and 56.75 for False Data Injection. KNN presents 23.73 for False Data Injection, 45.81 for TCP Relay, and 52.11 for Fuzzing.

Examining the confusion matrices for the SVM and XGB models in Fig. 5 and Fig. 6, as presented in Appendix .5, we observe that both models continue the pattern of misclassification predominantly involving confusion between attacks and Normal Traffic, similar to previous models. For example, Discovering Resources is misclassified as Normal in 2414 instances with SVM and 1197 instances with XGB, making it the attack with the highest number of misclassifications as Normal in both models.

In the SVM model, some attacks, such as Scanning Vulnerability and MQTT Cloud Broker Subscription, exhibit misclassifications that are spread across multiple attack categories, rather than being concentrated in a single one.

Focusing on attacks with lower F1-scores, Fuzzing in the SVM model has all of its instances misclassified as Normal traffic. Similarly, Modbus Register Reading instances are primarily classified as Normal or Exfiltration, with only three cases correctly classified. The Dictionary attack also sees 117 instances wrongly classified as Normal.

For the XGB model, Fuzzing follows a similar misclassification trend, mostly being confused with Normal traffic, although 120 instances are correctly classified. TCP Relay is also misclassified mostly as Normal, with the remaining cases spread across various other attack classes. Both False Data Injection and Command and Control are also largely misclassified as Normal. In contrast, Modbus Register Reading shows improvement with XGB, as the model correctly classifies many instances, although some misclassifications still persist in the Normal category.

Overall, all models perform exceptionally well on the RDOS class, with F1-scores close to 1.0, including the otherwise underperforming NB model. In contrast, models struggle significantly with Fuzzing, TCP Relay, Command and Control, and False Data Injection. Regarding Fuzzing, this class achieves the best results with XGB (59.22), while other models fail completely, such as SVM (0), and some have close results to 0, including LR (2.88). Notably, Modbus Register Reading has a score very close to 0 with SVM despite having good results with other models. The TCP Relay class is challenging for many models, with XGB achieving the best results (67.27). Command and Control obtains its best results once again with XGB (78.15). On the other hand, False Data Injection yields its best results with XGB (76.03%).

In Table 5.8, the F1-scores for the scenario with All PCA Components can be observed. RF and XGB are the standout performers, consistently delivering the highest F1-scores across almost all categories. Both models demonstrate a strong ability to handle the complexity and diversity of the various cyberattack patterns. XGB achieves perfect F1-scores in Brute Force, Crypto-ransomware,

Table 5.8: F1-Score for 17-class using All Components

	DT	KNN	LR	SVM	NB	RF	XGB
Normal	98.97	98.24	98.37	98.23	62.9	99.41	99.43
RDOS	99.98	99.92	99.88	99.92	98.67	99.98	99.99
Scanning vulnerability	99.48	99.57	95.64	95.73	51.9	99.72	99.93
Generic scanning	99.61	99.64	95.6	96.08	77.75	99.72	99.83
Brute Force	99.97	99.77	99.8	99.72	85.3	100	100
MQTT cloud broker subscription	99.06	97.21	99.05	99.24	84.86	99.52	99.76
Discovering Resources	90.3	90.32	74.3	71.7	23.48	94.35	93.6
Exfiltration	99.37	96.17	99.14	99.04	89.49	99.82	99.72
Insider Malcious	99.5	94.3	98.95	99.4	89.22	99.8	99.64
Modbus Register Reading	99.75	97.07	99.83	97.38	94.54	99.83	99.75
False Data Injection	96.25	87.63	96.04	96.19	79.8	98.23	99.17
Command and Control	77.97	80.15	81.44	81.6	19.54	85.42	87.83
Dictionary	99.71	97.15	99.81	90.81	82.85	100	100
TCP Relay	71.84	65.99	75.81	77.56	47.87	79.83	80.59
Fuzzing	63.89	57.57	32.15	0	6.91	73.93	84.98
Reverse shell	89.98	74.07	86.5	88.35	20.68	97.74	98.51
Crypto Ransomware	95.51	96.63	92.31	90.59	22.3	96	100

and Dictionary attacks, while RF performs similarly in Brute Force and Dictionary attacks. It's important to note that both models struggle in categories of attack, such as Command and Control, TCP Relay, and Fuzzing, with both models achieving their worst results in these classes: RF in Fuzzing (73.93%) and XGB in TCP Relay (80.59%). Despite this, XGB is the model that offers the best results for the categories of attacks.

The DT model performs well but does not reach the same level as the previous models. Its performance dips significantly, with more complex attack types, like Fuzzing (63.89), TCP Relay (71.84), and Command and Control (77.97). KNN also shows mixed results, excelling in some categories but faltering in others, the model struggles in categories like Fuzzing (57.57), TCP Relay (65.99) Reverse Shell (74.07), and Command and Control (80.15).

SVM and LR yield strong results in many categories but falter in others. LR presents good results in classes such as Brute Force (99.80) and RDOS (99.88), while SVM performs admirably in several categories, such as RDOS (99.92), Exfiltration (99.04), and Modbus Register Reading (97.38), keeping in mind that there are still outperformed by XGB and RF. Both models struggle significantly in correctly classifying Fuzzing with SVM, obtaining 0, and LR 32.15. In addition to this attack, these models face challenges in attacks such as Discovering resources, Command and Control, and TCP Relay.

NB consistently ranks at the bottom in most categories. NB performs best in RDOS (98.67), even in these cases, its scores are lower than those of other models. In categories such as Fuzzing (6.91), Command and Control (19.54), Discovering Resources (23.48), Reverse Shell (20.68), and Crypto Ransomware (22.30), NB performs very poorly.

Examining the results from the Confusion Matrix, of both SVM and XGB, displayed in Ap-

pendix .5 respectively in Fig.1 and Fig.2. In both Confusion Matrices, it's noticeable that the pattern seen in the previous models is maintained; most misclassifications involve attacks being classified as Normal, and Normal Traffic being classified as different attacks.

Focusing on the SVM attack distribution, most of the incorrect classifications of the Scanning Vulnerability classification result in this attack being misclassified as Generic Scanning. The same happens with Generic Scanning, with the majority of the misclassifications being classified as Scanning Vulnerabilities.

Examining the Discovering Resources classification, there exist 2049 instances (44.27%) of the attack being classified as Normal. Regarding the Fuzzing classification, which presented an F1-score of 0, it can be seen that these attacks are once again being classified as Normal. An improvement observed when using all PCA components instead of the reduced number of components is the correct classification of the attacks, Modbus Register Reading having a single instance that is misclassified.

Observing the XGB Confusion Matrix, Discovering Resources presents the most significant number of instances (9,48%) of attacks being classified as Normal. Looking into the attacks that offered the lowest F1-score, Command and Control, Fuzzing, and TCP misclassifications are concentrated in the Normal Traffic category, with TCP Relay having 2 instances classified as Scanning Vulnerability and 1 instance classified as Reverse Shell.

In this scenario, an overall improvement in the results from models trained with RF and XGB was observed, resulting in the best models. Both models demonstrated strong capabilities in handling the complexity of the dataset, delivering high F1-scores across most attack categories, and showing noticeable improvements in more challenging categories, such as False Data Injection and Reverse Shell, compared to models trained with fewer PCA components.

Despite the overall improvements, some challenges persisted, particularly in detecting more complex attacks, such as Fuzzing, TCP Relay, and Command and Control. While XGB managed to outperform other models in these challenging classes, achieving the highest F1-scores, there were still notable misclassifications, especially between attack types and Normal Traffic.

Nonetheless, the use of all PCA components resulted in a more robust performance across the board, with RF and XGB consistently leading in both accuracy and reliability, confirming their status as the top models in this classification task.

Considering the results from the 17-class classification with all features observed in Table 5.9, RF and XGB emerge as the top performers, achieving high scores across all classes.

RF reaches a perfect F1-score of 100 in critical classes such as Brute Force, Dictionary, and Crypto Ransomware, demonstrating its ability to classify these attacks precisely. XGB similarly excels, achieving perfect scores in Dictionary and Crypto Ransomware, and scoring near-perfect results in several other classes, including Exfiltration, Modbus Register Reading, and Insider Malicious.

This shows that RF and XGB are exceptionally effective at identifying various attacks, making them reliable models for multi-class classification tasks. Despite this, and both models presenting

Table 5.9: F1-Score of 17-class with All Features

	DT	KNN	LR	SVM	NB	RF	XGB
Normal	99.67	98.39	98.41	98.27	20.85	99.73	99.8
RDOS	100	99.95	99.89	99.93	99.56	100	100
Scanning vulnerability	99.82	99.6	93.93	92.89	52.05	99.98	99.99
Generic scanning	99.87	99.64	94.14	93.62	78.55	99.88	99.97
Brute Force	99.99	99.75	99.64	99.61	90.52	100	99.99
MQTT cloud broker subscription	99.89	97.76	99.36	99.5	28.28	99.94	99.93
Discovering Resources	97.26	90.63	74.8	72.26	20.84	98.02	98.51
Exfiltration	99.89	96.56	97.47	97.97	86.34	99.94	99.92
Insider Malcious	99.87	94.2	96.87	97.8	76.57	99.93	99.9
Modbus Register Reading	99.96	97.25	99.92	97.74	99.71	99.75	99.79
False Data Injection	99.85	87.59	96.85	97.31	71.63	99.66	99.71
Command and Control	94	76.38	79.38	81.57	19.92	94.11	95.71
Dictionary	100	97.34	99.71	87.71	85.31	100	100
TCP Relay	83.54	63.57	77.29	79.21	47.43	84.95	88.89
Fuzzing	86.17	55.19	28.57	0	3.53	87.05	90.8
Reverse shell	93.8	75.9	84.83	76.7	65.28	97.76	98.76
Crypto Ransomware	98.34	97.24	90.71	90.8	52.3	100	100

the best results among all the scenarios tested, they still struggle to correctly classify Fuzzing, TCP Relay, and Command and Control.

DT also performs well, though it falls slightly behind RF and XGB in overall consistency. DT achieves near-perfect F1-scores in classes like Brute Force (99.99) and RDOS (99.99). However, DT shows a decline in performance for more complex attacks, such as TCP Relay (83.54), indicating its limitations in identifying attack types that exhibit more subtle or intricate behaviors. KNN displays a similar pattern to DT, performing strongly in some classes while faltering in others. KNN achieves high F1-scores in classes such as Brute Force (99.75) and RDOS (99.95), but its performance drops significantly in more challenging scenarios, including Fuzzing (55.19), Command and Control (76.38), and TCP Relay (63.57).

LR shows strong F1-scores in Modbus Register Reading (99.92%) and Brute Force (99.64%), indicating that it effectively handles linear patterns in the data. However, LR struggles in classes like Discovering Resources (74.80) and Fuzzing (28.57), where more complex relationships between features are present. SVM excels in detecting RDOS (99.93), Brute Force (99.61), and Exfiltration (97.97). However, it falls short in attacks such as TCP Relay (79.21) and fails completely once more in Fuzzing (0).

NB stands out as the weakest performer again, consistently struggling across most classes. NB's F1-scores are notably low in critical classes like Normal (20.85), Command and Control (19.92), and Fuzzing (3.53), reflecting its inability to accurately classify these attacks. Even in classes where NB performs relatively better, such as RDOS (99.56), it is still outperformed by most other models.

Observing the Confusion matrix resulting from the SVM and the XGB models is exposed in

Appendix.5, respectively, in Fig.3 and Fig.4. It is noted that the number of classes being wrongly classified has decreased, maintaining the trend of mislabeling attacks as Normal and Normal as attacks. Discovering Vulnerability continues to be the class with the highest number of instances being misclassified as Normal.

Examining the SVM classification distribution, the Fuzzing attack classification consistently misclassifies all its instances as Normal Traffic. TCP Relay and Command and Control have their misidentified instances concentrated as Normal. Reverse Shell instances are classified as Scanning Vulnerability, Generic Scanning, and TCP Relay.

Regarding the XGB confusion matrix, this model presents the best results, with the lowest amount of attacks being misidentified. Observing the classes with the lowest F1-scores, TCP Relay, Fuzzing, and Command and Control once again have their misclassification aggregated in the Normal Traffic category.

The all-features scenario is the most effective overall, yielding the highest accuracy across all models. Utilizing all features enables models to harness the full complexity of the dataset, resulting in improved multi-class classification performance.

XGB is the top-performing model in all scenarios, particularly excelling in the All-Features scenario, where it achieves near-perfect classification results. XGB's boosting technique makes it highly effective for multi-class tasks, as it handles complex relationships between features and classes.

RF consistently performs just behind XGB in all scenarios, showing strong generalization and robustness. While RF and XGB are closely matched in performance, XGB has a slight edge, particularly when all features are utilized. DT and KNN perform well but are outclassed by XGB and RF. DT is powerful in the All-Features scenario, while KNN shows steady performance across scenarios but remains less effective in complex multi-class problems. NB is consistently the weakest model.

Notably, across all scenarios, the SVM model is incapable of correctly classifying the Fuzzing attack, consistently classifying all its instances as Normal.

5.1.4 Discussion

The consistent high performance of RF and XGB across most attack types can be attributed to their ensemble nature, which allows them to capture complex patterns in the data. These models are highly capable of handling diverse attack behaviors. The robustness of these models in handling high-dimensional data and capturing feature interactions significantly contributes to their overall effectiveness in the classification task.

While DT is a solid performer for many classes, its lower scores in certain areas suggest that it may struggle with generalizing to more complex attacks compared to ensemble models. These inconsistencies can result from DT's tendency to overfit and its limited ability to capture nuanced relationships between features. Despite this, DT still performs admirably in simpler attack scenarios and can be helpful when the decision boundaries are more precise.

The variation of the performance of KNN suggests that KNN is highly dependent on the nature of the attack, it performs well when the attack signatures are distinct and easily separable, but struggles when the differences between attack types are more nuanced. KNN's reliance on proximity in feature space makes it less effective for complex patterns, where more advanced models, such as RF and XGB, can capture underlying relationships more effectively.

This aligns with the limitations of LR, which assumes a linear decision boundary and may not be able to capture the full complexity of certain attack behaviors. Consequently, while LR is a reliable model for simpler attack types, it may not be the best choice for multi-class problems involving complex, non-linear relationships.

The poor performance of NB can be attributed to its strong assumption of feature independence, which is rarely valid in real-world datasets, particularly in complex attack scenarios where features are often correlated. Despite its poor performance in most attack types, NB achieves a reasonable performance in specific classes, such as RDOS, indicating that it can be effective in particular scenarios where its simplistic assumptions hold. However, NB's inability to model complex relationships between features makes it unsuitable for this multi-class classification task.

It is essential to compare the results obtained by the best models tested in the original X-IIOTID article [2] (DT, NB, KNN, SVM, LR, DNN, GRU), all trained with 30 PCA Components, and the best model obtained in this thesis (i.e., XGB with all features). The comparison of these models in terms of recall is presented in Table 5.10 for the 9-class multi-class classification and in Table 5.11 for the 16-class multi-class classification.

Attack	Best model in [2]	XGB -this thesis
C&C	**100 (NB)**	94.66
Crypto_ransom	**99.92 (SVM)**	98.90
Exfiltration	**99.93 (GRU)**	**99.93**
Exploitation	**98.52 (DT)**	97.54
LM	**99.83 (SVM)**	98.70
RDoS	**99.99 (DT)**	**99.99**
Reconnaissance	99.22 (DT)	**99.35**
Tampering	**99.47 (DT)**	99.41
Weaponization	**99.97 (DT)**	99.96

Table 5.10: Recall comparison DT algorithm and XGB model across different attack categories

In the context of the 10-class multi-classification task, XGB demonstrates competitive performance across complex attack types such as Reconnaissance (99.35 vs. 99.22), showcasing its ability to handle diverse attack patterns effectively. While the best model from [2] achieves marginally higher scores in some cases, XGB's performance remains robust and consistent, particularly in scenarios where specialized models like NB, DT, SVM, and GRU might struggle to generalize.

However, the best model from [2] outperforms XGB in specific classes, such as Exploitation (98.52 vs. 97.54), Lateral Movement (99.83 vs. 98.70), and Weaponization (99.97 vs. 99.96). Both models achieve perfect or near-perfect scores in RDoS (99.99) and Exfiltration (99.93 vs.

99.93), highlighting their strengths in these areas. This comparison highlights that while XGB offers a more unified and consistent approach across all attack types, the best model from the article excels in specific categories, utilizing specialized algorithms such as NB, DT, SVM, and GRU.

