

# **PERFORMANCE ANALYSIS OF MACHINE LEARNING MODELS FOR THE DETECTION OF CYBER THREATS AGAINST SATELLITE NETWORKS**

**A THESIS SUBMITTED TO THE  
DEPARTMENT OF SYSTEM ENGINEERING  
INSTITUTE OF SPACE SCIENCE AND ENGINEERING  
AN AFFILIATE OF  
AFRICAN UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF MASTER OF SCIENCE IN  
SYSTEM ENGINEERING**

**BY**

**ARAROMI HAONAT OLAJUMOKE  
(Reg, No. 41079)**



Institute of Space Science and Engineering

[www.isse.edu.ng](http://www.isse.edu.ng)

National Space Research and  
Development Agency,  
Airport Road Abuja, Nigeria



African University of Science and Technology

[www.aust.edu.ng](http://www.aust.edu.ng)

P.M.B. 681, Garki, Abuja  
F.C.T, Nigeria.

**AUGUST 2024**

## DECLARATION

I hereby certify that this thesis titled "PERFORMANCE ANALYSIS OF MACHINE LEARNING MODELS FOR THE DETECTION OF CYBER THREATS AGAINST SATELLITE NETWORKS" was carried out by me.



19/08/2024

.....  
ARAOMI HAONAT OLAJUMOKE

.....  
Date / Signature


## CERTIFICATION

This is to certify that the thesis titled “**Performance Analysis Of Machine Learning Models For The Detection Of Cyber Threats Against Satellite Networks**” submitted to the School of Postgraduate Studies, African University of Science and Technology (AUST), Abuja, Nigeria, for the award of the Master's degree is a record of original research carried out by **ARAOMI HAONAT OLAJUMOKE** in the **SYSTEMS ENGINEERING DEPARTMENT** of the Institute of Space Science and Engineering (ISSE), an affiliate of AUST.

Araromi Haonat Olajumoke

---

**Name**



---

**Sign**

19/08/2024

---

**Date**


**SIGNATURE**

**PERFORMANCE ANALYSIS OF MACHINE LEARNING MODELS FOR THE  
DETECTION OF CYBER THREATS AGAINST SATELLITE NETWORKS**

BY

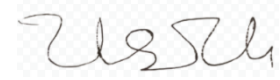
ARAOMI HAONAT OLAJUMOKE

A THESIS APPROVED BY THE DEPARTMENT OF SYSTEMS ENGINEERING

**RECOMMENDED:**  \_\_\_\_\_ 19/08/2024

Supervisor: Engr. Dr. Felix Ale

Date

  
\_\_\_\_\_

Co-Supervisor: Dr. Sanusi Muhammed

Date

19/08/2024

**Approved:**  \_\_\_\_\_ 19/08/2024

Head of Department: Engr. Dr. Felix Ale

Date

## ABSTRACT

The proliferation of satellite networks for various critical applications in the space sector has heightened the need for robust cybersecurity measures to safeguard these systems from malicious intrusions. An intrusion detection system serves as the backbone for providing high-level network security. Different network attacks have been discovered and are gradually becoming more sophisticated and complicated. Despite the availability of many existing intrusion detection systems, intuitive cybersecurity systems are needed due to alarmingly increasing intrusion attacks. Furthermore, with new intrusion attacks, the efficacy of existing systems is depleted unless they evolve. This study conducts experiments that compare three types of supervised machine learning algorithms, including Decision Tree (CART), SVM (Black-box), and KNN (Lazy learner). Thus, these different algorithms were compared using various evaluation metrics, which are accuracy, recall, false alarm rate, and precision, and manual feature selections were done to select important features from the dataset that increase relevance and reduce complexity along with the Training time complexity on three intrusion datasets (STIN, UNSW-NB15, and CIC-IDS2017(Wednesday)). CART DT achieves an accuracy of 93.42% with 31 features of the STIN dataset in 63.63 seconds, an accuracy of 93.13% with 8 features in 6.22 seconds, 76.63% with 42 features of the UNSW-NB15 dataset in 48.5 seconds, 76.63% with 6 features in 3.71 seconds, 99.87% with 68 features of the CIC-IDS 2017 Wednesday dataset in 59.29 seconds, 95.80% with 16 features in 4.96 seconds. SVM achieves an accuracy of 87.41% with 31 features of the STIN dataset in 559.51 seconds, an accuracy of 87.04% with 8 features in 286.78 seconds, 81.51% with 194 features of the UNSW-NB15 dataset in 421.06 seconds, 78.90% with 6 features in 176.05 seconds, 98.48% with 68 features of the CIC-IDS 2017 Wednesday dataset in 505.13 seconds, 96.92% with 16 features in 209.45 seconds. KNN achieves an accuracy of 86.28% with 194 features of the UNSW-NB15 dataset in 236.09 seconds. The results of this experiment give valuable insight for machine learning researchers into building a time-efficient and effective IDS using supervised machine learning for the Space sector. Although the secondary datasets used in this study provided good results, the use of primary datasets is suggested to enhance and improve the accuracy, integrity, and real-timeliness of the threat intelligence and resilience of the satellite networks in the space industry.

**Keywords:** Intrusion detection system (IDS), Satellite network, Machine learning, Cyber Security, Cyber Attacks, STIN, UNSW-NB15, and CIC-IDS2017 Dataset.

## **DEDICATION**

I dedicate this work to Allah, my family, friends, Staff, and colleagues at the Institute of Space Science and Engineering Abuja.

## ACKNOWLEDGMENT

Most importantly, I am very thankful to the Almighty Allah for His divine blessings before, during, and after the end of this research. I am immensely grateful to my project supervisor, Engr. Dr. Felix Ale, whose invaluable guidance, support, and motivation played a critical role in successfully completing this research. I also extend my heartfelt thanks to my co-supervisor, Dr. Sanusi Muhammed, for his insightful contributions and assistance throughout this project. My gratitude also goes to my provost, Engr. Dr. Seyi Festus Olatoyinbo, for his unwavering support and encouragement during this journey.

A special acknowledgement goes to Engr. Jide Salami, whose generous financial assistance made this academic endeavour possible. I am equally indebted to my brother, Sulaiman Araromi, for his crucial help with the thesis and to my friend, Abdulhammed Abdulsalam, whose expertise greatly contributed to the implementation phase of this research.

I deeply appreciate my parents, Prof and Mrs Araromi, and all my siblings, family, and friends for their constant kindness and support, without which this project could not have succeeded.

I also want to sincerely thank all the ISSE staff, especially Mrs. Rakiya Olojo and Engr. Dr. John Momoh, for their assistance in providing an enabling environment for this research. I am dearly obliged to my H.O.D. **Engr. Dr Felix Ale**, for his profound support and ISSE authority in allowing me to work on this thesis, has provided me with valuable information.

I appreciate you all, and God bless.

## TABLE OF CONTENTS

DECLARATION	ii
CERTIFICATION	iii
SIGNATURE	iv
ABSTRACT	v
DEDICATION	vi
ACKNOWLEDGMENT	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	xi
LIST OF FIGURES	xii
CHAPTER ONE	1
INTRODUCTION	1
1.1 BACKGROUND OF THE STUDY	1
1.2 STATEMENT OF THE PROBLEM	4
1.3 AIM OF THE STUDY	5
1.4 SPECIFIC OBJECTIVES OF THE STUDY	5
1.5 SCOPE OF THE STUDY	5
1.6 JUSTIFICATION OF THE STUDY	6
1.7 SIGNIFICANCE OF THE STUDY	7
1.8 MOTIVATION FOR THE STUDY	8
1.9 OPERATIONAL DEFINITION OF TERMS	8
1.10 CHAPTERS PREVIEW	9
CHAPTER TWO	11
LITERATURE REVIEW	11
2.1 SPACE	11
2.2 SATELLITE	12
2.2.1 ARCHITECTURE OF SATELLITE	14
2.2.2 SATELLITE SECURITY	16
2.2.3 VULNERABILITIES OF SATELLITE NETWORK	17
2.2.4 CYBER THREATS AGAINST SATELLITE NETWORKS	18
2.3 INTRUSION DETECTION SYSTEM.	23
2.4 MACHINE LEARNING.	26



2.4.1 MACHINE LEARNING FOR INTRUSION DETECTION.	27
2.4.2 CHALLENGES IN MACHINE LEARNING FOR INTRUSION DETECTION	37
2.5 FEATURE SELECTION	38
2.5.1 METHODS OF FEATURE SELECTION	40
2.6 SPACE-SPECIFIC FEATURES TO IDENTIFY CYBER THREATS IN SATELLITE NETWORKS	42
2.7 RELATED WORKS	43
CHAPTER THREE	49
MATERIALS AND METHODS	49
3.1 BENCHMARKING DATASETS.	49
3.1.1 STIN DATASET	49
3.1.2 UNSW-NB15 DATASET	50
3.1.3 CICIDS2017 DATASET	51
3.2 CLASSIFIER COMPLEXITY.	52
3.2.1 THEORETICAL ANALYSIS OF DECISION TREE.	52
3.2.2 THEORETICAL ANALYSIS OF KNN	53
3.2.3 THEORETICAL ANALYSIS OF SVM	54
3.3 DATA PREPROCESSING	54
3.3.1 DATA CLEANING	56
3.3.2 MINORITY REMOVAL	56
3.3.3 ENCODING	57
3.3.4 NORMALIZATION	57
3.4 TRAINING AND TESTING SET PREPARATION	58
3.5 FEATURE SELECTION	59
3.6 EXPERIMENTAL PROCEDURE.	60
3.7 MODEL EVALUATION	61
CHAPTER FOUR	63
RESULTS AND DISCUSSION	63
4.1 PERFORMANCE METRICS	64
4.2 PERFORMANCE ANALYSIS	65
4.3 COMPARISON OF CART DT WITH OTHER CLASSIFIERS.	73
4.4. COMPUTATIONAL COMPLEXITY OF THE ALGORITHMS	79
4.5 DISCUSSION OF RESULTS	81

4.6 OPERATIONAL PROCEDURE OF THE CODES	82
4.6.1 PREREQUISITES	82
4.6.2 SETTING UP THE PROJECT	83
4.6.3 RUNNING THE CODES	83
4.6.4 INTERPRETING THE RESULTS	84
CHAPTER FIVE	85
SUMMARY OF FINDINGS, CONCLUSION AND RECOMMENDATIONS	85
5.1 SUMMARY OF FINDINGS	85
5.2 RESEARCH LIMITATIONS	86
5.3 CONCLUSION	86
5.4 RECOMMENDATIONS	87
5.5 SUGGESTIONS FOR FUTURE RESEARCH	88
APPENDIX A: DATA PREPROCESSING	89
APPENDIX B: DECISION TREE (CART) MODEL IMPLEMENTATION	92
APPENDIX C: SVM MODEL IMPLEMENTATION	93
APPENDIX D: KNN MODEL IMPLEMENTATION	94
APPENDIX E: EVALUATION METRICS IMPLEMENTATION	95
REFERENCES	97

## LIST OF TABLES

Table 1: Comparative analysis of the existing approaches.	50
Table 2: Details of STIN dataset.	50
Table 3: Record distribution of UNSW-NB15 Dataset	51
Table 4: Record distribution of CICIDS2017 Wednesday Dataset	52
Table 5: Accuracy of Decision Tree classifier on STIN full feature set dataset.	74
Table 6: Accuracy of Decision Tree classifier on STIN reduced feature set dataset.	75
Table 7: Accuracy of SVM classifier on STIN full feature set dataset.	75
Table 8: Accuracy of SVM classifier on STIN reduced feature set dataset.	75
Table 9: Accuracy of classifiers on CIC-IDS 2017 Wednesday full feature set dataset.	76
Table 10: Accuracy of classifiers on CIC-IDS 2017 Wednesday reduced feature set dataset.	76
Table 11: Accuracy of classifiers on UNSW-NB15 full feature set dataset.	76
Table 12: Accuracy of classifiers on UNSW-NB15 reduced feature set dataset.	76
Table 13: Estimated execution time of all classifiers on all full features of the three datasets.	80
Table 14: Estimated execution time of all classifiers on the reduced features set of the three datasets.	80

## LIST OF FIGURES

Figure 1: Space segments (Thangavel et al., 2022)	12
Figure 2: Satellite-terrestrial communication networks (Nguyen & Chang, 2019)	13
Figure 3: A satellite system architecture (Al-Hraishawi et al., 2023)	16
Figure 4: Cyber threats identified by NASIC.	19
Figure 5: Categories of intrusion detection systems (IDS) (Eshakagdy et al., 2022)	26
Figure 6: KNN (Tahri et al., 2022)	29
Figure 7: Decision Trees (Dey, 2016)	31
Figure 8: Naïve Bayes (Tahri et al., 2022)	32
Figure 9: SVM (Tahri et al., 2022)	33
Figure 10: Reinforcement Learning Model (Dey, 2016)	37
Figure 11: Experimental Flowchart.	61
Figure 12: Confusion matrix of a full feature set for Decision Trees (STIN)	66
Figure 13: Confusion matrix of a Reduced feature set for Decision Trees (STIN)	67
Figure 14: Confusion matrix of a Full feature set for SVM (STIN)	67
Figure 15: Confusion matrix of a Reduced feature set for SVM (STIN)	67
Figure 16: Confusion matrix of a Full feature set for Decision Tree (CART) for UNSWB-15	68
Figure 17: Confusion matrix of a Reduced feature set for Decision Tree (CART) for UNSWB-1	69
Figure 18: Confusion matrix of a Full feature set for SVM for UNSWB-15	69
Figure 19: Confusion matrix of a Reduced feature set for SVM for UNSWB-15	70
Figure 20: Confusion matrix of a Full feature set for KNN for UNSWB-15	70

Figure 21: Confusion matrix of a Full feature set for Decision Tree (CART) for CICIDS2017-Wednesday	71
Figure 22: Confusion matrix of a Reduced feature set for Decision Tree (CART) for CICIDS2017-Wednesday	72
Figure 23: Confusion matrix of a Full feature set for SVM for CICIDS2017-Wednesday	72
Figure 24: Confusion matrix of a Reduced feature set for SVM for CICIDS2017-Wednesday	73
Figure 25: Performance comparison of all models on the STIN Full feature dataset	77
Figure 26: Performance comparison of all models on the STIN Reduced feature dataset	77
Figure 27: Performance comparison of all models on the CIC-IDS 2017 (Wednesday) Full feature dataset	78
Figure 28: Performance comparison of all models on the CIC-IDS 2017 (Wednesday) Reduced feature dataset	78
Figure 29: Performance comparison of all models on the UNSWB-15 Full feature dataset	79
Figure 30: Performance comparison of all models on the UNSWB-15 Reduced feature dataset	79

## CHAPTER ONE

### INTRODUCTION

#### 1.1 BACKGROUND OF THE STUDY

In today's interconnected world, a fast-proliferating network of satellites forms the critical infrastructure that supports global communication, navigation, weather forecasting, defensive operations, and more (Sylvester, 2024). Today's global space economy is enormous, forecasted to total more than \$600 billion annually in 2024 (Paul, 2024). Satellites orbiting Earth are pivotal for everything from GPS navigation to international banking transactions, making them indispensable assets in our daily lives and global infrastructure. The industry has experienced a wave of commercialisation, with numerous start-up companies emerging and attracting private investments. Public market interest has also enabled leading space start-ups to access large amounts of capital (Jora et al., 2023).

The space industry has been growing rapidly in recent years in terms of revenues and launches. According to a report by the Space Foundation, the space economy was worth \$469 billion in 2021, a 9% increase from 2020 (Ellerbeck, 2022). The report says over 1,000 spacecraft were put into orbit in the first six months of 2021—more than were launched in the first 52 years of space exploration (Ellerbeck, 2022).

It is crucial to recognise that satellites are more vulnerable than commonly perceived. According to the National Institute of Standards and Technology (NIST), a cyber-attack is a serious threat. It is defined as an attack conducted via cyberspace that targets an enterprise's use of cyberspace to disrupt, disable, destroy, or maliciously control a computing environment/infrastructure to compromise the integrity of the data or steal controlled information (Aerospace Corporation,

2022). Cyber-attacks are particularly attractive to adversaries in conflict situations. The boundary is often considered the communications link for satellites, i.e., the radio frequency link or the ground system. If this boundary is breached, there is little internal protection within the satellite, allowing an adversary to operate freely inside the system — reminiscent of the early days of traditional cybersecurity when border firewalls were the sole protection from intrusion.

Cyber-attacks have traditionally been associated with ransomware, wherein hackers attempt to breach records, releasing them only after ransom payment. In order to give an idea of the impact of these kinds of attacks, an official UK source registered eighty-three data breaches in February 2022 alone, with over five million records at risk (ITGovernance, 2022). Just before the conflict in Ukraine, an increased number of such attacks were registered targeting Ukrainian banks and government institutions in the second half of February 2022. Several countries are looking into counter-space capabilities that include electronic methods. Compared to anti-satellite (ASAT) capabilities, interference with a satellite through a cyber-attack can be conducted in a way that is cheaper, faster, and more difficult to trace (Rajagopalan, 2019). Cyber-attacks on satellites often relate to accessing the satellite system via ground stations. Several attempts, often considered by cyber-experts as experimental tests and preparatory, are known but not widely reported by satellite operators for obvious commercial reasons.

In addition, compromised or by-design malicious satellites can be orchestrated to target benign satellites, taking advantage of the limited security measures deployed onboard for inter-satellite network communication. With regards to the continued operation of a satellite, (Distributed) Denial of Service (DDoS) attacks (Greenberg, 2022) can be caused by increased network interactions, exploitation of communication protocol weaknesses, or tampering with the Operating System (OS)/Firmware, resulting in a non-functional state known as bricks. Power depletion is

another prominent attack that can affect smart satellites, as it is a constrained resource that can be affected by unnecessarily increasing the workload of the satellite and its sensors, which can be achieved either through dedicated cyber-attacks or as a by-product (consequence) of DoS and DDoS attacks. Furthermore, as satellites transmit data between ground stations, eavesdropping attacks such as Man-In-The-Middle (MITM) and data manipulation attacks can impact the confidentiality and integrity of smart satellites.

The space sector has increasingly relied on computer systems and networks to operate and control their spacecraft and satellites. As a result, cyberattacks on the space industry have become a growing concern, as they can cause significant damage to equipment and infrastructure and compromise sensitive data and information (Jordan & Mitchell, 2015). Therefore, there is a need for effective cybersecurity measures to prevent and mitigate the impact of cyberattacks. Cybersecurity threats to the space industry come from various sources, including nation-states, criminal organizations, and individual hackers. Cyberattacks can take many forms, such as malware, phishing, social engineering, and denial-of-service (DoS) attacks (Jordan & Mitchell, 2015). Moreover, the consequences of cyberattacks can be severe, ranging from data breaches to system shutdowns, physical destruction, and even endangering human life in space.

Traditional security methods, such as firewalls and intrusion detection systems, are often inadequate to address the evolving threats posed by cybercriminals and nation-states. These attackers constantly develop new techniques and exploits, making it difficult for traditional security measures to keep pace (Awuor, 2023; Muhammad et al., 2022). Also, intrusion detection is important to ensure space-based wireless network security. In recent years, with advances in artificial intelligence (AI), intrusion detection methods using AI have been gradually proposed. Usually, these methods involve significant computing, communications, and storage resources.



Machine learning (ML) offers a promising approach for improving cybersecurity. ML algorithms can learn from large amounts of data to identify patterns and anomalies that may indicate an impending cyberattack. This allows for more proactive and effective security measures (*Machine learning (ML) in cybersecurity*, 2023; Apruzzese et al., 2023). This study addresses this need by evaluating the machine-learning model for predicting satellite network cyberattacks.

Murphy (2012) discussed the probabilistic perspective of machine learning, which involves modelling uncertainty in data and making predictions based on statistical inference. This perspective benefits cybersecurity, where cyberattacks are highly uncertain and unpredictable. By modelling uncertainty and probability, machine learning models can make more accurate predictions and improve the overall effectiveness of cybersecurity measures.

One of the challenges in machine learning is bias and fairness. Mehrabi et al. (2022) discussed the issue of bias in machine learning and its potential impact on cybersecurity. Bias can occur when machine learning models are trained on biased data, resulting in inaccurate predictions and inappropriate decision-making. Therefore, it is crucial to ensure that machine learning models are trained on unbiased and representative data to avoid bias and improve the fairness of the prediction.