While the best models from [2] achieve slightly higher accuracy for most attack types, XGB remains a highly desirable choice due to its practicality and efficiency. The approach presented in [2] relies on multiple specialized models (NB, SVM, DT, GRU), each tailored to specific attack types, which increases complexity in deployment and maintenance. In contrast, XGB provides a single, unified model that performs consistently well across all attack types, simplifying implementation and reducing operational overhead.

Moreover, the performance gap between the best models and XGB is minimal, with differences often less than 1%. By using a single model, XGB ensures consistency, reduces training time, and allows for easier updates to incorporate new data or attack types. These benefits make XGB a compelling choice despite not consistently achieving the absolute highest score.

Attack	**Best model in [2]**	**XGB - this work**
C&C	92.11 (NB)	**95.91**
Crypto_ransom	98.06 (NB)	**100**
Exfiltration	99.92 (GRU)	**99.96**
Resverse Shell	**99.15 (NB)**	93.60
MQTT	99.80 (DT)	**99.98**
Modbus Register Reading	99.80 (GRU)	**99.92**
TCP Relay	**99.44 (DT)**	83.49
RDoS	99.96 (GRU)	**99.98**
Generic Scanning	99.92 (SVM)	**99.98**
Scanning Vulnerabilities	99.82 (DNN, GRU)	**99.99**
Fuzzing	**99.99 (DT, KNN)**	85.17
Dicovering Resources	**99.84 (DT)**	98.19
Tampering	**99.99 (SVM)**	99.90
Brute Force	**99.99 (KNN, SVM, GRU)**	**99.99**
Dictionary	**99.84 (DT)**	99.61
Malicious Insider	**99.96 (DNN)**	99.89

Table 5.11: Recall comparison DT algorithm and XGB model across different attack categories

Regarding the 17 multi-class classification, XGB outperforms the best models from [2] in several attack categories. For instance, XGB shows a significant improvement in categories such as Command and Control (95.91 vs. 92.11), Modbus Register Reading (99.80 vs. 99.92), and Scanning Vulnerabilities (99.99 vs. 99.82). These results highlight XGB's ability to handle complex attack types effectively, often outperforming the best models from the article.

However, the best results from [2] maintain an edge in specific categories, such as Reverse Shell (99.15 vs. 93.60), TCP Relay (99.44 vs. 83.49), and Fuzzing (99.99 vs. 85.17). These results indicate that specialized models like NB, DT, and GRU can excel in specific scenarios where XGB struggles. Both models perform equally well in Brute Force (99.99), demonstrating

their strengths in this category. This comparison underscores that while XGB provides a more unified and consistent approach across most attack types, the best models from the article retain their advantages in specific, highly specialized scenarios.

The presented results indicate that while XGB demonstrates superior performance in the majority of attack types, including Command and Control, Crypto-ransomware, Exfiltration, MQTT, and RDoS, the best results from [2] achieved slightly higher scores in some specific attack categories. These results highlight XGB's ability to handle a wide variety of attack patterns effectively, often outperforming specialized models like NB, DT, and GRU. Despite the slight variations in specific cases, its consistent performance across diverse categories makes it a reliable and versatile solution for real-world applications.

When comparing the results obtained in [2] with those presented in this work, which analyzes the scenario using 30 PCA components, the performance in this work falls behind. Several factors likely contribute to these differences in the results, despite employing a similar pre-processing approach as described in the article.

A key distinction is the exclusion of the MiTM and False Data Injection classes in this work, which could have impacted the overall classification performance. Additionally, while both studies applied normalization, there is a difference in the range: this work normalizes the data within the interval [-1, 1], whereas the article normalizes to [0, 1]. Another significant difference lies in the specifics of the normalization method and the handling of missing data, which were not explicitly detailed by the authors of the original article. These variations in pre-processing steps likely account for the discrepancies in the results observed between the two studies.

5.2 Attack Correlation

This section analyses the results of the attack correlation. In subsection 5.2.1, the results from the clustering using all features are presented subsection 5.2.2 displays the results from clustering using the selected features, and subsection 5.2.3 contains the results from the clustering after the selection of features using NSGA-II.

5.2.1 All Features

The following subsection presents the clustering results obtained by using all features of each sub-dataset, where each sub-dataset comprises the data from each of the attacks present in the original SCVI-APT-2021 dataset.

Initial Compromise

The Results of the clustering of the Initial Compromise attack are presented in Fig.5.6 and Fig.5.7. It can be seen how the clusters are distributed across the several APTs.

APT1 is primarily represented in cluster 1 (10 instances) and cluster 0 (8 instances), with only a single instance in cluster 3 and no presence in cluster 2. This distribution suggests that APT1 exhibits two distinct types of behavior, represented by clusters 0 and 1.

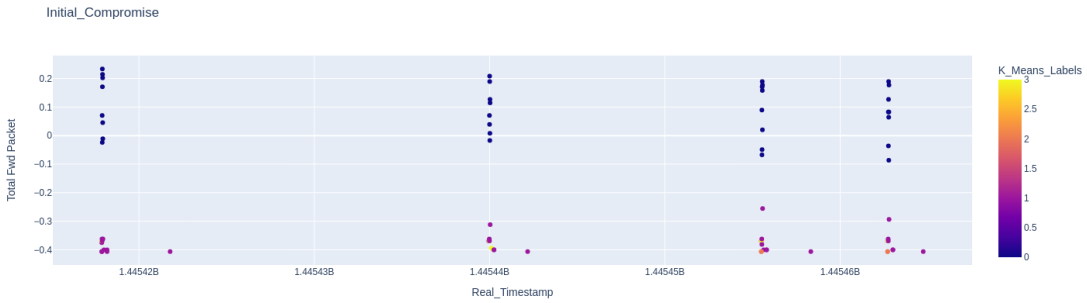


Figure 5.6: Initial Compromise clustering using all features

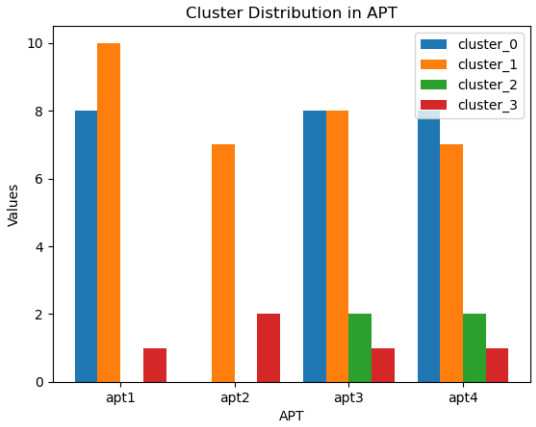


Figure 5.7: Initial Compromise clustering distribution using all features

APT2's attacks are concentrated in cluster 1 (7 instances), with two instances in cluster 3 and no presence in cluster 0 or cluster 2. This suggests that APT2's behavior aligns more closely with the characteristics captured by cluster 1, while the small presence in cluster 3 indicates a distinct but less common behavior. The lack of representation in clusters 0 and 2 suggests that APT2's attack patterns do not match those captured by these clusters.

APT3 is distributed relatively evenly across clusters 0 and 1, with 8 instances each. It also has a small presence in cluster 2 (2 instances) and cluster 3 (1 instance). This indicates that APT3 exhibits more varied behavior than APT2, as it is present in three clusters.

APT4 exhibits a distribution similar to APT3, with most of its attacks concentrated in clusters 0 and 1 (8 and 7 instances, respectively). It also has two instances in cluster 2 and one in cluster 3, indicating that APT4 employs a range of attack patterns.

Reconnaissance

The clustering of the Reconnaissance attack stage is presented in Fig.5.8 and Fig.5.9. Here, it can be seen how the clusters are distributed across the several APTs.

Cluster 0 is the dominant cluster across all four APTs, containing the majority of instances for APT1, APT2, APT3, and APT4. APT1, APT2, and APT3 are almost entirely concentrated in

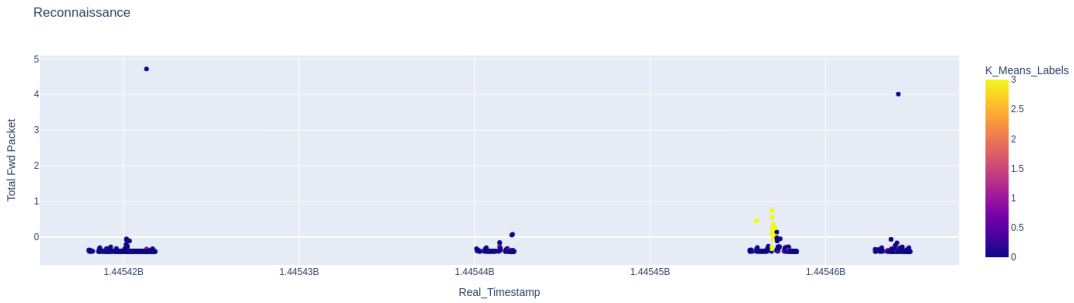


Figure 5.8: Reconnaissance clustering using all features

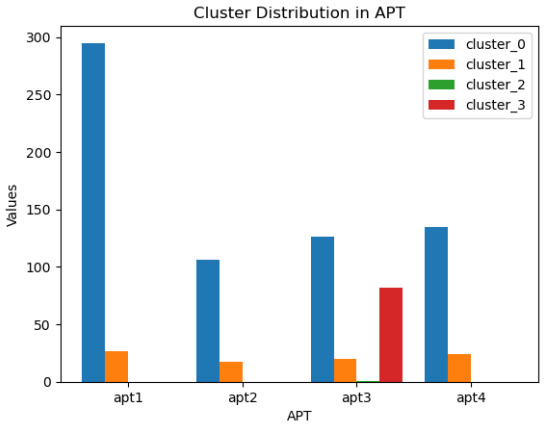


Figure 5.9: Distribution of points in reconnaissance clustering using all features

cluster 0, with a small number of instances in cluster 1. No APT1 attacks are found in clusters 2 or 3, indicating that APT's instances are very similar in behavior, forming a highly cohesive group. The small number of attacks in cluster 1 suggests that there may be minor variations in APT's instances, but they are mostly captured by cluster 0.

APT3 shows more diversity in its distribution across clusters. While a large number of APT3 attacks are found in cluster 0, there is also significant representation in cluster 3, with a small portion in cluster 1 and a single instance in cluster 2. This suggests that APT3 attacks exhibit more varied behaviors compared to APT1 and APT2, which are mainly concentrated in one cluster.

Pivoting

In regard to the clustering of the Pivoting attack stage, the results are presented in Fig.5.10 and Fig.5.11. Here, it can be seen how the clusters are distributed in the several APTs.

APT1 attacks are predominantly found in cluster 3, with a substantial number of attacks also in cluster 1. Clusters 0 and 2 contain relatively few APT1 attacks. This suggests that cluster 3 may represent a broad category or behavior pattern to which APT1 attacks belong, but the overlap with cluster 1 indicates that some variation exists in how APT1 attacks are categorized.

APT2 attacks are similarly concentrated in cluster 3, but with a notable number of attacks

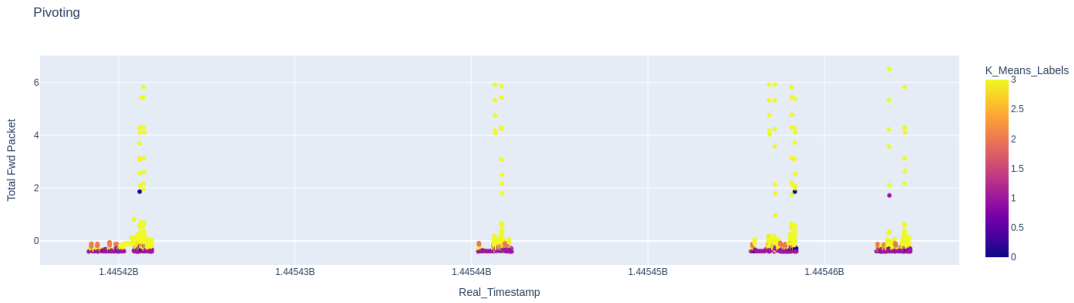


Figure 5.10: Pivoting clustering using all features

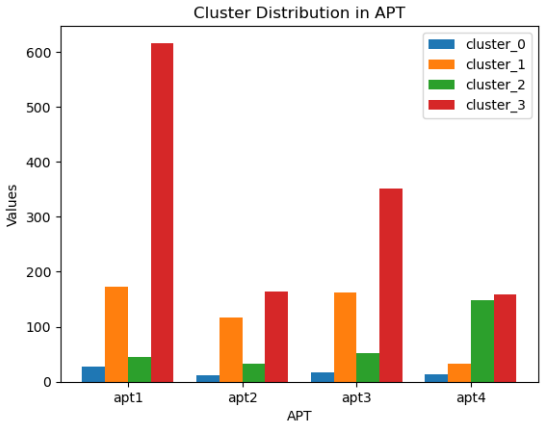


Figure 5.11: Pivoting clustering distribution using all features

also in cluster 1. Compared to APT1, APT2 has a smaller overall distribution but shows a similar pattern of concentration in cluster 3. This suggests that APT2 attacks share some characteristics with APT1, but also exhibit distinct behaviors, as indicated by their presence in cluster 1 and, to a lesser extent, clusters 0 and 2.

APT3 attacks also show a strong presence in cluster 3, though with a significant proportion in cluster 1. Cluster 2 contains a noticeable number of APT3 attacks as well. The broader distribution of APT3 attacks across multiple clusters, especially cluster 2, suggests that APT3 may exhibit more varied behaviors than APT1 and APT2, potentially indicating a wider range of techniques or tactics within this group.

Unlike the previous APTs, APT4 shows a more balanced distribution across clusters 2 and 3. This suggests that APT4 attacks may be more diverse in nature, possibly spanning different attack behaviors or characteristics that are captured in these two clusters. The relatively lower concentration in cluster 3 compared to the other APTs further highlights that APT4 may exhibit a distinct attack pattern.

Lateral Movement

The clustering results of the Lateral Movement attack stage are presented in Fig. 5.12 and Fig.5.13. Here, it can be observed how the clusters are distributed across the several APTs, making it evident that cluster 3 is the dominant cluster in the overall sub-datasets.

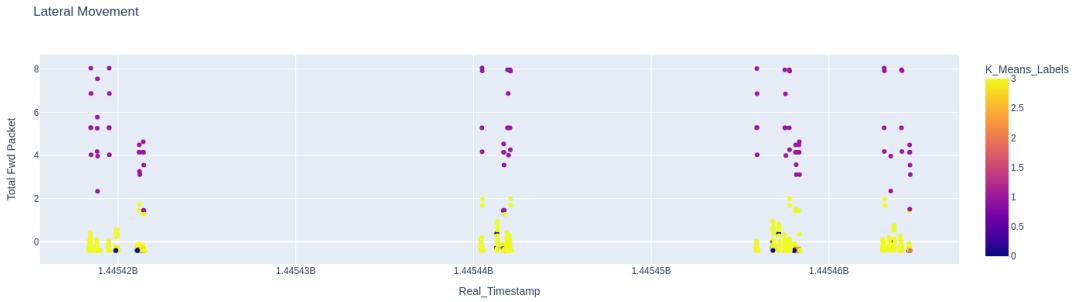


Figure 5.12: Lateral Movement clustering using all features

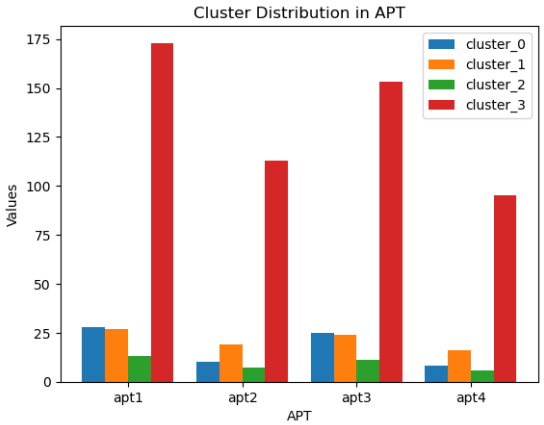


Figure 5.13: Lateral Movement clustering distribution using all features

APT1 is primarily concentrated in cluster 3, which contains the majority of instances, 173 instances. A considerably smaller number of APT1 attacks are also found in clusters 0 and 1, each with 28 and 27 instances, respectively, and only 13 in cluster 2. APT2 also shows a strong presence in cluster 3 with 113 instances. The rest of the points are distributed by the rest of the clusters with cluster 1 with 19 instances, cluster o with 10 and finally cluster 2 with 7 instances. APT3 has a similar distribution to the rest of the APTs, with the majority of attacks clustered in cluster 3 and a significantly smaller number spread across the rest of the clusters, with cluster 0 the second biggest cluster with 25 instances, followed by cluster 1 with 24 points, and finally cluster 2 with only 11 points. APT4 has a similar distribution pattern, with most attacks concentrated in Cluster 3, while smaller portions are spread across the other clusters.