## **1.2 STATEMENT OF THE PROBLEM**

It is well known that the space sector is critical to the national economy, even at the global level; therefore, any cyberattacks against the space industry will cause a colossal waste of economic resources and potential data breaches to severe disruptions of satellite communications and compromise of critical infrastructure which can lead to economic loss, loss of property, and even loss of lives.

However, efforts have been continuously made to protect the satellite network using traditional and artificial intelligence techniques. Due to the prevailing and continuous incidence of attacks, there have been limited innovative approaches to combat security, especially in developing countries. Therefore, this research seeks to address the gap in current cybersecurity practices within the satellite system by evaluating intelligent intrusion detection system models capable of predicting and preemptively mitigating cyberattacks.

### **1.3 AIM OF THE STUDY**

To evaluate machine learning models for intelligent intrusion detection systems to predict cyberattacks on satellite networks.

### **1.4 SPECIFIC OBJECTIVES OF THE STUDY**

1. Acquire cyberattack datasets from the STIN, UNSWB15, and CICIDS2017 platforms to train and evaluate network intrusion detection.
2. Analyze feature selection to identify relevant features that capture the characteristics of network traffic attacks.
3. Train the datasets by evaluating and comparing the performance of Decision Trees (CART), K-nearest neighbours (KNN), and Support Vector Machines (SVM) to predict cyberattacks on satellite network systems.
4. To compare the time complexity of the full feature set and the reduced feature selection set of the Decision Tree (CART), KNN, and SVM models.

### **1.5 SCOPE OF THE STUDY**

This project evaluates machine learning models for detecting cyberattacks on satellite networks. It will utilize three established network traffic datasets (STIN, UNSWB15, and CICIDS2017) for training and testing. The project will encompass the following key areas:

- Data acquisition and preprocessing from the chosen datasets.
- Implement and train three machine learning models (Decision Tree, KNN, SVM) for intrusion detection.
- Evaluation of model performance using relevant metrics.
- Comparison and analysis of the model's strengths and weaknesses.
- Discussion of deployment considerations for satellite network.

This thesis excludes real-world network traffic capture and analysis from satellite systems due to potential security concerns and data access limitations.

## **1.6 JUSTIFICATION OF THE STUDY**

The space sector is a vital and growing sector of the global economy and a strategic domain for national security and scientific exploration. However, the satellite systems enabling these activities are increasingly vulnerable to cyberattacks, which can compromise their functionality, integrity, and availability. Cyberattacks on satellite systems can have severe consequences, such as disrupting critical services, damaging expensive assets, endangering human lives, and escalating conflicts.

Therefore, effective and efficient methods for predicting cyberattacks on satellite networks are needed so that appropriate countermeasures can be taken in advance to prevent or mitigate potential damage. Machine learning is a promising technique for cyberattack prediction, as it can learn from data and detect patterns and anomalies that indicate malicious behaviour.

The primary significance of this study is that it will provide an effective method for predicting cyberattacks in the space industry. Over the years, the space sector has suffered significant losses due to cyberattacks. For instance, in 2018, NASA reported a data breach that compromised social

security numbers, among other sensitive information. With machine learning, it is possible to analyze massive amounts of data to detect and predict cyberattacks before they occur. The work of Abaimov and Martellini (2022) highlights the importance of understanding machine learning in cybersecurity.

It will also provide insight into the best machine-learning algorithms for predicting cyberattacks in the space industry. The study conducted by Sarker (2021) on machine learning algorithms, real-world applications, and research directions shows that machine learning algorithms are diverse and can be used for different purposes. The appropriate machine learning algorithm selection is vital for developing a predictive model for cyberattacks in the space sector.

## **1.7 SIGNIFICANCE OF THE STUDY**

However, significant cybersecurity implications must be considered as the space sector expands. The increasing reliance on satellites for critical services, such as communication and navigation systems, emphasizes the importance of ensuring their security (Hennecken, 2020). Satellites are vulnerable to cyber-attacks that can disrupt essential services and compromise their control, potentially leading to malfunctions or becoming space debris hazards. Therefore, prioritizing cybersecurity measures, including robust protocols and standards, is crucial to protect critical infrastructure, maintain service continuity, and prevent cyber-attacks (Bailey, 2020). Intrusion detection plays a significant role in space cybersecurity, helping to identify unexpected events or behaviours that may indicate equipment failures, malicious attacks, or environmental factors (Falco & Boschetti, 2021).

## **1.8 MOTIVATION FOR THE STUDY**

The rapid commercialization of space and increased ease and lower costs associated with launching satellites into space have resulted in global supply chains of privatized satellite networks for commercial and military purposes. It is an industry that extends to every sector of the economy and can contribute to sustainable development in regional areas (Bruckardt, 2022).

The requirement for robust space security is stringent and unique because space (Pavlov & Martinovic, 2019) 1) is a single point of failure for several industry sectors, increasing the number and capabilities of attackers 2) involves a complex supply chain and prolonged system lifecycle, rising malware backdoors 3) employs commercial off-the-shelf technology, bringing vulnerabilities from many platforms and 4) is a resource-constrained environment, seeking lightweight solutions more than terrestrial systems. This makes space security a unique challenge and an attractive target for cybercriminals.

Intrusion detection has become increasingly crucial in satellites as cyber-attacks targeting space systems and infrastructures have grown in number and sophistication (Pearson, 2022). With the growing dependency of critical infrastructures, including communication, navigation, and surveillance, on space-based assets, the security of space systems directly impacts national security. The understanding and knowledge of space security, particularly in intrusion detection systems, are currently limited.

## **1.9 OPERATIONAL DEFINITION OF TERMS**

1. Intrusion Detection System (IDS): This security technology detects unauthorized access or attacks on a network by monitoring and analyzing network traffic for suspicious activities and known attack patterns.

2. Machine Learning (ML) is a subset of artificial intelligence that enables systems to learn from data, identify patterns, and make decisions with minimal human intervention. This study uses ML models to predict cyberattacks on satellite networks.
3. Cyberattacks: Deliberate attempts by individuals or organizations to breach the information systems of another individual or organization. These attacks can aim to steal, alter, or destroy data, disrupt operations, or gain unauthorized access to computing resources. Examples include malware, phishing, denial-of-service (DoS) attacks, and advanced persistent threats (APTs).
4. Satellite: An artificial object placed into orbit around the Earth or another celestial body to perform specific functions, such as communication, weather monitoring, navigation, and scientific observation. Satellites have various instruments and communication systems to gather and transmit data to Earth.
5. Satellite Networks: Specialized networks that use satellites to establish communication links between different geographic locations on Earth. These networks are crucial for global communications, broadcasting, navigation, and remote sensing. Satellite networks involve ground stations that transmit and receive data to and from the satellites, as well as the satellites that relay the data across vast distances.
6. Feature Selection selects a subset of relevant features for model construction. It helps improve model accuracy and reduce overfitting.

## **1.10 CHAPTERS PREVIEW**

The rest of this thesis is organized as follows: Chapter 2, which is a literature review, reviews existing research on intrusion detection systems, machine learning models, and their applications in cybersecurity, particularly in the context of satellite networks. It discusses the strengths and limitations of various approaches and identifies gaps that this study aims to address. Chapter 3, the

materials and methods, outlines this study's research design and procedures. It details the data collection process, pre-processing techniques, and the machine learning models employed. The chapter also describes the evaluation metrics used to assess model performance. Chapter 4, which is the results and discussion, presents the findings from evaluating the machine learning models. It includes a detailed analysis of each model's performance and a comparative analysis to determine the most suitable model for deployment in satellite network IDS. Chapter 5, which is the conclusion and recommendation, summarizes the research's key findings and discusses their implications for satellite network security. It offers recommendations for practitioners and outlines directions for future research, focusing on improving IDS effectiveness and adaptability to emerging cyber threats.

## CHAPTER TWO

### LITERATURE REVIEW

#### 2.1 SPACE

SPACE, a concept that has intrigued humanity for centuries, represents not only the vast unknown regions beyond our planet but also the potential for groundbreaking discoveries and advancements in science and technology (Wang, 2023). Exploring space has always captured the imagination of people worldwide, pushing the boundaries of what we know about the universe and our place within it. From exploring distant galaxies to understanding the mysteries of black holes and dark matter, the study of space continues to reveal new wonders and challenges that inspire generations of scientists and explorers alike (Szolucha, 2022).

The quest to unravel the secrets of the cosmos drives innovation and collaboration among nations, fostering a shared vision of expanding our knowledge and capabilities beyond Earth's borders. As we look up at the night sky, contemplating the infinite possibilities that space holds, we are reminded of the boundless opportunities for discovery and growth ahead (Dubosq et al., 2022). Satellites are one of the leading products and services in this sector, and the terms "space" and "satellite" are sometimes used interchangeably (Jora et al., 2023).



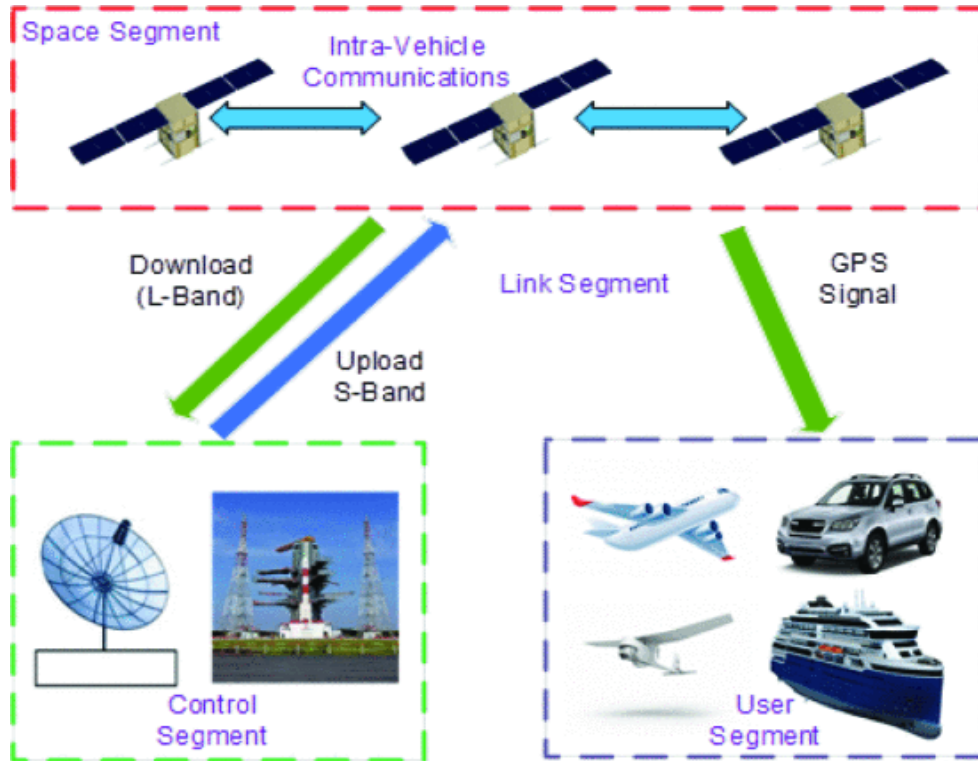


Figure 1: Space segments (Thangavel et al., 2022)

## 2.2 SATELLITE

A satellite is a self-contained communications device that can receive messages from Earth and retransmit them using a transponder, which functions as both a radio transmitter and receiver. According to the National Aeronautics and Space Administration (NASA), any natural space body or machine that orbits around a planet or a star is a satellite (What Is a Satellite, 2021). Thanks to the satellite, large areas of Earth may be seen at once. As a result, satellites can gather data more quickly and efficiently than devices on the ground, as shown in Fig. 2. The ability of satellites to transmit signals from one place to numerous destinations is their fundamental benefit. As a result, “point-to-multipoint” communications like broadcasting are perfect for satellite technology (Rath & Mishra, 2020). Satellite communication is the best option for underserved and remote locations with dispersed populations because it does not require significant investments on the ground. A satellite node that an attacker has targeted becomes quickly exhausted and is challenging to repair.

Therefore, high-level protection for modern networks requires developing effective intrusion detection techniques. To address these issues, further classification algorithms are presented. Therefore, high-level protection for modern networks necessitates the development of effective intrusion detection techniques due to the rise in network intrusion attacks (Li et al., 2020).

To form a complete space system, the satellite(s) needs to be paired with a ground component, which is “a set of geographically distributed stations with powerful satellite communications (SATCOM) equipment that can send command and control telemetry to satellites and receive telemetry data from the satellite’s systems and instruments” (Hutchins, 2016). We usually refer to instructions sent to the satellite as telecommands and to collected data as telemetry.

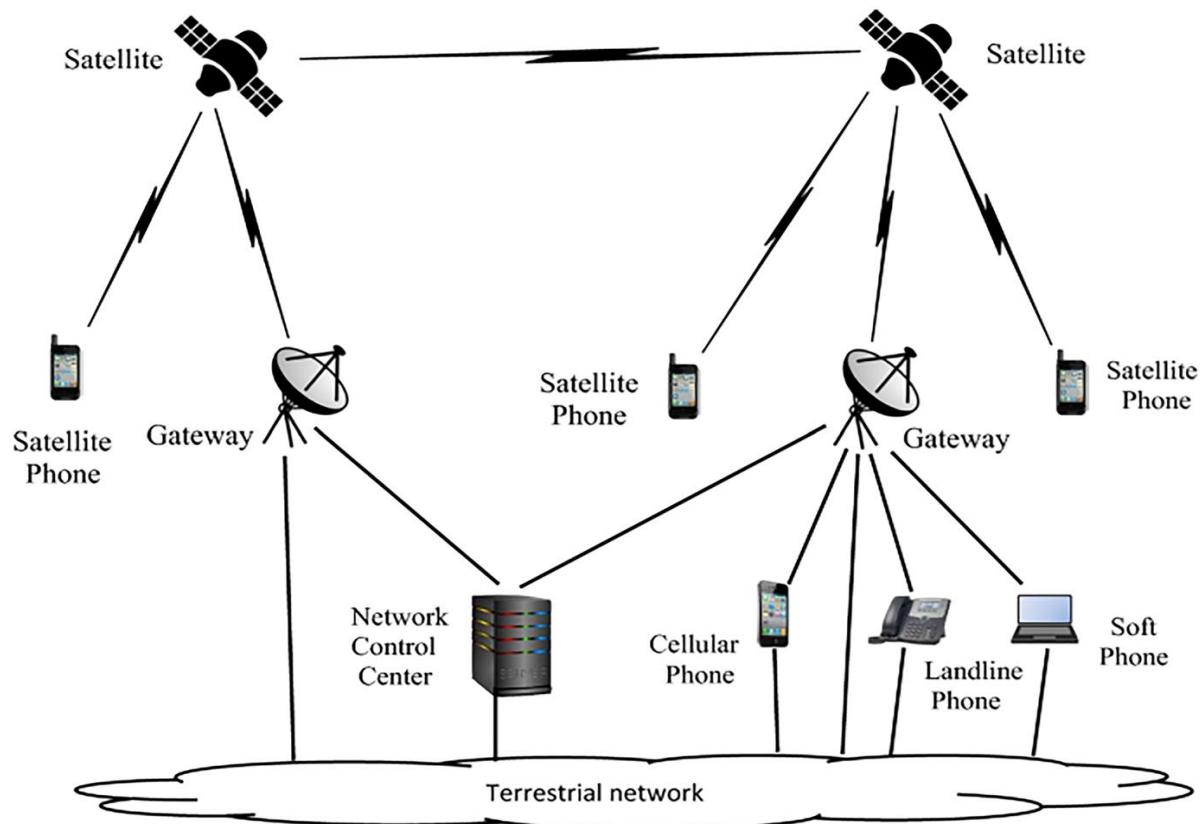


Figure 2: Satellite-terrestrial communication networks (Nguyen & Chang, 2019)

Depending on the size and equipment present on board, the intended functionality of the satellite can include a wide variety of actions: for example, they can be used in communication to relay audio or video signals, they can collect information about Earth's surface (such as pictures or meteorological information), or the military can use them to survey areas of interest. Satellites can be classified either by their functionality (communication, Earth observation, Global Navigation Satellite Systems - GNSS, etc.), their distance from the Earth's surface (Low Earth Orbit - between 160 and 1000 km, Middle Earth Orbit - up to 35786 km, High Earth Orbit - above 35786 km and Geostationary Orbit - at exactly 35786 km) or by their size (NanoSats – up to 10 kg, SmallSats - up to 500 kg etc.) (Types of Satellites and Applications, 2021).

However, the satellite can be divided into two major parts: the platform and the payload. The platform is a mostly standardized part of the satellite that provides the structural foundation and equipment necessary for the satellite to endure outer space. The payload is usually mission-specific and highly specialized, providing the satellite's logical and functional capabilities to achieve its intended goal.

### **2.2.1 Architecture of Satellite**

The space segment of a satellite architecture is one of the three main components of a satellite system. This segment comprises the satellites and their associated subsystems, such as power, propulsion, attitude control, payload, and telemetry. The satellites are launched into specific orbits depending on their intended function and coverage area. For example, GEO satellites support business in navigation, data, mobile television, and radio broadcasting systems. At the same time, MEO satellites are deployed to deliver low-latency and high-bandwidth data connectivity to service providers, agencies and industries and to support the network connectivity in the avionic/maritime domain. LEO satellite constellations have also been adopted for imaging, low-

bandwidth telecommunications, and broadband internet applications. Each of these satellites is placed in orbit by a launch vehicle. The space segment also includes military and defence communication systems and commercial SATCOM transponders and payloads. Note that the aforementioned communication links involving satellites use frequencies in the L-band, in the range [1 - 2] GHz (Tedeschi et al., 2022).

The ground segment comprises the terrestrial stations communicating with the satellites and providing control, monitoring, and data processing functions. The ground segment includes gateways: satellite operators, network operations centres, tracking stations, telemetry stations, and command (TT&C) stations. Gateways are the stations that connect the satellite network to other networks, such as the Internet, cellular, or terrestrial. Network operations centres are the facilities that manage the satellite network's overall performance, configuration, and maintenance. TT&C stations send and receive commands and data to and from the satellites to ensure their proper functioning and orbit (Tedeschi et al., 2022).

The user segment includes the user terminals, such as satellite mobile phones, ships, and aeroplanes, to name a few. These devices can communicate with satellites by leveraging the link between the ground and user segments, such as the forward link (Abe et al., 2018). At the same time, their communication with the gateways can take place over any communication technology. The forward link consists of an uplink (base station to satellite) and a downlink (satellite to mobile user). Conversely, constellations like Iridium, Globalstar, Thuraya and Inmarsat allow a direct connection of the user handsets to the satellites, using the User to Satellite (US) link that typically uses frequencies in the L-band.