Data Exfiltration

The results of the Data Exfiltration attack stage clustering are presented in Fig.5.14 and Fig.5.15. Here, it can be observed how the clusters are distributed across the several APTs.

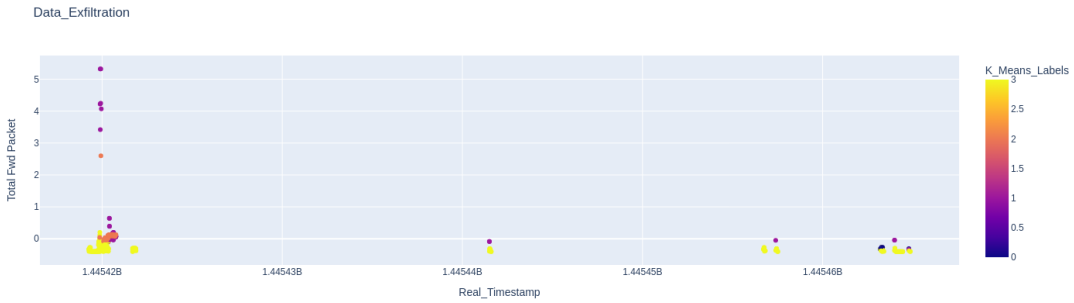


Figure 5.14: Data Exfiltration clustering using all features

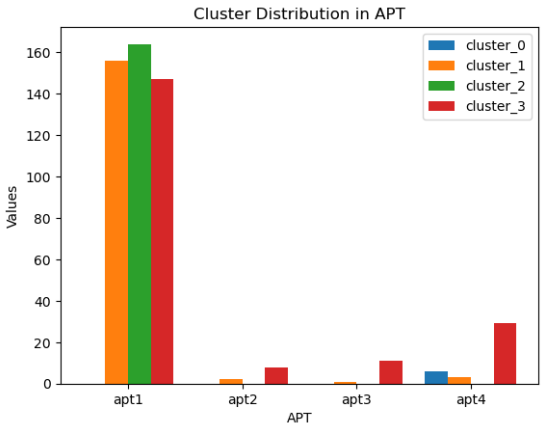


Figure 5.15: Data Exfiltration clustering distribution using all features

APT1 is spread almost evenly across clusters 1, 2, and 3, with no instances in cluster 0. This suggests that APT1 has multiple attack patterns that are sufficiently distinct to be captured by different clusters. The even distribution across three clusters implies that APT1 may employ varied tactics, with each cluster representing a different aspect of APT1's attack strategy.

APT2 is sparsely represented, with a total of 10 instances split between clusters 1 and 3. This suggests that APT2 attacks are rarer or less varied compared to APT1, with only a few instances falling into two distinct clusters. The fact that APT2 does not appear in cluster 0 or cluster 2 suggests that its attack patterns are limited to the characteristics captured by clusters 1 and 3.

APT3 follows a similar pattern to APT2, with most of its attacks concentrated in cluster 3, and a single instance in cluster 1. This suggests that APT3 exhibits some unique characteristics, which are primarily captured by cluster 3.

APT4 is represented in clusters 0, 1, and 3, with the majority of its attacks concentrated in cluster 3. The presence of 6 instances in cluster 0 suggests that APT4 shares some characteristics with other APTs captured in this cluster, though the majority of its attacks are distinct enough to

be placed in cluster 3. The small number of instances in cluster 1 suggests that APT4 may have some overlap with APT1 or share some attack features with the instances clustered there.

5.2.2 Selected Features

In this section, the features selected based on a domain knowledge and used for the clustering of the sub-datatsets were: ['Src IP', 'Src Port', 'Dst IP', 'Dst Port', 'Protocol', 'Timestamp', 'Flow Duration', 'Total Fwd Packet', 'Total Bwd packets', 'Total Length of Fwd Packet', 'Total Length of Bwd Packet'].

Initial Compromise

The results from the clustering of the sub-dataset initial compromise are presented in Fig.5.16 and the result from the distribution of the points of the APTs in the clusters in Fig.5.17. This scenario presents a more balanced distribution of the APTs across the clusters, with all APTs instances having a similar distribution, with the majority of the data points concentrated in clusters 0 and 2, with 8 instances in each, and a single instance in cluster 3.

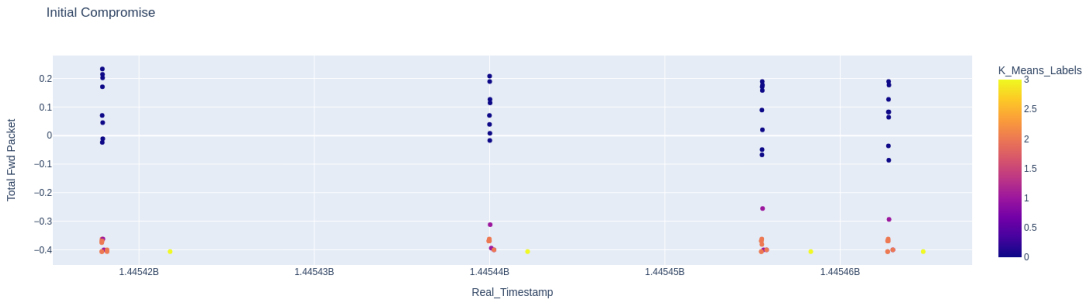


Figure 5.16: Initial Compromise clustering using selected features

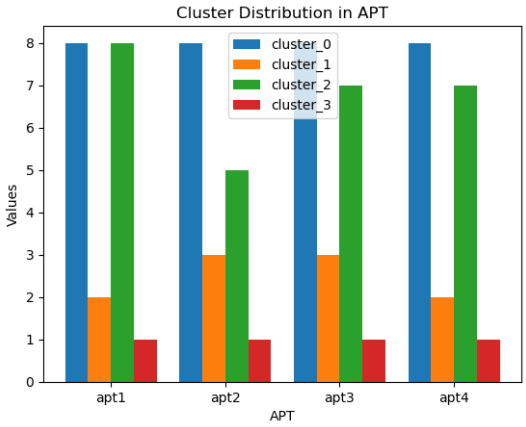


Figure 5.17: Initial Compromise clustering distribution using selected features

APT1 presents the same data points in clusters 0 and 2, with only a single instance in cluster 3 and 2 instances in cluster 1. This distribution suggests that APT1 exhibits two distinct types of

behavior, represented by clusters 0 and 2. APT2's points are concentrated in cluster 0 and cluster 2, with 3 instances in cluster 1. APT3 is distributed relatively evenly across clusters 0 and 2, with 7 instances in cluster 0 and 8 instances in cluster 2, respectively. It also has a small presence in cluster 1 (3 instances) and cluster 3 (1 instance). APT4 exhibits a distribution similar to that of the other APTs, with most of its attacks concentrated in clusters 0 and 2, a single instance in cluster 3, and 2 instances in cluster 1.

Reconnaissance

The sub-dataset of the attack reconnaissance obtained the distribution presented below in Fig.5.18 and in Fig.5.19. This scenario presents a more varied distribution of data points across the clusters, indicating that the attack patterns may be less cohesive or defined. Cluster 1 is the most populated cluster, particularly for APT1 and APT3, which both have a high number of instances here. Cluster 2 is the least populated cluster, with 0 instances of APT1 and APT4 and very few from APT2 and APT3.

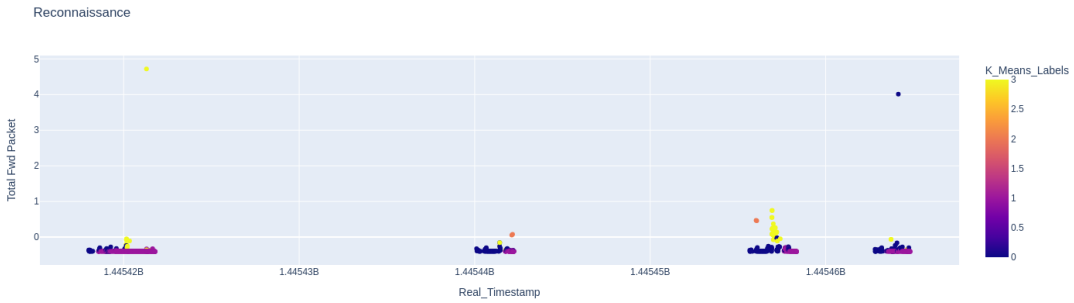


Figure 5.18: Reconnaissance clustering using selected features

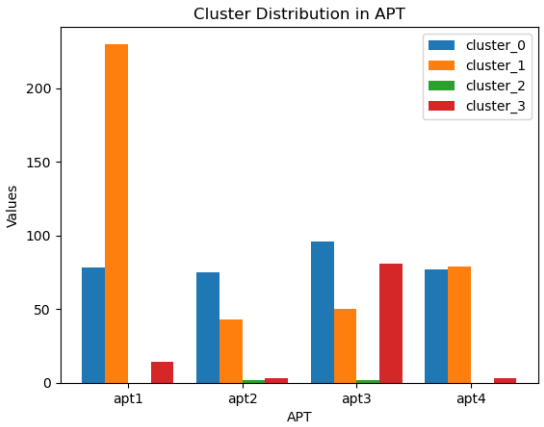


Figure 5.19: Reconnaissance clustering distribution using selected features

APT1 is highly concentrated in clusters 1 (230 instances) and 0 (78 instances), with a small presence in cluster 3 (14 instances), and it does not appear in cluster 2 at all. APT2 has a more balanced distribution across clusters 0 (75 instances) and 2 (43 instances), with very few instances

in clusters 3 (3) and 4 (2). APT3 is more evenly distributed across clusters 0 (96), 3 (81), and 2 (50), with only a small number in cluster 3 (2), making it the APT with the highest number of points in cluster 0. APT4 is very closely distributed in cluster 1 (79 instances) and cluster 0 (77 instances), with minimal representation in cluster 1 (3) and none in cluster 3.

Pivoting

The APTs distribution using the pivoting sub-datasets in the clusters can be observed in Fig.5.20 and in Fig.5.21. Here, it can be seen that cluster 3 is the dominant cluster across the APTs, as it is the cluster with the most points in each of the APTs, while cluster 1 is the least pronounced, obtaining the fewest points.

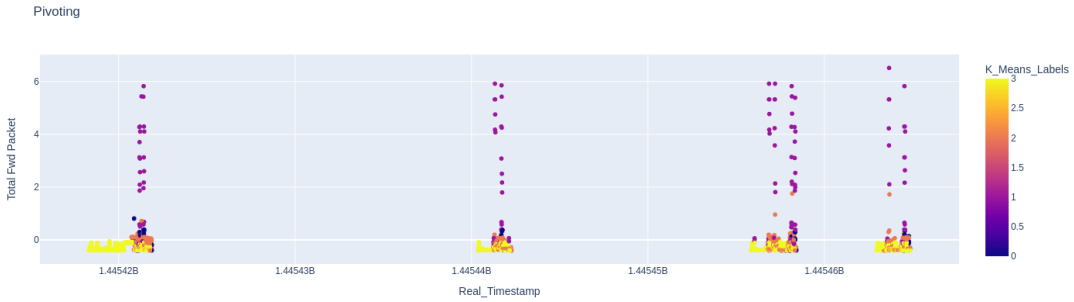


Figure 5.20: Pivoting clustering using selected features

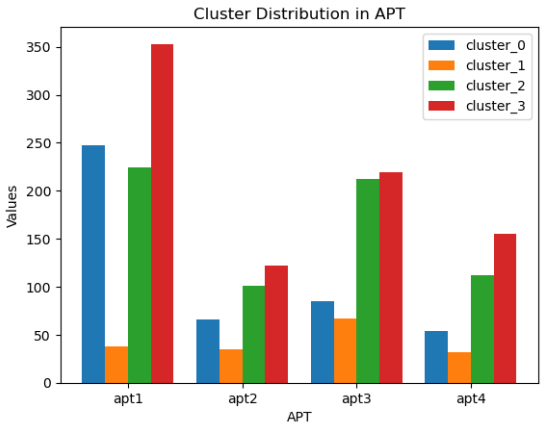


Figure 5.21: Pivoting clustering distribution using selected features

APT1 distribution is heavily skewed towards cluster 3 (353 attacks) and cluster 0 (247 attacks), making these clusters the dominant ones for APT1. Cluster 2 also shows a significant contribution of 224 attacks, whereas cluster 1 has the fewest at 38. APT2 follows a similar pattern to APT1, though on a smaller scale. Cluster 3 (122 attacks) and cluster 2 (101 attacks), cluster 0 captures 66 attacks, which is a moderate contribution, while Cluster 3 is again underrepresented with just 35 attacks. The same behavior is observed in APT4, with cluster 0 (155 instances) and cluster 2 (112 instances) being the dominant clusters, followed by cluster 0 (54 instances) and, lastly,

cluster 1 (32 instances), which has the fewest points. The same happens in APT3 with the gap between cluster 3 and cluster 2 shrinks, with cluster 3 having 219 instances and cluster 2 having 212 instances, the gap between cluster 0 and 1 is also small with respectively having 85 and 67 instances.

Lateral Movement

The distribution of the attack lateral movement sub-dataset is shown below in Fig.5.22 and Fig.5.22. In this sub-dataset, clusters 0 and 2 are predominant, containing the most data points, with cluster 3 having a minor presence in the clustering.

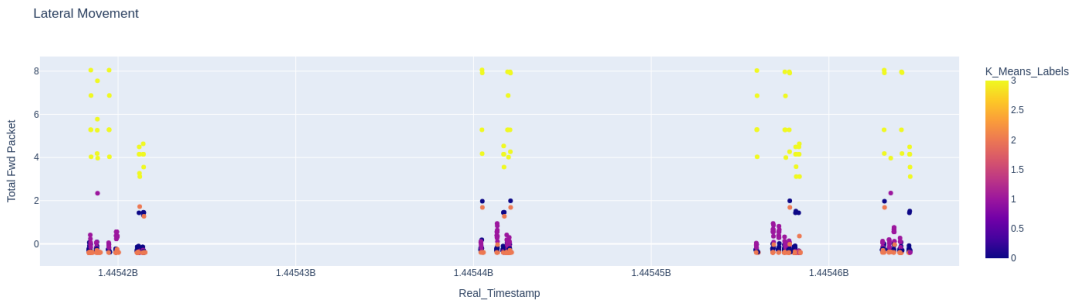


Figure 5.22: Lateral Movement clustering using selected features

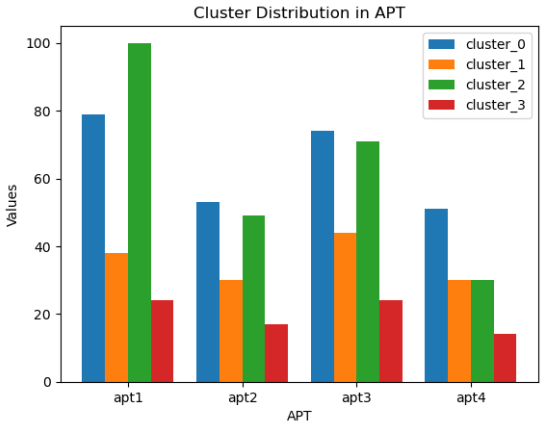


Figure 5.23: Lateral Movement clustering distribution using selected features

The APT1 cluster 2 contains the highest number of attacks (100), suggesting that this cluster might represent a significant subset of APT1's characteristics. Cluster 0 is the second-largest cluster, with 79 attacks, while clusters 1 and 3 are relatively smaller, with clusters 1 (38) and 3 (24) having fewer instances. The APT2 distribution is slightly more balanced than that of APT1, with cluster 0 being dominant, but with a relatively even spread across the other clusters. Cluster 0 has the highest number of attacks (53), indicating that it might be the most characteristic cluster for APT2. Cluster 1 (30) and cluster 2 (49) are relatively balanced, but cluster 3 (17) is much smaller. Similar to APT1, APT3 exhibits a strong concentration of attacks in clusters 0 (74) and 2 (71), with

cluster 1 (44) showing a moderate number of attacks, and Cluster 3 (24) being the smallest. APT4 similarly to APT2 has a relatively evenly distributed attack count across the clusters compared to APT1 and APT3, but cluster 0 still dominates slightly with 51 instances, and clusters 1 and 2 have equal and smaller numbers (30 each) cluster 3 (14) remains the smallest cluster.

Data Exfiltration

The distribution of the attack pivoting sub-dataset is presented below in Fig.5.24 and Fig.5.25. Cluster 0 is dominating in APT1, while cluster 2 is dominating in the other APTs.