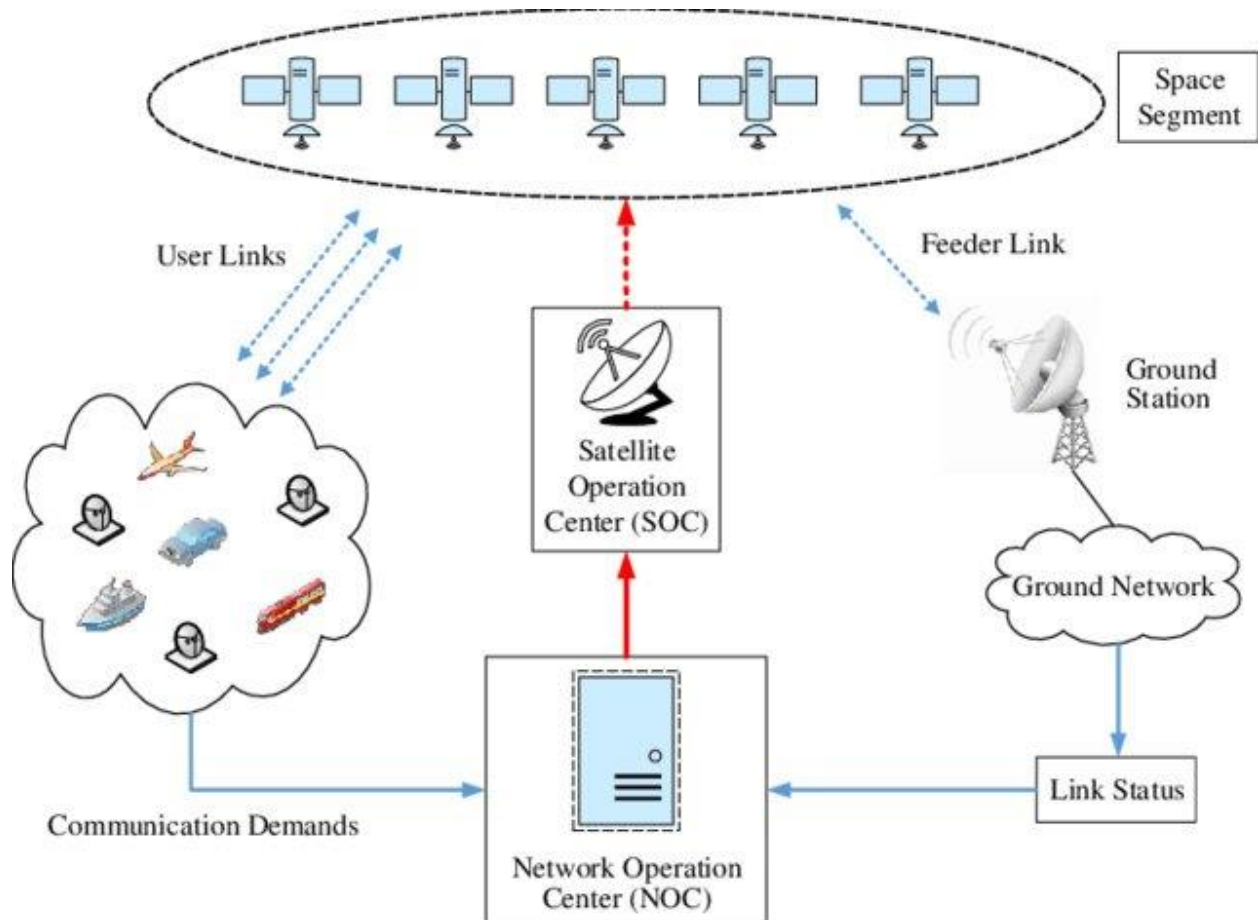


Figure 3: A satellite system architecture (Al-Hraishawi et al., 2023)

### 2.2.2 Satellite Security

Like all engineering systems, satellite systems can be subjected to undesirable scenarios that may be brought about intentionally or not. From the very rough conditions of Outer Space to targeted attacks conducted by malicious individuals, many risks can cause adverse effects on the well-being of a satellite. While some effects can be easily and readily reversible, some can be catastrophic and result in a total inability to use the system. To properly assess the risks that are involved in any system, it is beneficial to define a set of terms that will help us reliably identify and differentiate between different objects; we will use the following terms as defined in (pfleeger, 2015):

- a) Vulnerability - a weakness that is present in a system that could cause harm

- b) Threat - a specific set of circumstances that could cause harm
- c) Attack - an intentional exploitation of a vulnerability

Another set of terms that will prove important for understanding security threats in satellite systems is the terms that define good security principles. They are described in the ISO7498-2:1989 standard and have been recognized as important security principles in space systems, being present in the CCSDS report “Security Threats Against Space Missions” (CCSDS, 2015); again, the definitions provided in (pfleeger, 2015) will serve as our base:

- a) Confidentiality - the ability to be accessed only by authorized parties
- b) Integrity - the ability to be modified only by authorized parties
- c) Availability - the ability to be used by any authorized party
- d) Authentication - the ability to confirm the identity of a person/system that performs an action.
- e) Accountability (nonrepudiation) - the ability to confirm that someone/something performed a certain action without it being possible for them to deny it in any convincing manner

The above security properties can and, in well-designed systems, should be applied to any valuable asset, such as hardware, software, human personnel, communication channels, etc.

This provides a broad analysis of the various dangers that should be considered when designing satellite systems, especially intentional cyberattacks and how they can be identified, prevented, and mitigated.

### **2.2.3 Vulnerabilities of Satellite Network**

In the current technological environment, satellite communication has become increasingly important in various applications and capabilities that enable commercial and military operations.

Furthermore, as the number of satellites deployed has increased, space-based assets have become a target for hackers attempting to steal critical data, potentially resulting in catastrophic repercussions (Brandon et al., 2018).

Additionally, a legacy satellite communications platform in space is more challenging to maintain than a terrestrial communication system (Alvarez & Walls, 2016). A terrestrial communication system allows quick upgrading and testing to assure communications, encryption, and increased cybersecurity, but the same is not valid for legacy satellite communications platforms in space. As a result, satellite networks are more vulnerable to inconsistency in software updating, inadequate encryption, and outdated IT equipment installed (Pavur, 2021).

Another point to note is that using botnets, ransomware, trojans, viruses, and other hacking tools can potentially disrupt the satellite network and possibly bring it to a halt (Aslan et al., 2023). Once the satellite communication (SATCOM) infrastructure is hacked, the problem may extend throughout the whole terrestrial infrastructure network. As a result, after the network has been penetrated, hackers may monitor traffic via the terminal, allowing them access to additional sensitive data such as log-in, traffic flows, photos, voice conversations, and so on (Pavur, 2021).

#### **2.2.4 Cyber Threats Against Satellite Networks**

In the context of satellite systems, Figure 4 shows four segments of cyber threats classified as space, user, link and ground (Brandon et al., 2018). There are two categories of cyber threats and attacks for satellite networks: passive and active.

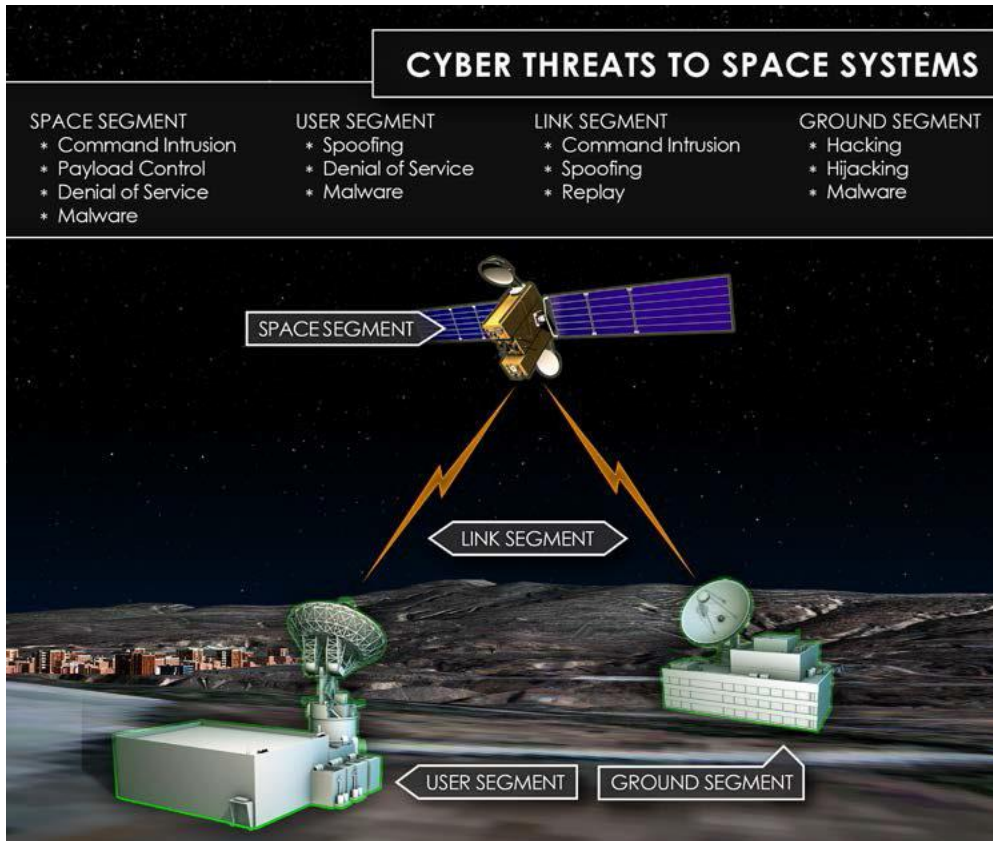


Figure 4: Cyber threats identified by NASIC.

#### 2.2.4.1. Passive Attacks

Using a satellite terminal with a fundamental understanding of communication protocols, a passive adversary scrutinises and monitors communications designated for alternate terminals. The acts of eavesdropping and spoofing constitute a distinct subset of passive attacks, wherein unauthorized access to information occurs between two interconnected devices within the expansive realm of the Internet (Wang, 2018). Nonetheless, these attacks do not directly impact system resources and present a challenge in detection, given the absence of discernible alterations or manipulations to the intercepted data.



#### **2.2.4.2 Active Attacks**

A malicious actor possesses the ability to exploit a network by modifying or altering its content, potentially leading to a significant impact on the system's resources and functionalities (Eder-Neuhauser et al., 2017). Consequently, the integrity and availability of the system are compromised, rendering it vulnerable to active attacks. This attack category encompasses many possibilities and techniques that can be employed.

- 1. Unauthorised Access and Hacking:** Unauthorised access and hacking present significant challenges to the security of ground stations. Threat actors exploit ground station software, hardware, or network infrastructure vulnerabilities to gain unauthorized entry. Upon gaining access, they can disrupt operations, exfiltrate sensitive data, or launch additional attacks on the system or interconnected systems. Introducing malware into ground stations is a common tactic facilitated by infected email attachments, compromised software updates, or physical intrusion. Social engineering, which involves manipulating individuals to reveal sensitive information or take actions that compromise system integrity, also enables unauthorised access. Examples of such techniques include impersonating authorised users or technicians to gain access or persuading unwitting users to disclose login credentials (Varadharajan, 2023).
- 2. Malware:** Malware poses a significant threat to satellite systems, particularly given their autonomous nature and limited maintenance accessibility. Malicious software can spread from one satellite to another, posing complex challenges. Malware can lead to system malfunctions or the generation of falsified data, with serious implications for critical missions. For example, infecting a weather monitoring satellite with malware could transmit inaccurate data, potentially impacting weather forecasting accuracy. Satellites

may be infected with malware during production, through tainted software updates, or via ground stations overseeing satellite operations.