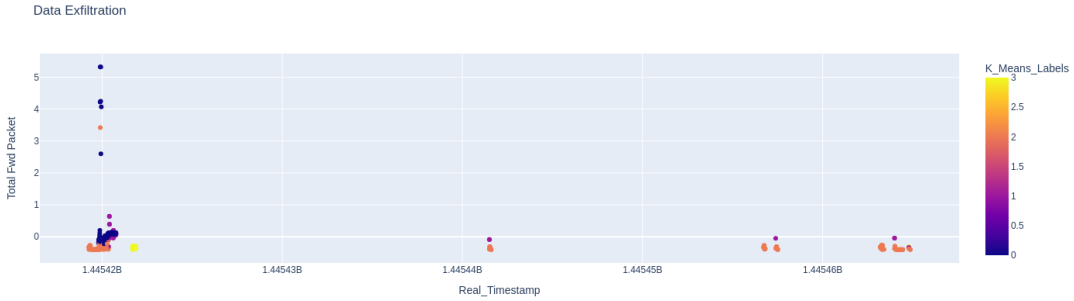
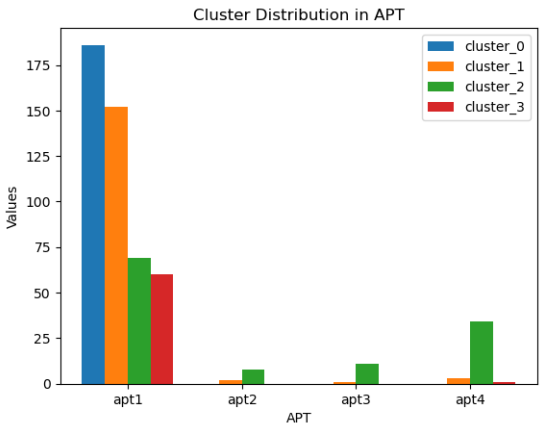


Figure 5.24: Data Exfiltration clustering using selected features

In APT1, the attacks are well distributed across the clusters, with clusters 0 (186 instances) and 1 (152 instances) accounting for the majority of the attacks. Cluster 2 (69 instances) and cluster 3 (60 instances) are less populated but still contain significant counts. In APT2, the number of attacks is quite sparse, with only a few attacks scattered across cluster 1 and cluster 2. Similarly, APT3 has very few attacks across the clusters, with the majority concentrated in cluster 2. Neither APT has a presence in clusters 0 and 3. In APT4, most of the data points are found in cluster 2, with a very small number in cluster 1 and cluster 3, and without presence in cluster 0.



m

Figure 5.25: Data Exfiltration clustering distribution using selected features

5.2.3 GA Features

This section presents the results obtained when the features used for clustering are selected using the NSGA-II algorithm. The termination criterion for NSGA-II used in the feature selection process is defined to terminate if the sum of the best results across the multi-objective optimization remains unchanged over 150 generations. This condition ensures that the algorithm stops when no further improvements can be made, preventing unnecessary computations and promoting computational efficiency while still achieving optimal results.

In Table 5.12, the results from the objective function are presented, where it can be observed that none of the APT phases obtained a total separation of the clusters (objective function equals 0). Data exfiltration obtained the lowest values (0.875), indicating that APTs are well-separated, with different clusters dominating each APT, but still presenting some noise. Initial Compromise and Pivoting present moderate results, indicating that some cluster dominance is achieved, but there is still some overload. Lateral Movement and Reconnaissance present the worst results, indicating no dominant cluster for the APTs, the algorithm struggled to assign memberships to the attack's points, which can stem from overlapping behavior across the APTs.

Table 5.12: NSGA-II objective function results

Attack	Result
Pivoting	1.406
Initial Compromise	1.359
Data Exfiltration	0.875
Reconnaissance	1.934
Lateral Movement	1.721

Initial Compromise

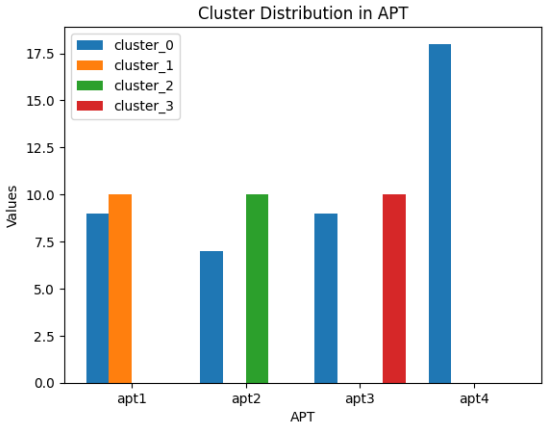


Figure 5.26: Initial Compromise clustering distribution using GA features

In Fig.5.26, the results from the clustering of the initial compromise data points, using the features 'Dst IP_172.28.128.11', 'Src Port_34606', 'Src Port_38429', 'Src Port_49208', 'Src Port_53664',

'Src Port_53760' can be observed.

Cluster 0 is the dominant cluster in the clustering. In APT1, only clusters 0 and 1 have data points, with cluster 0 containing 10 points and cluster 1 containing 9 points. In APT2, the points are again distributed across two clusters, 0 and 2, with cluster 2 having the majority of the APT points. APT3 is distributed in clusters 0 and 3, with cluster 3 having the majority of the points. Contrary to the other APTs, APT4 contains points in a single cluster.

In the scenario, the result aligns most closely with the objective of having one dominant cluster per APT. However, a small spillover into Cluster 0 for APT1, APT2, and APT3 exists, but there are no points in Clusters 2 and 3, which is an improvement towards achieving the desired distribution.

Reconnaissance

In Fig.5.27, the results from the clustering of the reconnaissance data points, using the features 'Fwd Header Length', 'FWD Init Win Bytes', 'Src Port_36058 ', 'Src Port_44048', 'Src Port_60766', can be observed.

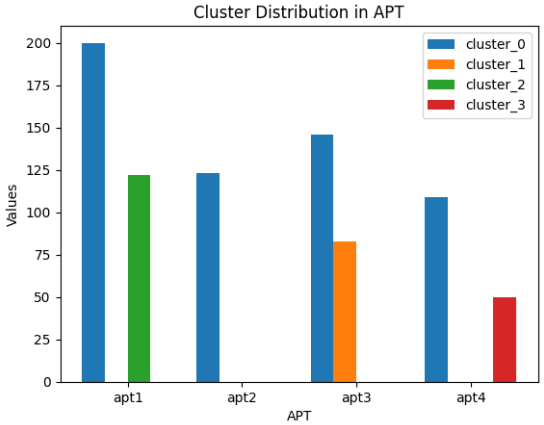


Figure 5.27: Reconnaissance clustering distribution using GA features

Cluster 0 is the cluster with the most points, having points from all of the APTs, and is the dominant cluster in all of the APTs. APT1 presents 200 instances in cluster 0 and 122 in cluster 2, being the only APT with points in cluster 2. APT3 exhibits the same behavior, with 146 points in cluster 0 and the rest in cluster 1, making it the only APT with points in cluster 1. This also occurs in APT4, being the only APT with points in cluster 3, but with cluster 0 having the majority of the APT points. Unlike the other APTs, APT2 has all of her points in cluster 0.

Here, it's observed that APT2 has a perfect single-cluster dominance. Other APTs (APT1, APT3, and APT4) have a dominant cluster but still have a considerable number of points in one other cluster. The split is mainly between Cluster 0 and another cluster, which is not aligned with the objective of having a different dominant cluster by APT.

Pivoting

In Fig.5.28, the results from the clustering of the pivoting data points, using the features 'Timestamp', 'FWD Init Win Bytes', 'Src Port_49208', 'Src Port_53664', 'Src Port_58474', 'Dst Port_43007', can be observed.

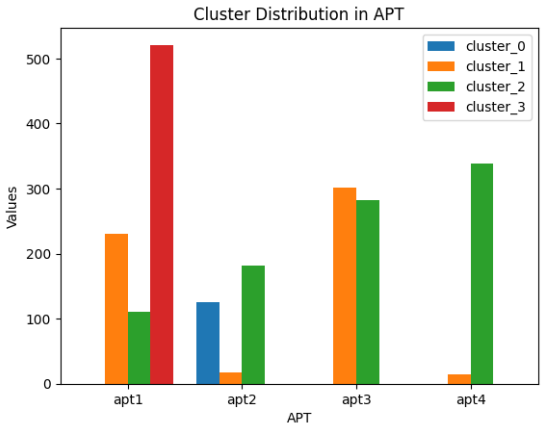


Figure 5.28: Pivoting clustering distribution using GA features

The distribution of the pivoting points across the several clusters for each APT is done as follows: APT1 has the bulk of its points in cluster 3 (521 points), being the only APT with points in this cluster; it also presents 231 instances in cluster 1 and 110 cases in cluster 2. APT2 also presents its point dispersed across 3 clusters, having 182 instances in cluster 2, 124 in cluster 0, and 17 points in cluster 1, making this APT the only one with occurrences in cluster 0. APT3 is scattered in cluster 1 and cluster 2, with cluster 1 having a slightly higher number of instances. APT4 has almost all of its samples concentrated in cluster 2 (338 cases) and a minimal number, 15 points, in cluster 1.

The feature selection using the GA approach, in regard to the pivoting attack, shows the best alignment with the objective, with APT1 being dominated by cluster 3, APT3 cluster 1, even though APT2 and APT3 share the same dominating cluster, and APT1 still has considerable points in other clusters.

Lateral Movement

In Fig.5.29, the results from the clustering of the lateral movement data points, using the features 'Timestamp', 'Src Port_34362', 'Src Port_52150', 'Src Port_52586', 'Dst Port_61715', can be observed. Here, it can be observed that APT1 has points in all the clusters, with clusters 0 and 2 having a large proportion of the data for this APT, with cluster 0 having slightly more points than cluster 2. Clusters 1 and 3 have almost the same number of data points, with the former having the highest amount of APT data. APT3 also has points in all the clusters, but unlike APT1, it has a clear dominant cluster, cluster 3, with clusters 0 and 3 having basically the same amount of data, and cluster 1 having slightly more points than the other two clusters. APT2 presents the bulk of its data being distributed in clusters 1 and 3, and a small amount of data is present in cluster 2,

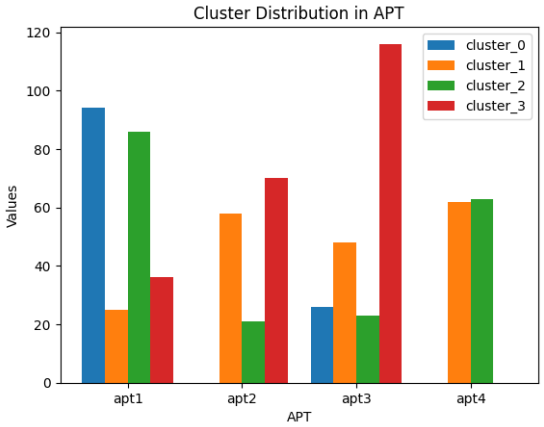


Figure 5.29: Lateral movement clustering distribution using GA features

with cluster 0 not having any points. APT4 has its points evenly distributed across clusters 1 and 2, without any evident dominance of either cluster.

Data Exfiltration

In Fig.5.30, the results from the clustering of the data exfiltration data points, using the features 'Flow Bytes/s ', 'Src Port_54444', 'Src Port_59912', 'Src Port_60736', 'Src Port_64704', can be observed.

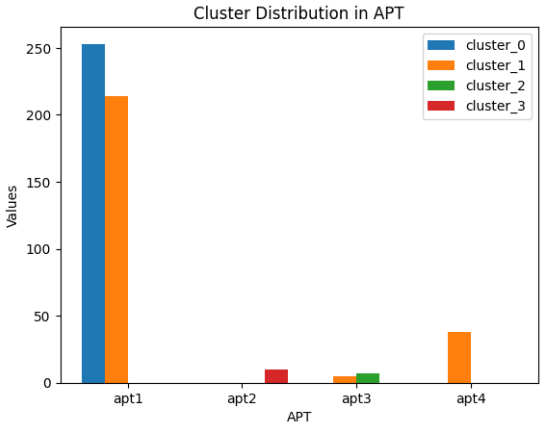


Figure 5.30: Data Exfiltration clustering distribution using GA features

APT1 has its data points allocated in clusters 0 and 1, with cluster 0 having the highest number of instances, 252, and cluster 1 falling slightly behind with 214 cases, being the only APT with points in cluster 1. APT3 also has it's points distributed in two clusters 1 (7 instances) and 3 (5 cases). APT2 and APT4 contain all their points in a single cluster, cluster 3 and cluster 1, respectively.

GA Features results show better cluster dominance for APT2 and APT4, but still suffer from split points in APT1 and APT3, especially in APT1, where Cluster 0 has a large portion of points.

5.2.4 Distances

Starting by analyzing the distance between the centroids of the same attack as observed in the Fig.5.31, examining first the initial compromise phase. This phase presents a reasonably tight clustering with a range between 1.48 (distance between APT3 and APT4) and 2.65 (distance between APT1 and APT4), indicating a similarity in the way the initial breach is conducted. Overall, the behavior during the initial compromise stage is consistent between the different groups, although there are small variations.

Moving on to the reconnaissance phase, the distance between the centroids is wider, varying from 1.64 (APT2 and APT4) up to 4.23 (APT1 and APT3). APT3 shows a clear divergence from the other APTs, suggesting the use of different techniques or the focus on a distinct target. In contrast, APT1, APT2, and APT4 maintain relatively close behavioral patterns. These groups of attacks present a new pattern with 3 of the APTs operating in similar ways, while one of them acts differently.

The Lateral Movement phase presents a different picture. The APT groups are extremely close, with distances varying between 1.44 (APT1 and APT3) and 1.91 (APT1 and APT4). These indicate that the strategies are nearly identical across all apts.

During the pivoting phase, the APT behavior once again converges tightly, with small distances between 0.56 (APT2 and APT4) and 2.03 (APT1 and APT4). APT2 and APT4 are especially close, almost identical, hinting that the strategies for expanding control within the compromised system are highly similar. Overall, pivoting shows a very low degree of variation among the groups, similar to what was observed during lateral movement.

Finally, when examining the data exfiltration, a wider spread emerges again between the distance among the APTs, presenting a range from 2.17 (APT2 and APT3) to 5.45 (APT1 and APT4). APT2 and APT3 remain relatively close to each other, suggesting similar behavior, on the other hand APT stands out with significantly large distances to the others APTs. Indicating that APT1 stands out in the techniques used, separating its behavior from the rest of the group.

Overall, it can be observed that the Lateral Movement and Pivoting phases exhibit extremely high similarity among APT behaviors, suggesting the use of common methods. Initial compromise shows moderate similarity, indicating some variations in how attackers first gain access. Reconnaissance and especially Data Exfiltration reveal more divergence, with certain groups standing out for using different strategies. In particular, APT3 behaves differently during reconnaissance, while APT1 is notably distinct during data exfiltration.

Focusing on each APT across all its attack stages as presented in Fig.5.32, focusing on whether the phases within each APT are closer to one another or whether individual stages are more similar to other APTs' phases. Apt1's five stages form a fairly coherent cluster. Reconnaissance is the closest to Lateral Movement and Pivoting, and Data Exfiltration is moderately close to all the phases, and especially close to Pivoting. Initial compromise is the most distant from its peers, suggesting that the tactics used are distinct from those of the other attacks.

APT2 is also fairly consistent, with all distances between APT2 phases falling in a narrow

range of 2.86 to 7.81. Reconnaissance, Lateral Movement, Pivoting, and Data Exfiltration are especially tight. Like APT1, initial compromise is the most behaviorally distinct, but the group overall maintains a consistent signature.

APT3 shows moderate internal variation, reconnaissance is closer to APT2 reconnaissance (3.97), more distant from its own lateral movement (4.29) than other APTs' reconnaissance phase, indicating weak intra-APT cohesion. Pivoting and data exfiltration phases are more cohesive internally, presenting a distance of 3.50. Lateral movement (distance of 4.29 from Reconnaissance, 7.08 from Initial Compromise) sits somewhere in between. APT3 exhibits less internal coherence, especially between early and late stages. It is the least self-similar APT in the set.

APT4 maintains moderate consistency within its phases, reconnaissance is relatively close to lateral movement, 4.29 and 2.97 to pivoting showing reasonable cohesion. Pivoting and data exfiltration are also close 3.62. Initial compromise once again is the furthest from the other attacks in the APT, with the closest attack being pivoting at 6.01. APT4 is mostly internally consistent, especially in its later stages (pivoting, exfiltration). Early stages show slightly more variation and possible overlap with other APTs, suggesting shared toolkits or tactics in the initial breach.

Overall, the APTs present a moderate consistence within its phases, presenting the same patterns in the several APTs, initial compromise is the attack phase more distant, indicating a degree of separation between initial compromise and the subsequent phases, pivoting is the phase of the APTs that is closet to all the other phases indicating a degree of similarly, the rest of the phases present more variation, with reconnaissance normally being the closest to pivoting and data exfiltration but in APT1 is the closest to lateral movement and pivoting. The overall centroid distances are illustrated in Fig.7.

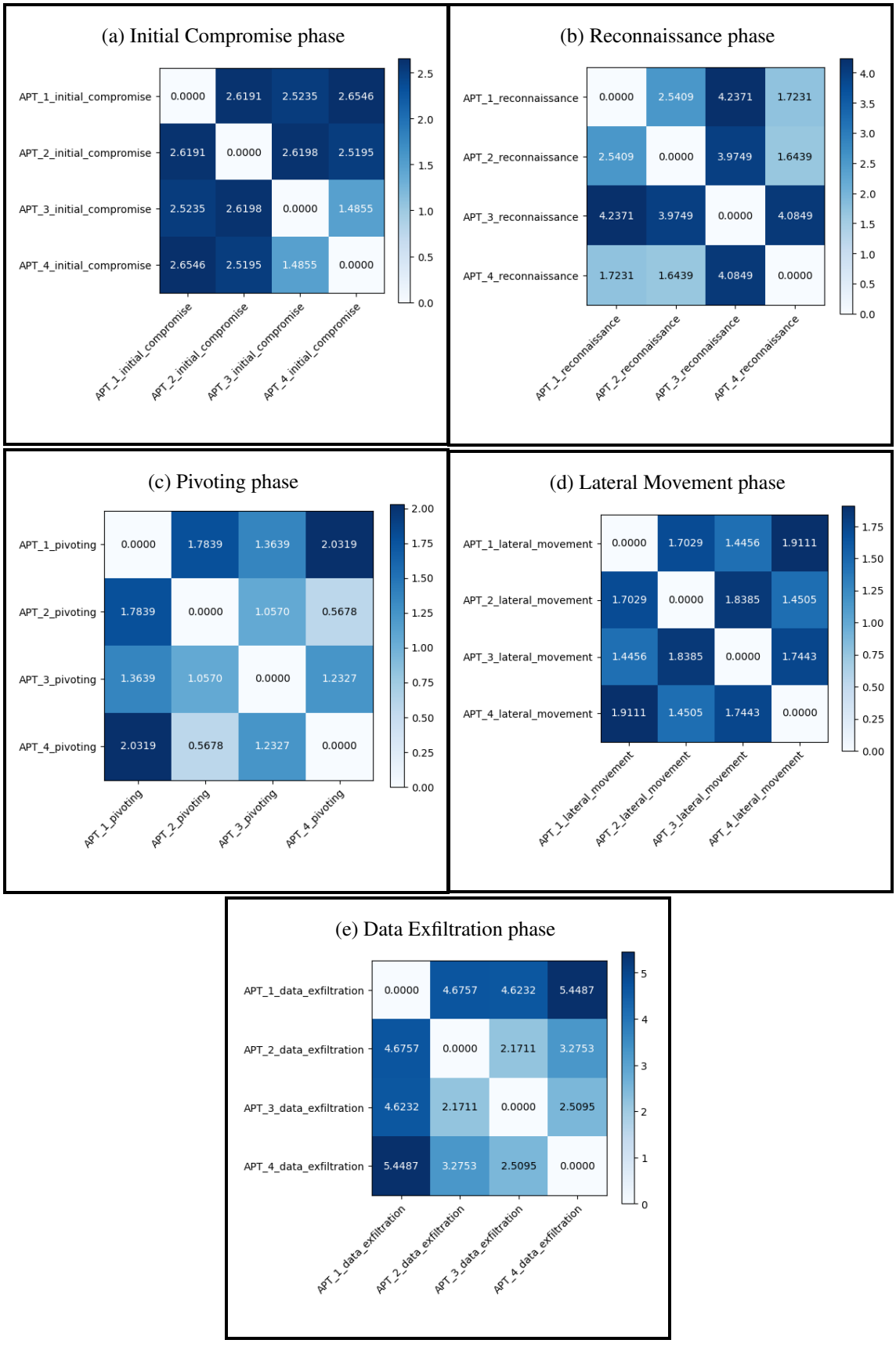


Figure 5.31: Distance between centroids in each attack phase

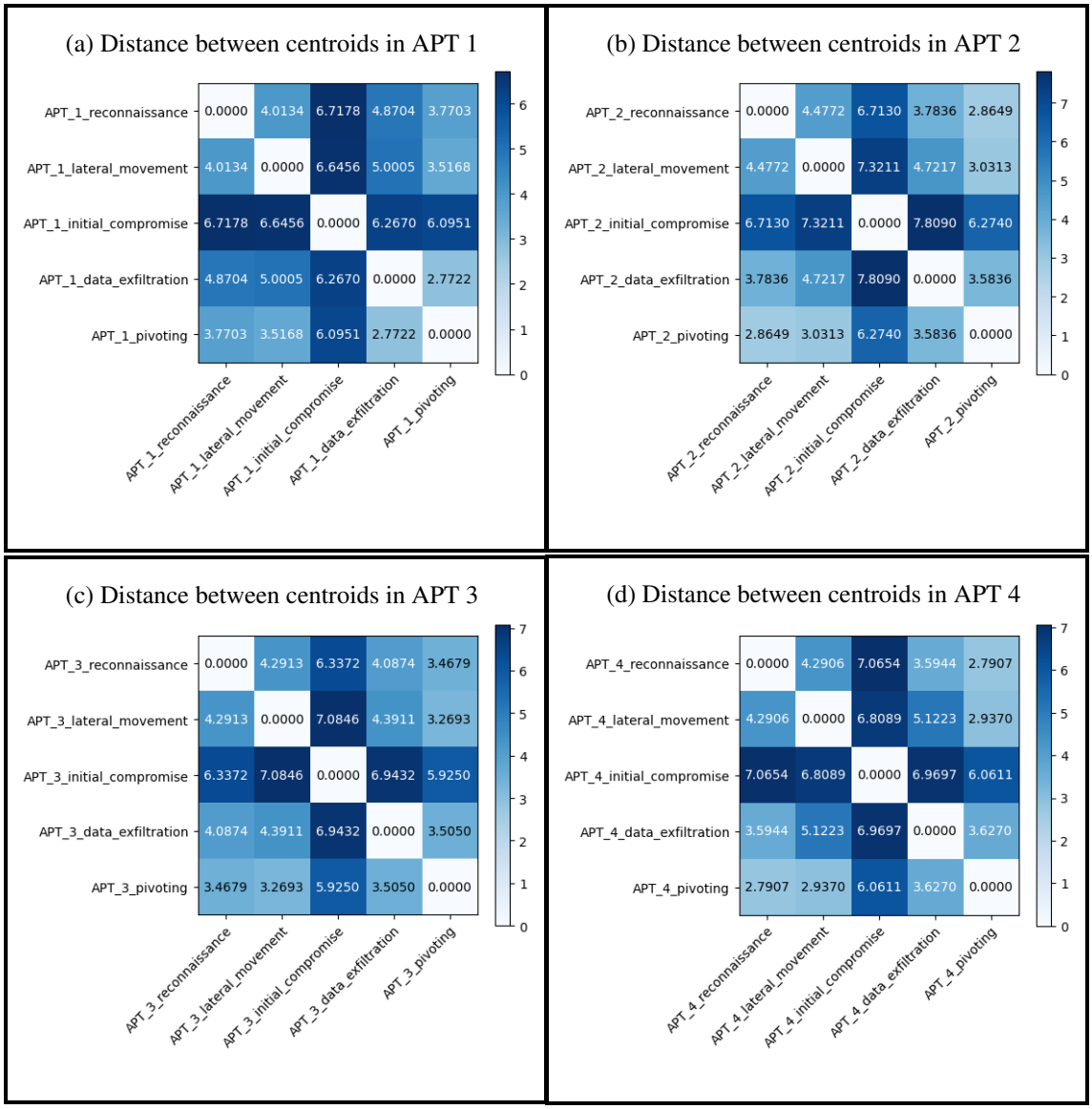


Figure 5.32: Distance between APTs centroids

Chapter 6

Conclusion and Future Work

This thesis set out to perform an analysis of the existing OT cybersecurity datasets, the development of an intrusion detection based on ML, and the exploration of a technique for attack correlation. Through this comprehensive approach, this work offers valuable insights and practical methodologies designed to enhance cyber defense mechanisms within critical infrastructures.

The evaluation of various ML models for intrusion detection revealed that ensemble methods, particularly RF and XGB, consistently outperformed other algorithms across most attack categories, especially when using all features of the dataset. Their ensemble architecture enables them to effectively model complex feature interactions and handle high-dimensional data, which contributes to their superior classification capabilities. However, feature reduction using 30 PCA components was found to be ineffective, as it led to a noticeable decline in classification performance across all models, indicating that important discriminative information was lost during dimensionality reduction. A comparative analysis with the best-performing models from a related study [2] demonstrated that while specialized models, such as GRU, SVM, and DT, outperformed XGB in some specific attack classes, XGB offered near-equivalent performance across most categories. More importantly, XGB's advantage lies in its unified and consistent approach, which simplifies deployment and maintenance by avoiding the complexity of managing multiple specialized models. This trade-off between slightly higher per-class accuracy and overall system efficiency highlights XGB as a highly practical and effective solution for real-world intrusion detection in OT environments.

Regarding attack correlation, clustering with all features often led to dominant clusters, especially for APT1 and APT2, but lacked nuance and over-concentrated on specific behaviors. Manually selected features improved balance, revealing more behavioral diversity, particularly for APT3 and APT4. GA selected features provided the clearest separation, aligning more consistently with the goal of one dominant cluster per APT per phase. Overall, feature selection, especially via genetic algorithms, enhanced clustering quality and interpretability by surfacing distinct APT behavior patterns.

The analysis of Euclidean distances between centroid representations of APT attack stages provides valuable insight into the structural and behavioral patterns of different threat actors. The results show that stages such as pivoting and lateral movement often exhibit stronger internal co-

hesion within each APT, while stages like reconnaissance and initial compromise vary more significantly across APTs, highlighting differences in initial access strategies and early engagement behaviors.

However, to deepen the reliability and applicability of these findings, further investigation is warranted. A richer dataset encompassing a greater number of attacks, more APT groups, and a more balanced distribution of samples would allow for more robust statistical inferences. Additionally, incorporating detailed subcategories of techniques rather than aggregating stages into broader labels could reveal subtle but critical behavioral nuances. Conducting this analysis over a longer time span and across a wider, more heterogeneous network of campaigns would better reflect the evolving tactics of adversaries and support more granular threat attribution and proactive defense strategies.

Bibliography

[1] Amirhossein Ahmadi, Mojtaba Nabipour, Saman Taheri, Behnam Mohammadi-Ivatloo, and Vahid Vahidinasab. A new false data injection attack detection model for cyberattack resilient energy forecasting. *IEEE Transactions on Industrial Informatics*, 19(1):371–381, 2023.

[2] Muna Al-Hawawreh, Elena Sitnikova, and Neda Aboutorab. X-iiotid: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things. *IEEE Internet of Things Journal*, 9(5):3962–3977, 2022.

[3] Noura Alenezi and Ahamed Aljuhani. Intelligent intrusion detection for industrial internet of things using clustering techniques. *Computer Systems Science and Engineering*, 46:1–17, 01 2023.

[4] N. S. Altman. *An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression*, volume 46. 1992.

[5] Hakan Can Altunay and Zafer Albayrak. A hybrid cnn+lstm-based intrusion detection system for industrial iot networks. *Engineering Science and Technology, an International Journal*, 38:101322, 2023.

[6] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, page 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[7] Mohammad Ashrafuzzaman, Saikat Das, Yacine Chakhchoukh, Sajjan Shiva, and Frederick T. Sheldon. Detecting stealthy false data injection attacks in the smart grid using ensemble-based machine learning. *Computers & Security*, 97:101994, 2020.

[8] Giuseppe Bernieri, Mauro Conti, and Federica Pascucci. Mimepot: a model-based honeypot for industrial control networks. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 433–438, 2019.

[9] Giuseppe Bernieri, Mauro Conti, and Federico Turrin. Evaluation of machine learning algorithms for anomaly detection in industrial networks. In *2019 IEEE International Symposium on Measurements & Networking (M&N)*, pages 1–6, 2019.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[11] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[12] Tim M. Booij, Irina Chiscop, Erik Meeuwissen, Nour Moustafa, and Frank T. H. den Hartog. Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets. *IEEE Internet of Things Journal*, 9(1):485–496, 2022.

[13] Timothy Chadza, Konstantinos G. Kyriakopoulos, and Sangarapillai Lambotharan. Analysis of hidden markov model learning algorithms for the detection and prediction of multi-stage network attacks. *Future Generation Computer Systems*, 108:636–649.

[14] Oumaima Chakir, Abdeslam Rehaimi, Yassine Sadqi, El Arbi Abdellaoui Alaoui, Moez Krichen, Gurjot Singh Gaba, and Andrei Gurtov. An empirical assessment of ensemble methods and traditional machine learning techniques for web-based attack detection in industry 5.0. *Journal of King Saud University - Computer and Information Sciences*, 35(3):103–119, 2023.

[15] Ping Chen, Lieven Desmet, and Christophe Huygens. A study on advanced persistent threats. In Bart De Decker and André Zúquete, editors, *Communications and Multimedia Security*, pages 63–72, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794. ACM, 2016.

[17] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. 2016.

[18] Samin Y. Chowdhury, Brandon Dudley, and Ruimin Sun. The case for virtual plc-enabled honeypot design. In *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 351–357, 2023.

[19] R. K. Cunningham, R. P. Lippmann, D. J. Fried, S. L. Garfinkel, I. Graf, K. R. Kendall, S. E. Webster, D. Wyschogrod, and M. A. Zissman. Evaluating intrusion detection systems without attacking your friends: The 1998 darpa intrusion detection evaluation. 1998.

[20] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[21] Alireza Dehlaghi-Ghadim, Mahshid Helali Moghadam, Ali Balador, and Hans Hansson. Anomaly detection dataset for industrial control systems. *IEEE Access*, 11:107982–107996, 2023.

[22] Raisa Abedin Disha and Sajjad Waheed. A comparative study of machine learning models for network intrusion detection system using unsw-nb 15 dataset. In *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, pages 1–5, 2021.

[23] Erik Miguel de Elias, Vinicius Sanches Carriel, Guilherme Werneck De Oliveira, Aldri Luiz Dos Santos, Michele Nogueira, Roberto Hirata Junior, and Daniel Macêdo Batista. A hybrid cnn-lstm model for iiot edge privacy-aware intrusion detection. In *2022 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6, 2022.

[24] Yushan Fang, Yu Yao, Xiaoli Lin, Jiaxuan Wang, and Hao Zhai. A feature selection based on genetic algorithm for intrusion detection of industrial control systems. *Computers Security*, 139:103675, 2024.

[25] Mohamed Amine Ferrag, Othmane Friha, Djallel Hamouda, Leandros Maglaras, and Helge Janicke. Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning. *IEEE Access*, 10:40281–40306, 2022.

[26] Javier Franco, Ahmet Aris, Berk Canberk, and A. Selcuk Uluagac. A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. *IEEE Communications Surveys & Tutorials*, 23(4):2351–2383, 2021.

[27] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 1*, 2:559–572, 1901.

[28] Giovanni Battista Gaggero, Alessandro Armellin, Giancarlo Portomauro, and Mario Marchese. Industrial control system-anomaly detection dataset (ics-add) for cyber-physical security monitoring in smart industry environments. *IEEE Access*, 12:64140–64149, 2024.

[29] Erfan Ghiasvand, Suprio Ray, Shahrear Iqbal, Sajjad Dadkhah, and Ali A. Ghorbani. Cicapt-iiot: A provenance-based apt attack dataset for iiot environment. *arXiv preprint arXiv:2407.11278*, July 2024.

[30] Jonathan Goh, Sridhar Adepu, Khurum Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. pages 88–99, 11 2017.

[31] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning: Data mining, inference, and prediction. *Springer Series in Statistics*, 2009.

[32] Pilar Holgado, Víctor A. Villagrá, and Luis Vázquez. Real-time multistep attack prediction based on hidden markov models. *IEEE Transactions on Dependable and Secure Computing*, 17(1):134–147, 2020.