- 3. Jamming and Interference:** Jamming and interference pose an additional peril to satellite systems. Hackers can disrupt satellite signals, resulting in system failures or data misdirection with significant consequences. For instance, the transmission of jamming signals can cause a navigation satellite to provide erroneous information to an aircraft, endangering lives. The sources of jamming and interference encompass malicious actors utilizing jamming equipment, electronic devices near the satellite, or natural phenomena such as solar flares (Diro et al., 2024).
- 4. Denial of Service (DoS) Attacks:** Denial of Service (DoS) attacks aim to render a specific system or service inaccessible to its designated users by inundating it with a substantial volume of traffic or requests. Within satellite systems, a DoS attack can precipitate system shutdown or unavailability, disrupting crucial services like communication, navigation, and remote sensing. Perpetrators employ diverse tactics, including overwhelming the system with traffic from numerous infected devices (termed a distributed denial of service (DDoS) attack), exploiting software vulnerabilities, or targeting the network infrastructure underpinning the system (Thangavel et al., 2022).
- 5. Spoofing and GPS Signal Manipulation:** Spoofing involves transmitting counterfeit signals to deceive satellite receivers and modify their functions. Spoofing assaults can concentrate on GPS signals within spatial information systems, resulting in imprecise positioning, navigation, and timing data (Xiao, 2022). By manipulating GPS signals, malicious entities can impede essential transportation, logistics, and communication operations.

Implementing anti-spoofing methodologies, such as signal authentication and sophisticated receiver technologies, is imperative for mitigating this hazard (Suo et al., 2022).

6. **Physical Attacks and Kinetic Anti-Satellite (ASAT) Weapons:** The security of space assets is confronted with a substantial threat from physical attacks. Kinetic Anti-Satellite (ASAT) weapons are designed to incapacitate satellites through direct collisions or fragmentation. These attacks can lead to the degradation of critical functionalities, the generation of space debris, and potential harm to other operational satellites (“Anti-satellite weapons and international law,” 2023; Bongers & Torres, 2023). The advancement and deployment of ASAT weapons underscore the need for enhanced space situational awareness, satellite manoeuvrability, and debris mitigation strategies (Bongers & Torres, 2023).
7. **Insider Threats:** Individuals within organizations or agencies who abuse their access privileges to compromise the security of space information systems are identified as insider threats (Bunn, 2023; Idris & Damilola, 2023). This could involve employees with malicious intentions, negligent conduct, or inadvertent actions resulting in unauthorized access, data breaches, or system disturbances. Robust mechanisms for detecting and preventing insider threats, such as stringent access controls, regular audits, and employee training programs, are vital for mitigating this risk (Idris & Damilola, 2023).
8. **Supply Chain Attacks:** Supply chain attacks target the software, hardware, or firmware components of space information systems during their production, distribution, or maintenance phases (Hammi & Zeadally, 2023). Adversaries may infiltrate the supply chain to introduce compromised or counterfeit components, thereby gaining unauthorized access, control, or manipulation of the systems (Berry, 2023). Rigorous security measures within the supply chain, including trusted manufacturing processes, component validation,

and secure software development practices, are essential for countering this threat (Berry, 2023).

9. **Cyber Espionage:** State-sponsored or industrial cyber espionage endeavours pose a significant threat to space information systems. Malicious entities may pursue sensitive information regarding space technology, satellite operations, or research and development projects. By infiltrating networks and systems, they can acquire valuable insights, intellectual property, or strategic advantages. Effective network monitoring, intrusion detection systems, and encryption technologies can assist in identifying and mitigating cyber espionage attempts (Knez, 2016). These cyber threats underscore the imperative for robust security measures to safeguard space information systems. Addressing these risks necessitates a multi-faceted approach involving technological innovations, policy frameworks, and international collaboration (Li, 2023). Enhanced security measures for software and hardware must be implemented in ground stations to detect and counter hacking endeavours. Regular assessments of vulnerabilities and protocols for managing patches are crucial for addressing potential system weaknesses.

These active attack techniques underscore the importance of implementing robust security measures to safeguard systems and networks from malicious actors seeking to compromise their integrity and availability.

### **2.3 INTRUSION DETECTION SYSTEM.**

Intrusion detection systems (IDS) are software-based components that act as an "alert" to safeguard data management from network intrusions. IDS may prohibit efforts to penetrate the Network by identifying misuse and illegal network access. Intrusion detection systems (IDSs) detect attacks and abnormal behaviour in satellite systems (Ashraf et al., 2022). IDS is categorized into five

categories, as shown in Fig. 4. IDSs can be divided according to their structure: Based on centralized IDS or distributed IDS. The centralized IDS analyses data at several locations (Eshakagdy et al., 2022). The distributed IDS, which includes multiple IDS through a large network, all of which are connected or connected to a central server that provides advanced network monitoring and incident analysis, can also be divided according to their deployment location into host-based intrusion detection systems (HIDS) and network-based intrusion detection systems (NIDS) (Vinayakumar et al., 2022). HIDS in the IDS system uses the system's activities in the form of log files running on the local host computer to detect attacks. HIDS analyzes system logs, file integrity, and system calls to detect unauthorized access, malware infections, or unusual behaviour on a specific host. By monitoring system activities and configurations, HIDS can identify deviations from normal behaviour and raise alerts for potential intrusions (Liu et al., 2018).

However, NIDS in the IDS system uses network behaviour. NIDS are strategically positioned at key points within the network to analyze incoming and outgoing traffic. They can detect anomalies such as unusual data packets, unauthorized access attempts, or denial of service attacks. NIDS uses signatures and behavioural analysis to identify potential threats (Ho et al., 2021). The network behaviours are gathered using network equipment mirroring by networking devices, such as routers, switches, and network taps, and analyzed to specify attacks and possible threats hidden within network traffic. The log files in NIDS are gathered through local sensors. While HIDS depends on the information in log files, which include system logs, sensor logs, file systems, software logs, user account information, disc resources, and others for each system, NIDS inspects the contents of each packet in network traffic flows. Many organizations utilize a hybrid of both HIDS and NIDS (Azar et al., 2023).

In addition, IDS can be divided according to the approach used to detect attacks and other hidden potential threats within network data into two categories: Anomaly-Based detection and Signature-Based detection, also known as “misuse detection” or “knowledge-based detection”. Anomaly-based detection detects deviations from normal behaviour. The role of this technique is to establish a baseline for the normal behaviour of network traffic and then compare the incoming traffic with this baseline to detect malicious attacks. This IDS type detects unknown and known attacks (Ahmed & Hamad, 2021). Signature-based detection has predefined signatures for known attacks that are matched with all connection patterns in the network to detect and stop any anomalous attacks. The main advantage of this type of IDS is that it detects known attacks. However, unknown attacks have not been detected due to the unavailability of attack signatures. According to their response, IDSs are classified into passive IDS, which monitor, log, and provide alerts to activity, and active IDS, which act based on software design (Eshakagdy et al., 2022). The maximum point of this method is the ability to detect an intrusion of a known pattern with a low false alarm rate. The setback of SIDS is the inability to detect a new attack of an unknown (dataset) pattern, such as a zero-day attack.

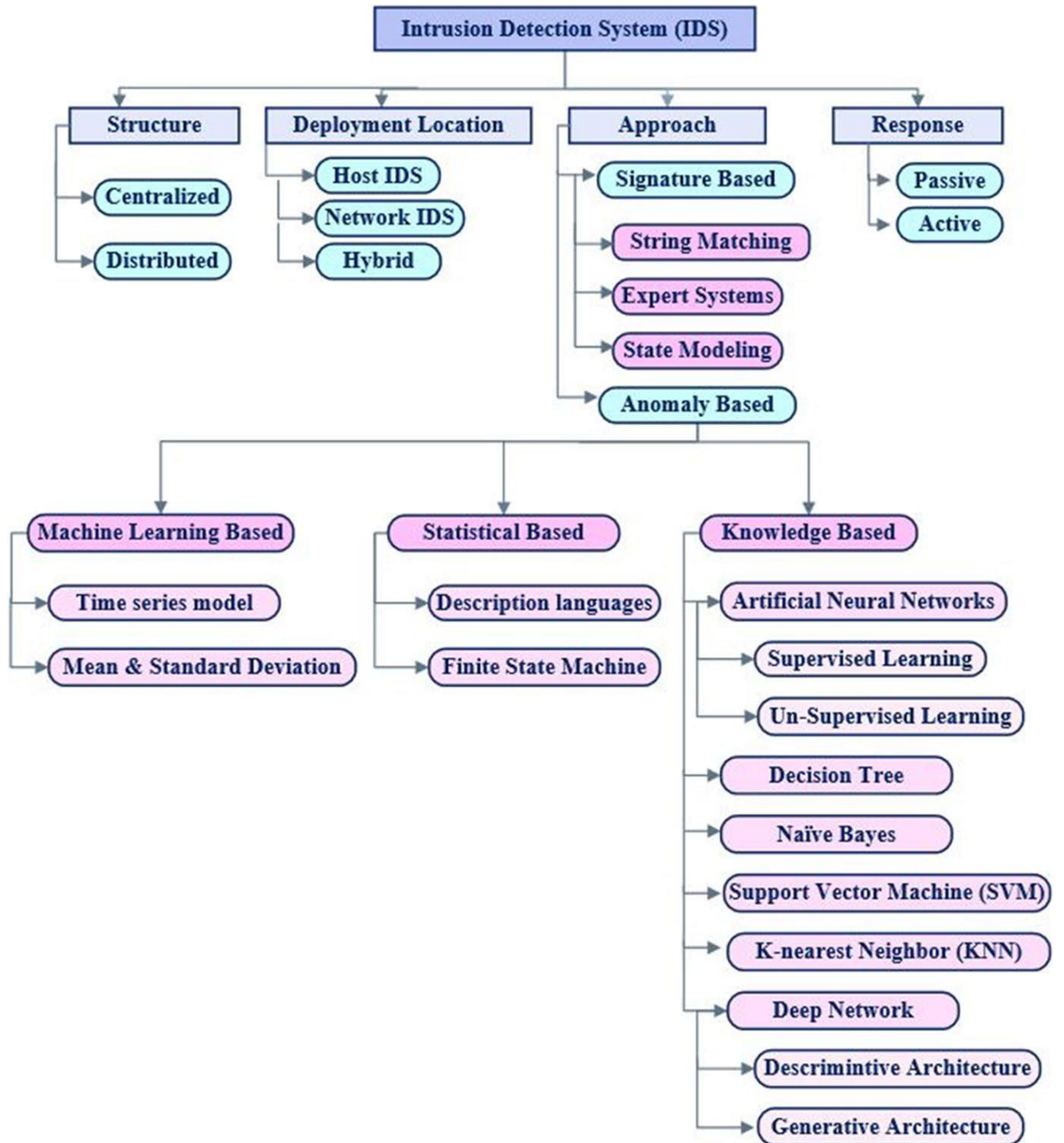


Figure 5: Categories of intrusion detection systems (IDS) (Eshakagdy et al., 2022)

## 2.4 MACHINE LEARNING.

Machine learning has become a fledged line of research by various researchers in recent years because of its ability to learn from vast amounts of data without thorough programming (Haque et al., 2023). Numerous researchers concentrate more on using machine learning for

anomaly detection in networking systems. Machine learning is a field under Artificial Intelligence (AI); the ability to learn from past data records without explicit programming and make accurate predictions has drawn much attention to using it as a tool for anomaly detection. ML is divided into supervised, unsupervised, and Reinforcement learning (Thoutam et al., 2023). In supervised learning, the training classification algorithm uses labelled data (input and output data). Likewise, unsupervised learning uses unlabeled (input data) data in the training classification algorithm without a predefined output or target class. Conversely, reinforcement comprises supervised and unsupervised learning approaches (Gajda et al., 2022).