[33] Martin Husák, Jana Komárková, Elias Bou-Harb, and Pavel Čeleda. Survey of attack projection, prediction, and forecasting in cyber security. *IEEE Communications Surveys & Tutorials*, 21(1):640–660, 2019.

[34] Sanjana Ingale, Milind Paraye, and Dayanand Ambawade. Enhancing multi-step attack prediction using hidden markov model and naive bayes. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 36–44, 2020.

[35] Sanjana Ingale, Milind Paraye, and Dayanand Ambawade. A survey on methodologies for multi-step attack prediction. In *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, pages 37–45, 2020.

[36] Urslla Uchechi Izuazu, Vivian Ukamaka Ihekoronye, Dong-Seong Kim, and Jae Min Lee. Securing critical infrastructure: A denoising data-driven approach for intrusion detection in ics network. In *2024 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 841–846, 2024.

[37] Dishan Jing and Hai-Bao Chen. Svm based network intrusion detection for the unsw-nb15 dataset. In *2019 IEEE 13th International Conference on ASIC (ASICON)*, pages 1–4, 2019.

[38] Amit V Kachavimath, Shubhangeni Vijay Nazare, and Sheetal S Akki. Distributed denial of service attack detection using naïve bayes and k-nearest neighbor for network forensics. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 711–717, 2020.

[39] Yakub Kayode Saheed, Oluwadamilare Harazeem Abdulganiyu, and Taha Ait Tchakoucht. A novel hybrid ensemble learning for anomaly detection in industrial sensor networks and scada systems for smart city infrastructures. *Journal of King Saud University - Computer and Information Sciences*, 35(5):101532, 2023.

[40] Kushagra Keserwani, Apoorva Aggarwal, and Anamika Chauhan. Attack detection in industrial iot using novel ensemble techniques. In *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, pages 1–6, 2023.

[41] Amina Khacha, Rafika Saadouni, Yasmine Harbi, and Zibouda Aliouat. Hybrid deep learning-based intrusion detection system for industrial internet of things. In *2022 5th International Symposium on Informatics and its Applications (ISIA)*, pages 1–6, 2022.

[42] Gary King and Langche Zeng. Logistic regression in rare events data. *Political Analysis*, 9(2):137–163, 2001.

[43] Manasa Koppula and Leo Joseph L. M. I. Lnkdsea: Machine learning based iot/iiot attack detection method. In *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, pages 655–662, 2023.

[44] Moshe Kravchik and Asaf Shabtai. Detecting cyber attacks in industrial control systems using convolutional neural networks. pages 72–83, 10 2018.

[45] Ayush Kumar and Vrizlynn Thing. Raptor: Advanced persistent threat detection in industrial iot via attack stage correlation, 01 2023.

[46] Fariba Laiq, Feras Al-Obeidat, Adnan Amin, and Fernando Moreira. Ddos attack detection in edge-iiot using ensemble learning. In *2023 7th Cyber Security in Networking Conference (CSNet)*, pages 204–207, 2023.

[47] Thi-Thu-Huong Le, Yustus Oktian, and Howon Kim. Xgboost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems. *Sustainability*, 14:8707, 07 2022.

[48] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[49] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In Igor V. Tetko, Věra Kůrková, Pavel Karpov, and Fabian Theis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*, pages 703–716, Cham, 2019. Springer International Publishing.

[50] Teng Li, Ya Jiang, Chi Lin, Mohammad S. Obaidat, Yulong Shen, and Jianfeng Ma. Deepag: Attack graph construction and threats prediction with bi-directional deep learning. *IEEE Transactions on Dependable and Secure Computing*, 20(1):740–757, 2023.

[51] Yueyang Li, Wenjun Fan, and Ruxue Luo. Machine learning-based approach for enhancing multi-step attack prediction. In *2023 24st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 48–53, 2023.

[52] Samuel Litchfield, David Formby, Jonathan Rogers, Sakis Meliopoulos, and Raheem Beyah. Rethinking the honeypot for cyber-physical systems. *IEEE Internet Computing*, 20(5):9–17, 2016.

[53] Jinxin Liu, Yu Shen, Murat Simsek, Burak Kantarci, Hussein T. Mouftah, Mehran Bagheri, and Petar Djukic. A new realistic benchmark for advanced persistent threats in network traffic. *IEEE Networking Letters*, 4(3):162–166, 2022.

[54] Sam Maesschalck, Vasileios Giotsas, Benjamin Green, and Nicholas Race. Don't get stung, cover your ics in honey: How do honeypots fit within industrial control system security. *Computers & Security*, 114:102598, 2022.

[55] Inês Martins, José Cecílio, Pedro M. Ferreira, and Alan Oliveira. Comparative analysis of cybersecurity datasets in industrial control systems. In *2024 IEEE International Workshop on Metrology for Industry 4.0 IoT (MetroInd4.0 IoT)*, pages 440–445, 2024.

[56] Aditya P. Mathur and Nils Ole Tippenhauer. Swat: a water treatment testbed for research and training on ics security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pages 31–36, 2016.

[57] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[58] Ammar Mohammed and Rania Kora. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*, 35(2):757–774, 2023.

[59] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6, 2015.

[60] Nour Moustafa and Jill Slay. The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. pages 1–14, 01 2016.

[61] Sowmya Myneni, Ankur Chowdhary, Abdulhakim Sabur, Sailik Sengupta, Garima Agrawal, Dijiang Huang, and Myong Kang. Dapt 2020 - constructing a benchmark dataset for advanced persistent threats. In Gang Wang, Arridhana Ciptadi, and Ali Ahmadzadeh, editors, *Deployable Machine Learning for Security Defense*, pages 138–163, Cham, 2020. Springer International Publishing.

[62] Sowmya Myneni, Ankur Chowdhary, Abdulhakim Sabur, Sailik Sengupta, Garima Agrawal, Dijiang Huang, and Myong H. Kang. Dapt 2020 - constructing a benchmark dataset for advanced persistent threats. 2020.

[63] Sowmya Myneni, Kritshekhar Jha, Abdulhakim Sabur, Garima Agrawal, Yuli Deng, Ankur Chowdhary, and Dijiang Huang. Unraveled, 2023. Accessed: 2024-12-29.

[64] Sowmya Myneni, Kritshekhar Jha, Abdulhakim Sabur, Garima Agrawal, Yuli Deng, Ankur Chowdhary, and Dijiang Huang. Unraveled — a semi-synthetic dataset for advanced persistent threats. *Computer Networks*, 227:109688, 2023.

[65] Óscar Navarro, Servilio Alonso Joan Balbastre, and Stefan Beyer. Gathering intelligence through realistic industrial control system honeypots. In Eric Luiijf, Inga Žutautaitė, and Bernhard M. Hämmerli, editors, *Critical Information Infrastructures Security*, pages 143–153, Cham, 2019. Springer International Publishing.

[66] Mudhafar Nuaimi, Lamia Chaari Fourati, and Bassem Ben Hamed. Intelligent approaches toward intrusion detection systems for industrial internet of things: A systematic comprehensive review. *Journal of Network and Computer Applications*, 215:103637, 2023.

[67] Ángel Luis Perales Gómez, Lorenzo Fernández Maimó, Alberto Huertas Celdrán, Félix J. García Clemente, Cristian Cadenas Sarmiento, Carlos Javier Del Canto Masa, and Rubén Méndez Nistal. On the generation of anomaly detection datasets in industrial control systems. *IEEE Access*, 7:177460–177473, 2019.

[68] Ángel Luis Perales Gómez, Lorenzo Fernández Maimó, Alberto Huertas Celdrán, Félix J. García Clemente, Cristian Cadenas Sarmiento, Carlos Javier Del Canto Masa, and Rubén Méndez Nistal. On the generation of anomaly detection datasets in industrial control systems. *IEEE Access*, 7:177460–177473, 2019.

[69] Stanislav Ponomarev and Travis Atkison. Industrial control system network intrusion detection by telemetry analysis. *IEEE Transactions on Dependable and Secure Computing*, 13(2):252–260, 2016.

[70] Ali Ahmadian Ramaki, Masoud Khosravi-Farmad, and Abbas Ghaemi Bafghi. Real time alert correlation and prediction using bayesian networks. In *2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, pages 98–103, 2015.

[71] Martha Rodríguez, Diana P. Tobón, and Danny Múnera. Anomaly classification in industrial internet of things: A review. *Intelligent Systems with Applications*, 18:200232, 2023.

[72] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education Inc., 3rd edition, 2009.

[73] Yao Shan, Yu Yao, Tong Zhao, and Wei Yang. Neupot: A neural network-based honeypot for detecting cyber threats in industrial control systems. *IEEE Transactions on Industrial Informatics*, 19(10):10512–10522, 2023.

[74] Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, and Ali A. Ghorbani. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers Security*, 31(3):357–374, 2012.

[75] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, 2018.

[76] Luís Sousa, José Cecílio, Pedro Ferreira, and Alan Oliveira. Reconfigurable and scalable honeynet for cyber-physical systems. *arXiv preprint arXiv:2404.04385*, 2024.

[77] Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, and Philip K. Chan. Kdd-cup-99, 1999.

[78] Marcio Teixeira, Maede Zolanvari, and Raj Jain. Wustl-iiot-2018, 2020.

[79] V. N. Vapnik and A. Ya. Chervonenkis. *On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities*. Springer International Publishing, 2015.

[80] Emmanouil Vasilomanolakis, Shreyas Srinivasa, Carlos Garcia Cordero, and Max Mühlhäuser. Multi-stage attack detection and signature generation with ics honeypots. In

NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, pages 1227–1232, 2016.

[81] Michael Winn, Mason Rice, Stephen Dunlap, Juan Lopez, and Barry Mullins. Constructing cost-effective and targetable industrial control system honeypots for production networks. *International Journal of Critical Infrastructure Protection*, 10:47–58, 2015.

[82] Yang Xin, Lingshuang Kong, Zhi Liu, Yuling Chen, Yanmiao Li, Hongliang Zhu, Mingcheng Gao, Haixia Hou, and Chunhua Wang. Machine learning and deep learning methods for cybersecurity. *IEEE Access*, 6:35365–35381, 2018.

[83] Chunlin Xiong, Tiantian Zhu, Weihao Dong, Linqi Ruan, Runqing Yang, Yueqiang Cheng, Yan Chen, Shuai Cheng, and Xutong Chen. Conan: A practical real-time apt detection system with high accuracy and efficiency. *IEEE Transactions on Dependable and Secure Computing*, 19(1):551–565, 2022.

[84] Abbas Yazdinejad, Mostafa Kazemi, Reza M. Parizi, Ali Dehghantanha, and Hadis Karimipour. An ensemble deep learning model for cyber threat hunting in industrial internet of things. *Digital Communications and Networks*, 9(1):101–110, 2023.

[85] Yagmur Yigit, Omer Kemal Kinaci, Trung Q. Duong, and Berk Canberk. Twinpot: Digital twin-assisted honeypot for cyber-secure smart seaports. In *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 740–745, 2023.

[86] Jianzhou You, Shichao Lv, Yue Sun, Hui Wen, and Limin Sun. Honeyvp: A cost-effective hybrid honeypot architecture for industrial control systems. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021.

[87] Jun Zhang, Lei Pan, Qing-Long Han, Chao Chen, Sheng Wen, and Yang Xiang. Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA Journal of Automatica Sinica*, 9(3):377–391, 2022.

[88] Yong Zhang, Jie Niu, Da Guo, Yinglei Teng, and Xuyan Bao. Unknown network attack detection based on open set recognition. *Procedia Computer Science*, 174:387–392, 01 2020.

[89] RUIZHE ZHAO. Nsl-kdd, 2022.

[90] Aaron Zimba, Hongsong Chen, Zhaoshun Wang, and Mumbi Chishimba. Modeling and detection of the multi-stages of advanced persistent threats attacks based on semi-supervised learning and complex networks characteristics. *Future Generation Computer Systems*, 106:501–517, 2020.

[91] Maede Zolanvari. Wustl-iiot-2021, 2021.

.1 Appendix 1

Table 1: Occurrence of attacks in datasets

Dataset	Attack Category	Attack Subcategory	Occurrence
X-IIOTID	Reconnaissance	Generic Scanning	50277
		Scanning Vulnerability	52852
		Discovering Resources	23148
		Fuzzing	1313
	Weaponization	Brute Force	47241
		Dictionary	2572
		Insider Malicious	17447
	Exploitation	Reverse Shell	1016
		Mitm	117
	LM	Modbus Register	5953
		MQTT	23524
		TCP Relay	2119
	C&C		2863
	Exfiltration		22134
	Ransomware		458
	RDoS		141261
	Tampering	Fake Notification	28
		False Data Injection	5094
ICS-Flow	Mitm		2584
	Replay		2358
	Port-Scan		1944
	DDoS		1934
	IP-Scan		192
WDT	Mitm		2155409
	physical fault		1548504
	anomaly		389
	DoS		5671542
	Scan		61
Electra	Reconnaissance	Code Recognition	30580
	False Data Injection	Response Modification	161245
		Error in response	1665236
		Command modification	575122
		Read Data	1722267
		Write data	2245189
	Replay		27654
SWAT	False Data Injection		54621

Edge-IIoTSet	DoS/DDoS	DDoS_UDP	14498
		DDoS_ICMP	14090
		DDoS_HTTP	10561
		DDoS_TCP	10247
	Information Gathering	Port Scanning	10071
		OS Fingerprinting	1001
		Vulnerability scanning	10076
	MitM		1214
	Injection Attacks	Cross-site Scripting	10052
		SQL Injection	10311
		Uploading file	10269
	Malware	Backdoor	10195
		Password cracking	9989
		Ransomware	10925

.2 Appendix 2.1

Table 2: X-IIoTID features description

Feature	Description
Date	Calendar date when the communication occurred
Timestamp	Timestamp of the first packet
Scr IP	Address from which the data originates within the network flow
Scr port	Port number on the source device used for the communication
Des IP	Address to which the data is directed in the network flow
Des port	Port number on the destination device where the communication is directed
Protocol	The transport layer protocol of the connection
Service	Application protocol being used in the connection.
Duration	How long the connection lasted
Scr bytes	Number of payload bytes the originator sent
Des bytes	Number of payload bytes the responder sent
Conn state	State of the connection
missed bytes	Number of bytes missed in content gaps, which is representative of packet loss
total bytes	Total number of bytes exchanged between sender and receiver
total packet	Total number of packets exchanged between sender and receiver
packet rate	Total number of packets per second
byte rate	Total number of bytes per second
Scr packets ratio	Ratio between packets sent by the sender and total packets
Des pkts ratio	Ratio between packets sent by receiver and total packets
Scr bytes ratio	Ratio between bytes sent by sender and total bytes
Des bytes ratio	Ratio between bytes sent by receiver and total bytes
Scr pkts	Number of packets that the originator sent
Scr ip bytes	Number of IP level bytes that the originator sent
Des pkts	Number of packets that the responder sent
Des ip bytes	Number of IP level bytes that the responder sent
is syn only	Connection has packet with SYN flag
Is SYN ACK	Connection has packet with SYN-ACK flag
is pure ack	Connection has packet with pure ACK flag
is with payload	Connection has packets with payload
FIN or RST	Connection has packet with FIN or RST flag
is SYN with RST	Connection has packets with SYN and RST flags
Bad checksum	Connection has packets with bad checksums
anomaly alert	1 if connection has an alert from zeek, 0 otherwise
Avg user time	Average time of user's process running program/code in the last 10 seconds
Std user time	Standard deviation of user time in the last 10 seconds

Avg nice time	Average of time used for defining the priority process in the last 10 seconds
Std nice time	Standard deviation of nice time
Avg system time	Average of time the processor works at OS function in the last 10 seconds
Std system time	Standard deviation of system time
Avg iowait time	Average of I/O wait time in the last 10 seconds
Std iowait time	Standard deviation of I/O wait time in the last 10 seconds
Avg ideal time	Average of ideal time (CPU is not busy and does not have an outstanding disk I/O request)in the last 10 seconds
Std ideal time	Standard deviation of ideal time in the last 10 seconds
Avg tps	Average of number of transfer requests per second in the device in the last 10 seconds
Std tps	Standard deviation of number of transfer requests per second in the device in the last 10 seconds
Avg rtps	Average of number of read transaction per second issued in the device in the last 10 seconds
Std rtps	Standard deviation of number of read transaction per second issued in the device in the last 10 seconds
Avg wtps	Average of number of write transaction per second issued in the device in the last 10 seconds
Std wtps	Standard deviation of number of write transactions per second issued in the device in the last 10 seconds
Avg ldavg 1	Average of system load during the last minute in window size 10 seconds
Std ldavg 1	Standard deviation of system load during the last minute in window size 10 seconds
Avg kbmemused	Average of used memory in kilobytes in the last 10 seconds
Std kbmemused	Standard deviation of used memory in kilobytes in the last 10 seconds
Avg num Proc/s	Average number of tasks created per second in the last 10 seconds
Std num proc/s	Standard deviation number of tasks created per second in the last 10 seconds
Avg num cswch/s	Average number of context switches per second in the last 10 seconds
std num cswch/s	Standard deviation number of context switches per second in the last 10 seconds
OSSEC alert	1 if connection has an alert from OSSEC, 0 otherwise
OSSEC alert level	OSSEC alert severity level
Login attempt	1 if there is attempt to login, 0 otherwise
Succesful login	1 if a successful login, 0 otherwise
File activity	1 if there is a file activities: 0 otherwise
Process activity	1 if a new process is executed or started: 0 otherwise
read write physical.process	1 if there is read or write activity to the physical process, 0 otherwise
is privileged	1 if performed activity is privileged; 0 otherwise

class1	Exists or not attack traffic
class2	Class of attack
class3	Subclass of attack

.3 Appendix 2.2

Table 3: SCVI-APT-2021 features description

Feature	Description
Flow ID	Unique identifier assigned to each network flow
Src IP	IP address of the device that initiated the communication
Src Port	Port number on the source device used for the communication
Dst IP	IP address of the device intended to receive the communication
Dst Port	Port number on the destination device where the communication is directed
Protocol	Network protocol used for communication in the flow
Timestamp	The exact time at which a packet or flow was captured
Flow Duration	Duration of the flow in Microsecond
Total Fwd Packet	Total packets in the forward direction
Total Bwd packets	Total packets in the backward direction
Total Length of Fwd Packet	Total size of packet in forward direction
Total Length of Bwd Packet	Total size of packet in backward direction
Fwd Packet Length Max	Maximum size of packet in forward direction
Fwd Packet Length Min	Minimum size of packet in forward direction
Fwd Packet Length Mean	Mean size of packet in forward direction
Fwd Packet Length Std	Standard deviation size of packet in forward direction
Bwd Packet Length Max	Maximum size of packet in backward direction
Bwd Packet Length Min	Minimum size of packet in backward direction
Bwd Packet Length Mean	Mean size of packet in backward direction
Bwd Packet Length Std	Standard deviation size of packet in backward direction
Flow Bytes/s	Number of flow bytes per second
Flow Packets/s	Number of flow packets per second
Flow IAT Mean	Mean time between two packets sent in the flow
Flow IAT Std	Standard deviation time between two packets sent in the flow
Flow IAT Max	Maximum time between two packets sent in the flow
Flow IAT Min	Minimum time between two packets sent in the flow
Fwd IAT Total	Total time between two packets sent in the forward direction
Fwd IAT Mean	Mean time between two packets sent in the forward direction
Fwd IAT Std	Standard deviation time between two packets sent in the forward direction

Fwd IAT Max	Maximum time between two packets sent in the forward direction
Fwd IAT Min	Minimum time between two packets sent in the forward direction
Bwd IAT Total	Total time between two packets sent in the backward direction
Bwd IAT Mean	Mean time between two packets sent in the backward direction
Bwd IAT Std	Standard deviation time between two packets sent in the backward direction
Bwd IAT Max	Maximum time between two packets sent in the backward direction
Bwd IAT Min	Minimum time between two packets sent in the backward direction
Fwd PSH Flags	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)
Bwd PSH Flags	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
Fwd URG Flags	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)
Bwd URG Flags	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
Fwd Header Length	Total bytes used for headers in the forward direction
Bwd Header Length	Total bytes used for headers in the backward direction
Fwd Packets/s	Number of forward packets per second
Bwd Packets/s	Number of backward packets per second
Packet Length Min	Minimum length of a packet
Packet Length Max	Maximum length of a packet
Packet Length Mean	Mean length of a packet
Packet Length Std	Standard deviation length of a packet
Packet Length Variance	Variance length of a packet
FIN Flag Count	Number of packets with FIN
SYN Flag Count	Number of packets with SYN
RST Flag Count	Number of packets with RST
PSH Flag Count	Number of packets with PSH
ACK Flag Count	Number of packets with ACK
URG Flag Count	Number of packets with URG
CWR Flag Count	Number of packets with CWR
ECE Flag Count	Number of packets with ECE
Down/Up Ratio	Download and upload ratio
Average Packet Size	Average size of packet
Fwd Segment Size Avg	Average size observed in the forward directio
Bwd Segment Size Avg	Average number of bytes bulk rate in the backward direction
Fwd Bytes/Bulk Avg	Average number of bytes bulk rate in the forward direction
Fwd Packet/Bulk Avg	Average number of bytes bulk rate in the forward direction
Fwd Bulk Rate Avg	Average number of bulk rate in the forward direction

Bwd Bytes/Bulk Avg	Average number of bytes bulk rate in the backward direction
Bwd Packet/Bulk Avg	Average number of packets bulk rate in the backward direction
Bwd Bulk Rate Avg	Average number of bulk rate in the backward direction
Subflow Fwd Packets	The average number of packets in a subflow in the forward direction
Subflow Fwd Bytes	The average number of bytes in a subflow in the forward direction
Subflow Bwd Packets	The average number of packets in a sub flow in the backward direction
Subflow Bwd Bytes	The average number of bytes in a sub flow in the backward direction
FWD Init Win Bytes	The total number of bytes sent in initial window in the forward direction
Bwd Init Win Bytes	The total number of bytes sent in initial window in the backward direction
Fwd Act Data Pkts	Count of packets with at least 1 byte of TCP data payload in the forward direction
Fwd Seg Size Min	Minimum segment size observed in the forward direction
Active Mean	Mean time a flow was active before becoming idle
Active Std	Standard deviation time a flow was active before becoming idle
Active Max	Maximum time a flow was active before becoming idle
Active Min	Minimum time a flow was active before becoming idle
Idle Mean	Mean time a flow was idle before becoming active
Idle Std	Standard deviation time a flow was idle before becoming active
Idle Max	Maximum time a flow was idle before becoming active
Idle Min	Minimum time a flow was idle before becoming active
Label	Labels of attacks

.4 Appendix 3

Table 4: Results from Grid Search in Binary Classification

Scenario	Model	Parameters
30 PCA Components	SVM	C = 1, loss = hinge
	DT	max_depth = 40, criterion = entropy
	LR	C = 1
	KNN	n_neighbors = 6,weights = distance
	NB	var_smoothing = 1e-07
	XGB	gamma = 0.0, learning_rate = 0.01, max_depth = 3, n_estimators = 50
	RF	criterion = entropy, max_depth = 40, n_estimators = 300
All PCA Components	SVM	C = 10, loss = hinge
	DT	max_depth = 30, criterion = entropy
	LR	C = 10
	KNN	n_neighbors = 6,weights = distance
	NB	var_smoothing = 1e-06
	XGB	gamma = 0.0, learning_rate = 0.01, max_depth = 3, n_estimators = 50
	RF	criterion = entropy, max_depth = 30, n_estimators = 200
All Features	SVM	C = 10, loss = hinge
	DT	max_depth = 20,criterion = entropy
	LR	C = 10
	KNN	n_neighbors = 6, weights = distance
	NB	var_smoothing = 0.01
	XGB	gamma = 0, learning_rate = 0.01, max_dept = 3, n_estimators = 50
	RF	criterion = entropy, max_depth = 40, n_estimators = 200

Table 5: Results from Grid Search in 9-Class Classification

Scenario	Model	Parameters
30 PCA Components	SVM	C = 10, loss = hinge
	DT	max_depth = 30, criterion = entropy
	LR	C = 100
	KNN	n_neighbors = 6,weights = distance
	NB	var_smoothing = 1e-07
	XGB	gamma = 0.25, learning_rate = 0.01, max_depth = 3, n_estimators = 50
	RF	criterion = entropy, max_depth = 40, n_estimators = 300
All PCA Components	SVM	C = 10, loss = hinge
	DT	max_depth = 30, criterion = entropy
	LR	C = 100
	KNN	n_neighbors = 6,weights = distance
	NB	var_smoothing = 0.001
	XGB	gamma = 0.0, learning_rate = 0.01, max_depth = 3, n_estimators = 50
	RF	criterion = entropy, max_depth = 40, n_estimators = 300
All Features	SVM	C = 10, loss = hinge
	DT	max_depth = 40,criterion = entropy
	LR	C = 100
	KNN	n_neighbors = 6, weights = distance
	NB	var_smoothing = 0.01
	XGB	gamma = 0, learning_rate = 0.01, max_dept = 3, n_estimators = 50
	RF	criterion = entropy, max_depth = 40, n_estimators = 200

Table 6: Results from Grid Search in 17-Class Classification

Scenario	Model	Parameters
30 PCA Components	SVM	C = 10, loss = squared_hinge
	DT	max_depth = 40, criterion = entropy
	LR	C = 1000
	KNN	n_neighbors = 6,weights = distance
	NB	var_smoothing = 1e-07
	XGB	gamma = 0.0, learning_rate = 0.01, max_depth = 3, n_estimators = 50
	RF	criterion = entropy, max_depth = 40, n_estimators = 300
All PCA Components	SVM	C = 10, loss = hinge
	DT	max_depth = 40, criterion = entropy
	LR	C = 1000
	KNN	n_neighbors = 6,weights = distance
	NB	var_smoothing = 0.001
	XGB	gamma = 0.0, learning_rate = 0.01, max_depth = 3, n_estimators = 50
	RF	criterion = entropy, max_depth = 40, n_estimators = 300
All Features	SVM	C = 10, loss = hinge
	DT	max_depth = 40,criterion = entropy
	LR	C = 1000
	KNN	n_neighbors = 6, weights = distance
	NB	var_smoothing = 0.01
	XGB	gamma = 0, learning_rate = 0.01, max_dept = 3, n_estimators = 50
	RF	criterion = entropy, max_depth = 40, n_estimators = 500