#### **2.4.1 Machine Learning for Intrusion Detection.**

As part of Artificial Intelligence, Machine Learning employs an inductive learning approach to acquire knowledge through practical examples (Kola, 2022). The learning process can be categorised into three main types: Supervised, Unsupervised, and Reinforcement Learning (Shaveta, 2023).

##### **2.4.1.1 Supervised Machine Learning Algorithms.**

In supervised learning, a correct classification is already assigned to train a data sample from the data source (conneau et al., 2017). It can also be seen as a formalization of learning from examples where an input and desired output rely on data with predefined target classes to identify relationships between the data and their respective target classes (Dey, 2016). The supervised learning process involves two key phases. Training and evaluation. The training phase requires initial preprocessing steps to ensure accurate generalisation, such as feature selection, encoding, and normalisation. Commonly utilised algorithms for classification tasks include SVM, KNN, DT, and Random Forest. Each classifier excels at detecting specific types of attacks, with some being more effective in particular attack categories. Conducting theoretical and empirical analyses of Machine Learning algorithms is crucial for understanding their



computational complexity when applied to specific problem domains like Intrusion Detection. This study investigates the empirical and mathematical aspects of SVM, KNN, and Decision Tree to determine the most suitable machine learning model that can be extended as an accurate and lightweight intrusion detection system in the space sector for satellite networks.

There are different algorithms of supervised learning techniques, as discussed below:

#### **2.4.1.1.1 Logistic Regression.**

Logistic Regression is a statistical method used for binary classification problems [Perlich, C., Provost, F., & Simonoff, J. (2003). Tree induction vs. logistic regression: A learning-curve analysis.]. Logistic regression is included in the class of generalised linear models, which consists of a wide variety of models developed to expand the conventional linear model to include target variables with different properties. (McCullagh and Nelder, 1989; Hosmer and Lemeshow, 2000)

#### **2.4.1.1.2 K-nearest neighbour (KNN) Classifier.**

As an unsupervised machine learning, which is predicted with a labelled dataset, KNN has been categorised among the non-parametric ML classifiers since there is an absence of any relationship between input and output. Also, a lazy learner algorithm is a result of its inability to carry out the learning process, except there is a need to classify the new dataset (Dewan et al., 2022). KNN has been implemented in various studies to predict ML models, such as intrusion detection and financial industries. This is because of its easy-to-apply characteristics to problems and the ability to tolerate and resist noise during the learning phase (Sindhu et al., 2022). Since the significant idea is to compute the distance, KNN poses some problematic issues, such as deciding the suitable value of K and high computational complexity. Additionally, during the learning phase of KNN, the training dataset might need to be under-sampled as it may result in too many tie problems in KNN.

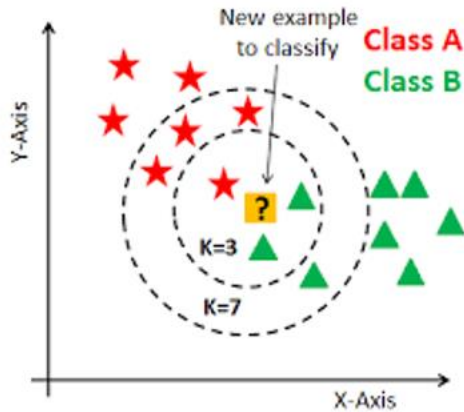


Figure 6: KNN (Tahri et al., 2022)

### 2.4.1.1.3 DECISION TREE

A decision tree is a non-parametric inductive learning classifier for classification and regression. Decision Trees are intuitive models that make decisions based on a series of rules inferred from the training data. They effectively handle numerical and categorical data, providing transparent and interpretable results. Decision Trees capture complex relationships in data and are particularly useful for feature selection and identifying important variables in intrusion detection (Chi et al., 2022). DT operates with bottom-down approaches and employs a greedy method for tree splitting; various decision trees have been introduced and engaged in the literature. The two splitting rules used in multiple versions of the decision tree are Gini and Entropy; Gini measures the probability that any element of the dataset will be mislabelled when it is randomly labelled. Entropy measures information that indicates the disorder of the feature with the target. The most recently used decision trees from the literature are C4.5, CART and C5.0. This study focused on CART since C4.5 and C5.0 are based on entropy using information gain; likewise, CART is based on the Gini splitting rule. The ability of the decision tree algorithm to adapt to categorical features, multiclass classification and missing values makes it unique from both KNN and SVM, as it requires less preprocessing of Data. DT is easy to understand, unlike black box SVM. An example of a decision

tree is shown below. The common challenge of DT is overfitting, which can be improved by Ensemble. Equations 1, 2, 3, and 4 indicate the mathematical notation of CART.

$$GainRatio(B) = \frac{Gain(B)}{SplitInfo(B)} \quad (1)$$

The node N splitting attribute determines which attribute gains the most information during the process. When the set V is partitioned based on attribute B, Information gain measures how much uncertainty is minimised.

$$Gini(B) = 1 - \sum_{i=1}^J p_i^2 \quad (2)$$

Where;

J is the total number of classes in the data set.

Pi is the ratio that a tuple in data set B.

The Gini index of B given any binary split on A partitions B into B1 and B2 is;

$$Gini_A(B) = \frac{|B1|}{|B|} Gini(B1) + \frac{|B2|}{|B|} Gini(B2) \quad (3)$$

Equation 10 is used to find the attribute with the minimum Gini Index

$$Gini(B) = Gini(B) + Gini_A(B) \quad (4)$$

The minimum Gini index features are used as the splitting features.

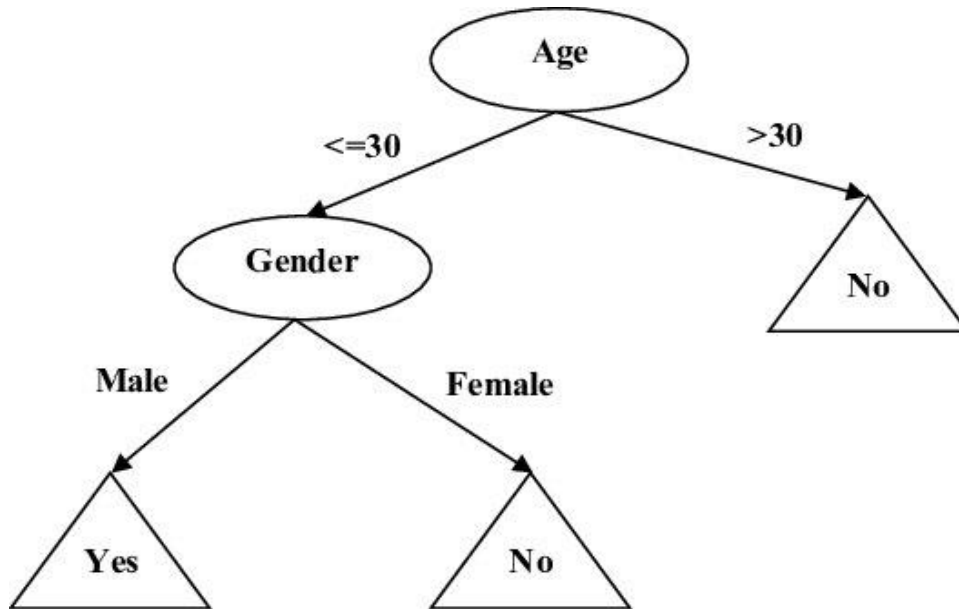


Figure 7: Decision Trees (Dey, 2016)

#### 2.4.1.1.4 NAIVE BAYES

This algorithm is mostly used and is a target of the text classification industry (Stoudenmire & Schwab, 2016). It is also used for clustering and classification purposes. Conditional probability is the backbone of the Naive Bayes algorithm, which creates trees based on the probability of occurring. These trees can also be regarded as a Bayesian Network. An example is shown below. It is constructed on the hypothesis that, for instance, for a given class, the attribute value is independent of the values of the attributes. This theory is called Class Conditional Independence.

$$P(H/X) = P(X/H).P(H)/P(X) \quad (5)$$

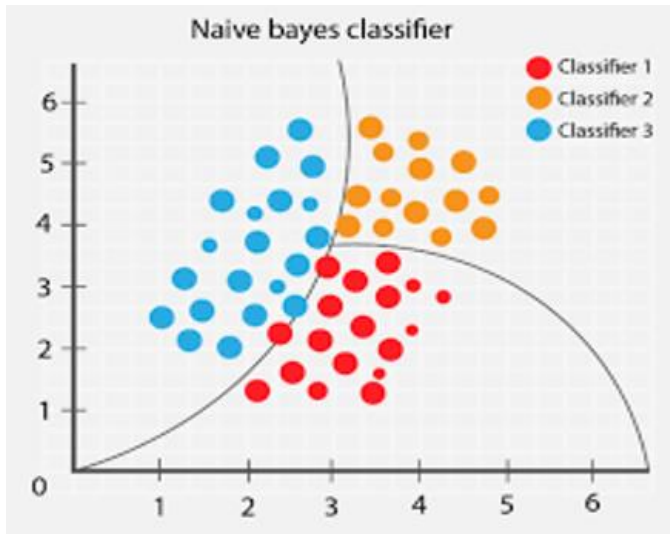


Figure 8: Naïve Bayes (Tahri et al., 2022)

#### 2.4.1.1.5 Support Vector Machine (SVM) Classifier.

SVM is also a non-parametric classifier and unsupervised Machine learning classifier. SVM has been widely used in various research domains, such as image, hypertext, and classification problems (Amir & Ali, 2022). Among the significance that SVM is not limited to (1) Possession of greater efficiency when it comes to higher dimensional space, unlike KNN, which might require under-sampling, (2) SVM also demonstrates significant advantage for its embedded feature selection techniques as a robust and helpful feature selection techniques compare to decision tree. Lastly, getting optimal results in SVM depends on its kernel function and parameters (Binitta & Leema, 2022). However, this value and parameter findings depend on the dataset as there are no specific values and parameters for the SVM kernel function and parameter. The diagram below shows a working support vector machine.