.5 Appendix 4

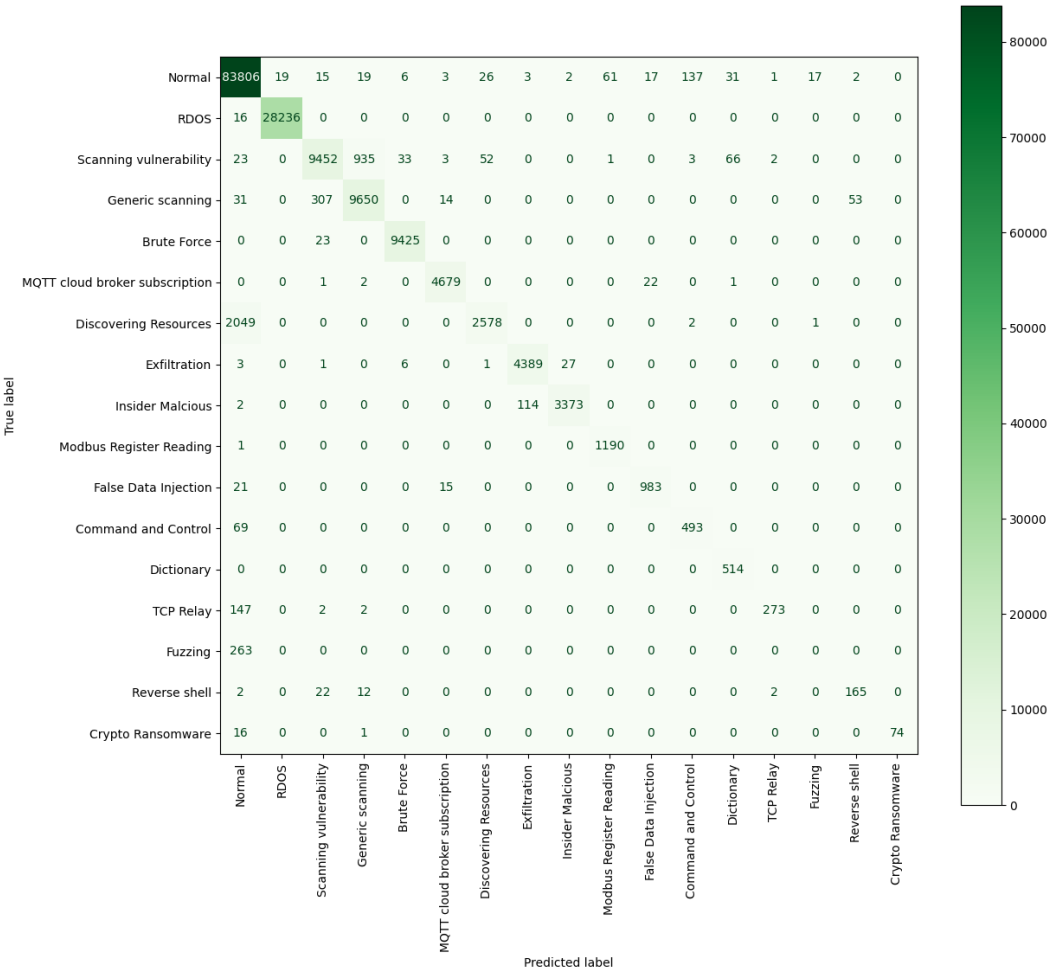


Figure 1: Confusion Matrix of 17 classes from SVM using all Components

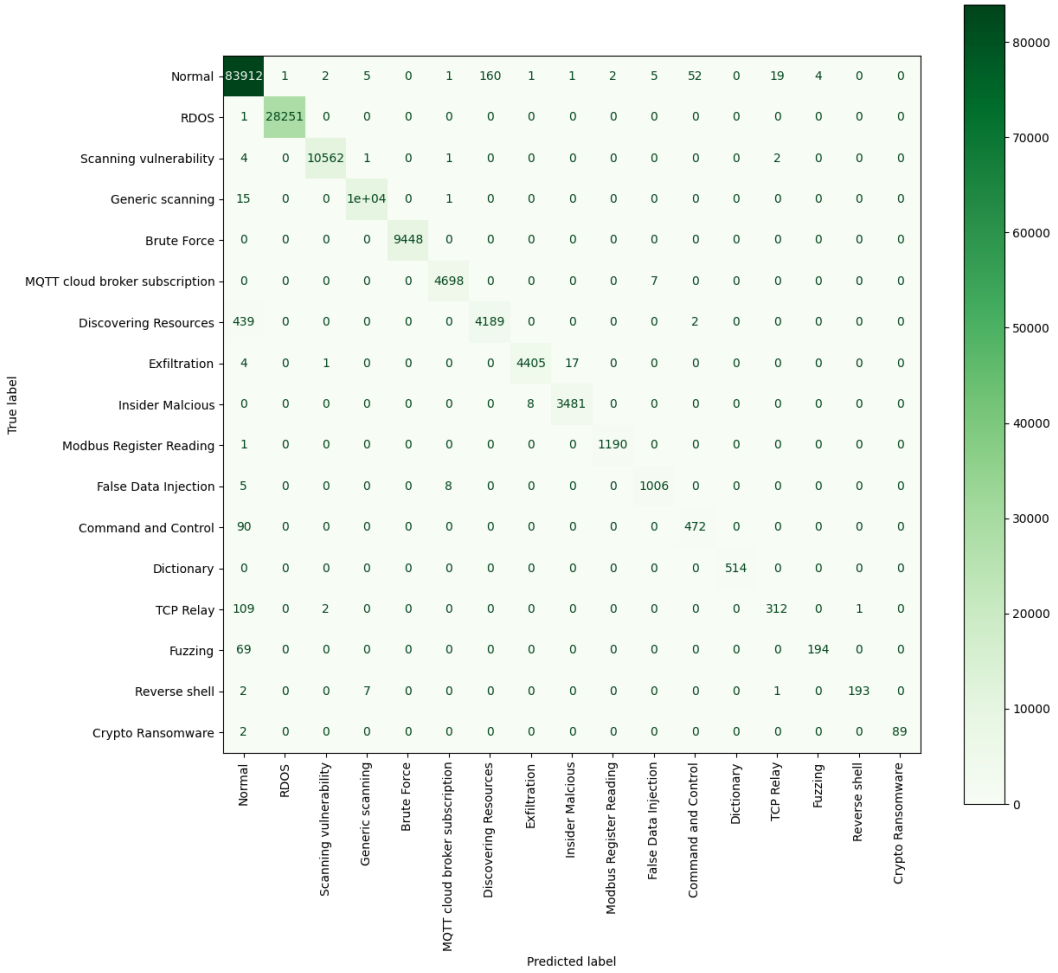


Figure 2: Confusion Matrix of 17 classes from XGB using all Components

.6 Appendix 5

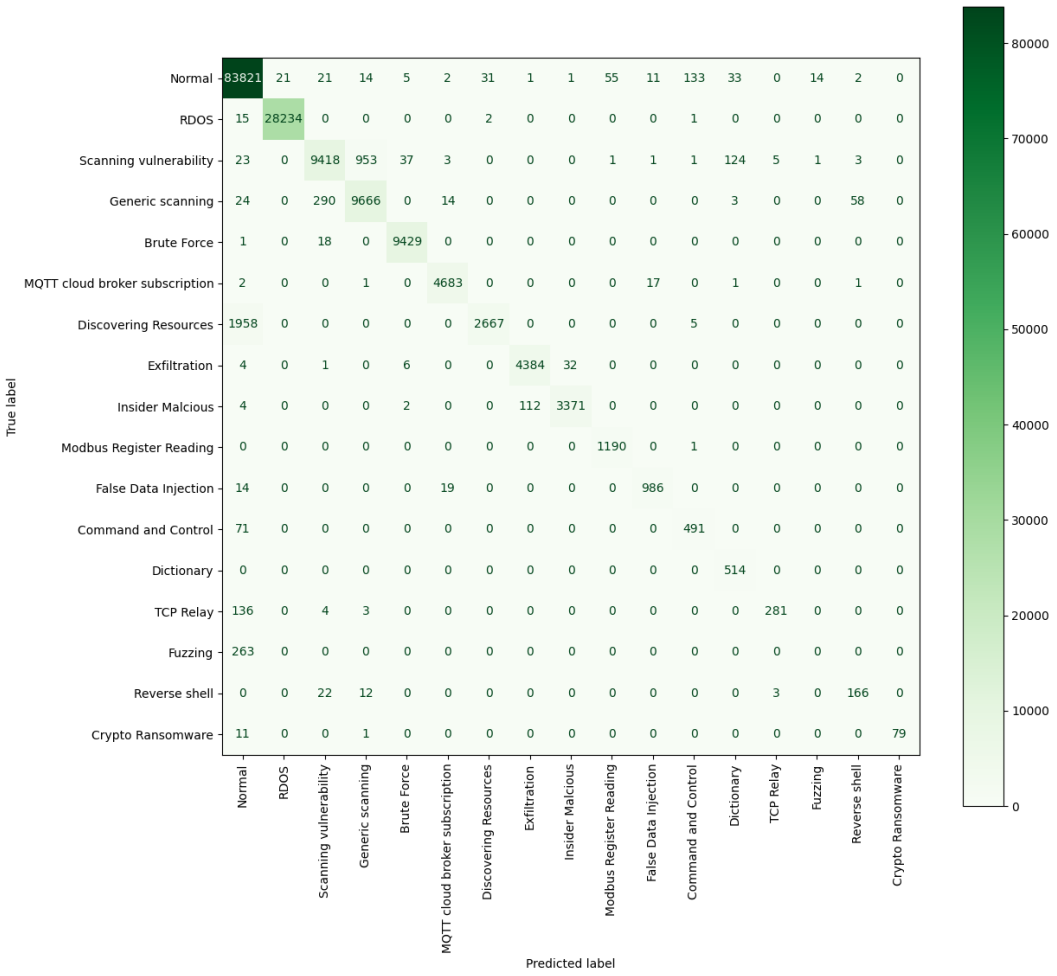


Figure 3: Confusion Matrix of 17 classes from SVM using all Features

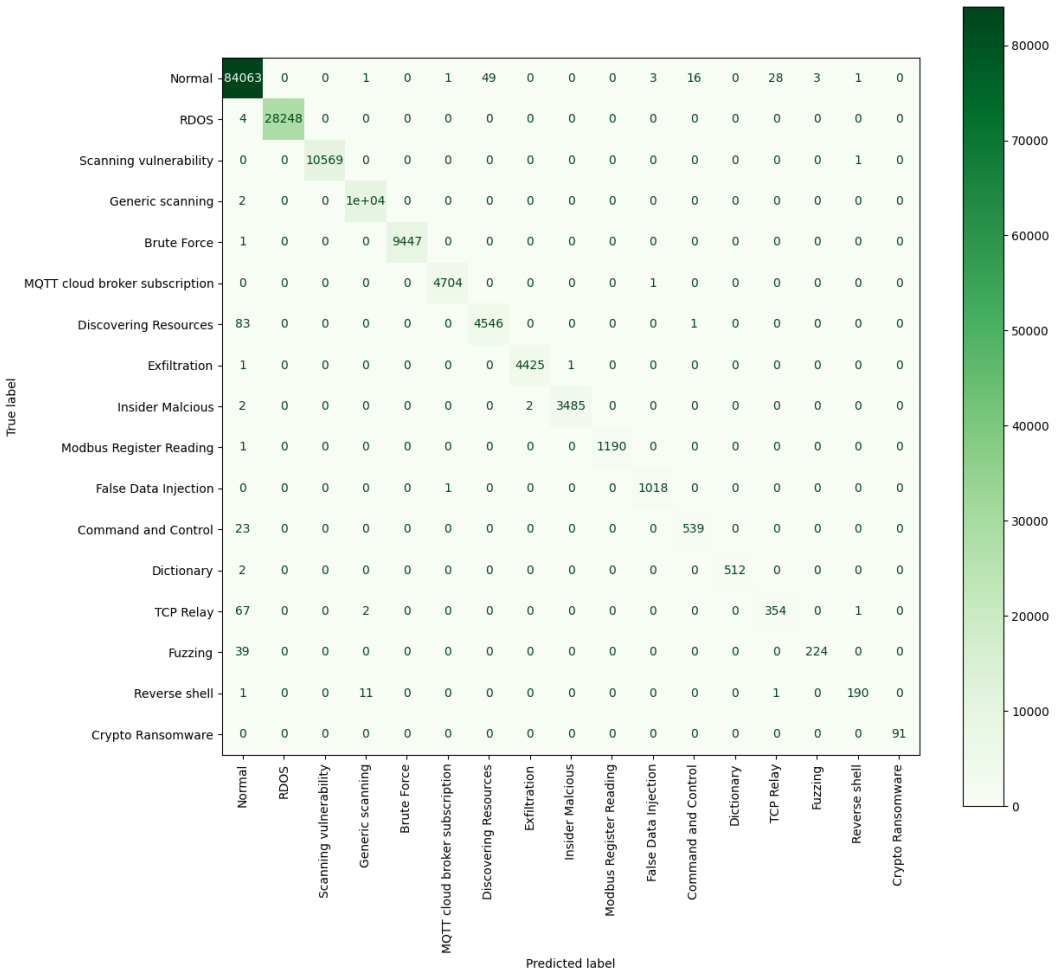


Figure 4: Confusion Matrix of 17 classes from XGB using all Features

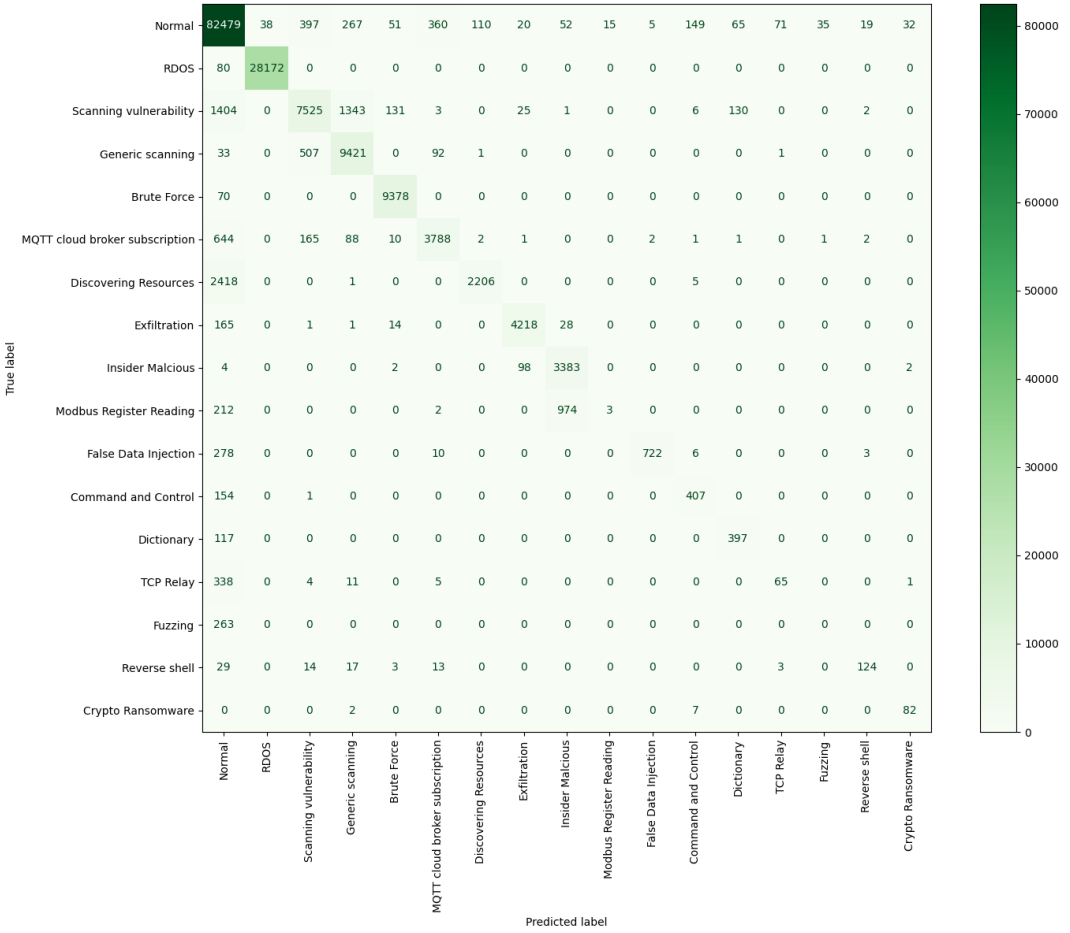


Figure 5: Confusion Matrix of 17 classes from SVM using 30 PCA Components

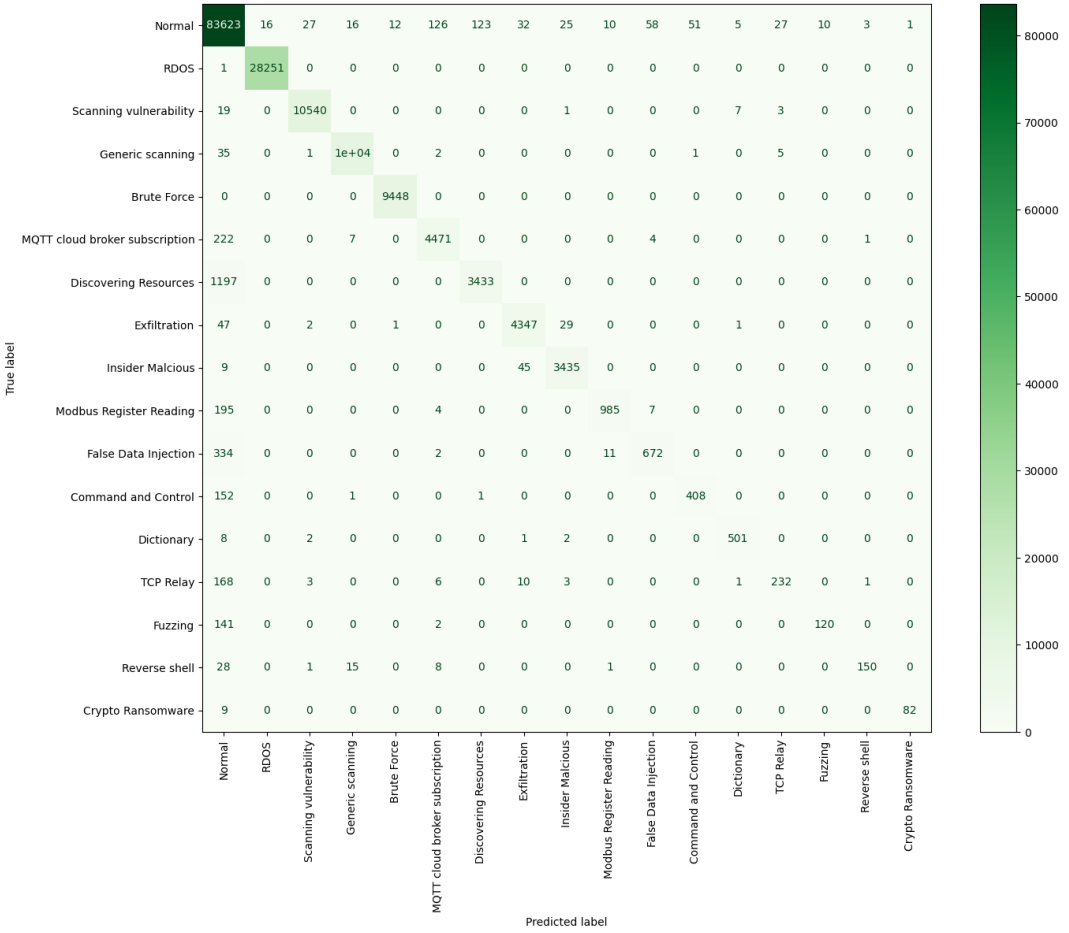


Figure 6: Confusion Matrix of 17 classes from XGB using 30 PCA Components

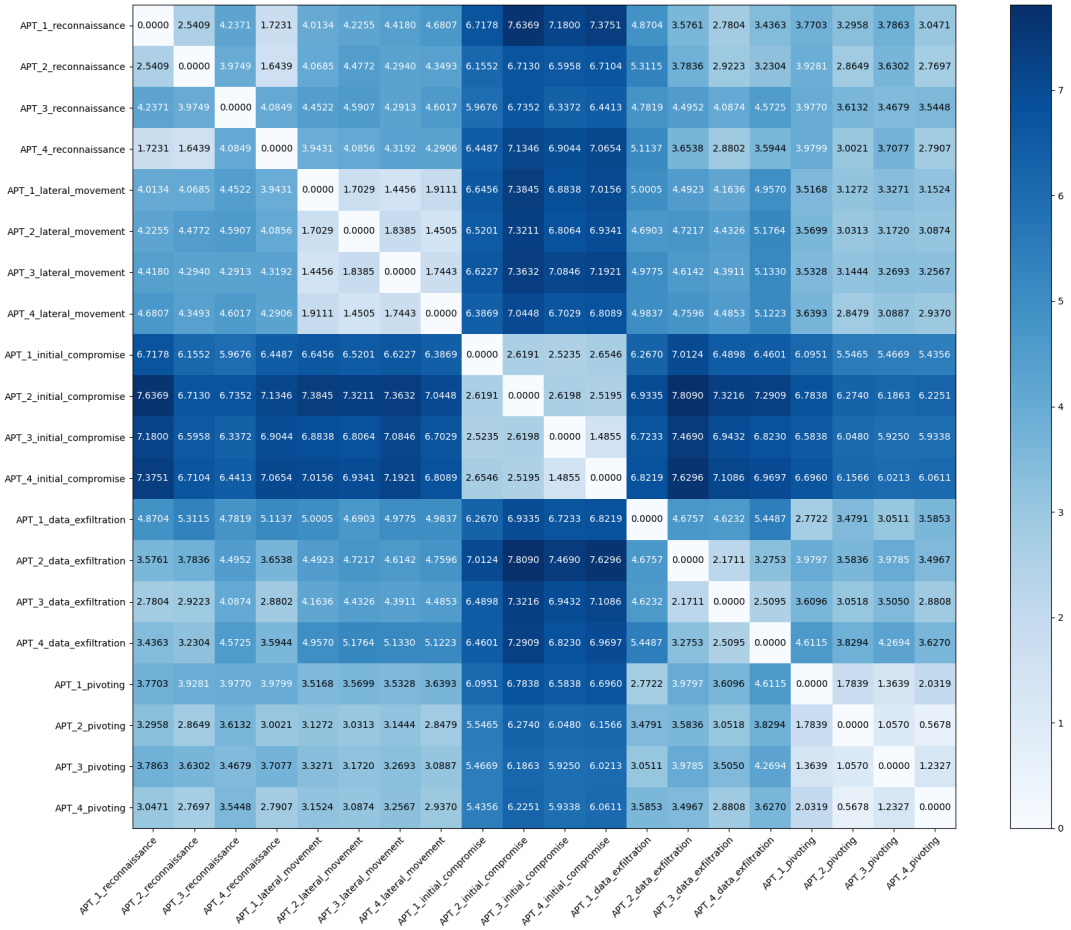


Figure 7: Centroid distance