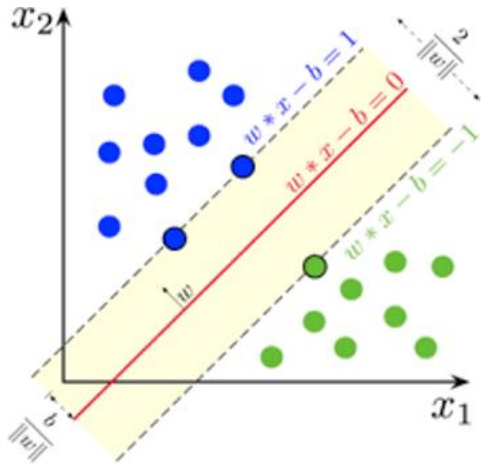


Figure 9: SVM (Tahri et al., 2022)

#### 2.4.1.2 Unsupervised Machine Learning Algorithms.

Unsupervised learning is a process in which ML algorithms learn from input space without predefined output. The significant difference between unsupervised and supervised is the absence of output space in unsupervised learning (David, 2023). Unsupervised learning, another branch of machine learning, encompasses techniques such as clustering and anomaly detection that are instrumental in various cybersecurity applications, including intrusion detection systems (Wahyono & Heryadi, 2019).

Clustering algorithms, such as Fuzzy-C-Means (FCM), are commonly used in unsupervised learning to group data points into clusters based on similarity. In cybersecurity, clustering can help identify patterns in network traffic or system behaviour that may indicate potential threats (Fu, 2022). By grouping similar data points, clustering algorithms can reveal anomalies or outliers that deviate from the norm, aiding in detecting suspicious activities within a network (Fu, 2022).

Anomaly detection, another key aspect of unsupervised learning, focuses on identifying data points significantly different from most datasets (Wahyono & Heryadi, 2019). Anomaly detection methods, like deep autoencoders, can learn the normal behaviour of a system and flag instances

that exhibit unusual patterns (Meidan et al., 2018). This approach is particularly useful for detecting unknown threats and zero-day attacks that do not have predefined signatures. By leveraging unsupervised anomaly detection techniques, organizations can enhance their intrusion detection capabilities and improve the security of their systems (Meidan et al., 2018).

The advantages of unsupervised learning techniques, such as clustering and anomaly detection, lie in their ability to uncover hidden patterns and anomalies in data without needing labelled examples. These methods can adapt to evolving threats and detect novel attack vectors that may go unnoticed by traditional signature-based systems. Additionally, unsupervised learning approaches are valuable for handling imbalanced datasets and detecting rare events that may indicate security breaches (Guo et al., 2021).

However, unsupervised learning methods also have limitations. Clustering algorithms may struggle with high-dimensional data or datasets with varying densities, impacting the quality of the clusters formed. Anomaly detection techniques may generate false positives if the model fails to distinguish between genuine anomalies and benign variations in the data. Moreover, unsupervised learning approaches may require careful parameters and feature selection tuning to achieve optimal performance (Prasad et al., 2022).

The choice between supervised and unsupervised machine learning approaches for satellite network threat classification is contingent upon the availability of labelled training data and the specific objectives of the intrusion detection system. Supervised learning approaches are more appropriate when labelled normal and malicious network traffic datasets are available for training classification models (Nguyen & Armitage, 2008). This scenario is typically encountered when there is existing knowledge about the signatures or patterns of different attacks, such as Denial of Service (DoS) or probing attacks collected from past incidents or honeypots. In such cases, labelled

examples can train models to recognise these known threats. Furthermore, in safety-critical satellite applications, the cost of false negatives (missed attacks) is extremely high, and supervised models trained on attack samples can provide higher precision in detecting these known threats (Nguyen & Armitage, 2008).

As new attack types emerge, they can be analysed, labelled, and incorporated into retraining the supervised model periodically to keep up with the evolving landscape (Ahsan et al., 2022). Additionally, if the goal is to detect specific attack categories like DoS, spoofing, or other types, labelled examples of each class allow for training tailored supervised classifiers (Ahsan et al., 2022).

On the other hand, unsupervised learning techniques are beneficial when labelled data is limited or unavailable, making them suitable for unknown attack detection. Unsupervised methods can identify anomalies that deviate from normal network behaviour patterns without relying on labelled attack examples, thereby allowing for the detection of previously unseen, zero-day attacks (Radoglou-Grammatikis et al., 2020). Moreover, obtaining accurate labels requires significant human effort, and unsupervised techniques can process unlabeled data and provide an initial separation of normal versus anomalous traffic, reducing the labelling needs.

If normal network traffic patterns evolve frequently, unsupervised techniques can adapt to the new normal behaviour without requiring retraining on labelled data (Usama et al., 2019). Furthermore, unsupervised methods may be useful for detecting insider threats by modelling normal user or system behaviour and flagging deviations that could indicate malicious insider activities (Usama et al., 2019).



Combining both approaches can be employed in a hybrid or ensemble architecture. This involves using unsupervised techniques to detect anomalies and isolate potential threats, analysing the detected anomalies and labelling them as known or unknown attacks, retraining supervised models on updated labelled data to improve known threat detection, and iteratively refining the system as new attacks are discovered (Takyi et al., 2018). This hybrid approach leverages the strengths of both methods: unsupervised for novel threat detection and supervised for accurate classification of known attacks, providing a comprehensive, evolving intrusion detection capability for satellite networks (Naveed et al., 2022).

#### **2.4.1.3 Reinforcement Machine Learning Algorithms.**

In this technique, the positive outcome of the decisions is determinant or dependent on the actions to take (Zoph & Le, 2016). The learner has no idea of the action to take until it is given a particular situation. Depending on the actions taken by the learner, the future is affected in terms of the situations. Below is a model for reinforcement learning. In the above model, input  $i$  is received by the agent. The agent also receives current state,  $s$ , state transition  $r$ , and input function  $I$  from the environment. With these inputs, the agent generates behaviour  $B$  and takes action  $A$ , which generates an outcome (Dey, 2016). The reinforcement learning technique is applied to natural language processing for dialogue creation, where a model simulates dialogues between virtual agents using policy gradients for reward to conversational properties (Li et al., 2016).

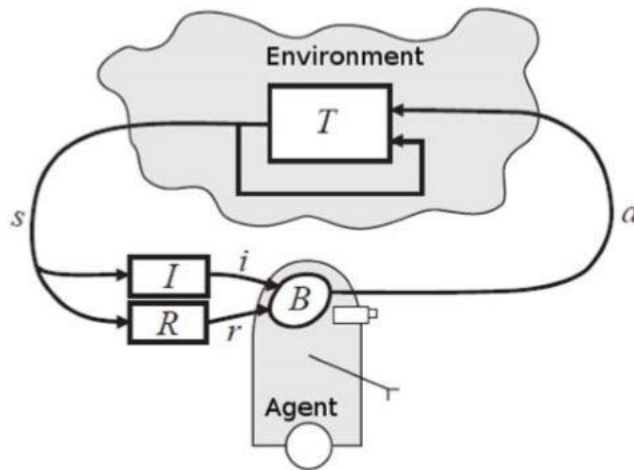


Figure 10: Reinforcement Learning Model (Dey, 2016)

#### 2.4.2 Challenges in Machine Learning for Intrusion Detection

Several potential issues need to be addressed when considering using machine learning (ML) in cybersecurity, including intrusion detection systems. These issues include the need for labelled data, computational complexity, and the risk of adversarial ML attacks.

1. Need for Labelled Data: Supervised ML algorithms, such as those used in intrusion detection, require labelled data for training. Labelling data can be time-consuming and resource-intensive, especially in cybersecurity, where accurately labelled datasets are crucial for training effective models. Insufficient or inaccurate labelling can lead to biased models and reduced detection accuracy, highlighting the importance of high-quality labelled datasets in ML-based intrusion detection systems (Orsini et al., 2022).

2. Computational Complexity: ML algorithms, particularly deep learning models, can be computationally intensive, requiring significant processing power and memory resources. In the context of real-time intrusion detection, the computational complexity of ML models may pose challenges in deploying efficient and responsive systems. Optimising ML algorithms for

performance and scalability is essential to ensuring timely threat detection and response in cybersecurity applications (Gao et al., 2019).

3. Risk of Adversarial ML Attacks: Adversarial ML attacks pose a significant threat to cybersecurity systems, including intrusion detection. Adversarial attacks involve manipulating input data to deceive ML models and cause misclassification. Adversarial examples can exploit vulnerabilities in ML algorithms, leading to false positives or negatives in intrusion detection systems. Robustness against adversarial attacks is crucial for ensuring the reliability and effectiveness of ML-based cybersecurity solutions (Jiang et al., 2020).

Addressing these issues requires a multi-faceted approach. Strategies such as data augmentation, transfer learning, and active learning can help mitigate the need for large labelled datasets and improve the efficiency of ML models in intrusion detection. Additionally, optimising algorithms for performance and leveraging hardware acceleration can address computational complexity challenges. Implementing robustness measures, such as adversarial training and model verification, can enhance the resilience of ML-based intrusion detection systems against adversarial attacks (McCarthy et al., 2022).

## **2.5 FEATURE SELECTION**

Feature selection is the process of selecting a relevant subset of features from the original feature set to improve model performance, efficiency, and interpretability. It helps reduce noise, multicollinearity, and redundancy in data (Sosa-Cabrera et al., 2023).

Choosing the right features is crucial for achieving effective machine learning model accuracy and efficiency, as the features represent the input data that the model uses to learn patterns and make predictions (Hasan et al., 2016). The quality and relevance of the features directly influence the model's capability to capture the underlying relationships present in the data and generalise well

to unseen instances. Several key reasons underscore the significance of feature selection (Hasan et al., 2016).

Real-world datasets often contain irrelevant or redundant features that can introduce noise and mislead the learning algorithm (Cai et al., 2018). By selecting only the most relevant features, the signal-to-noise ratio in the data improves, allowing the model to focus on truly predictive patterns (Das et al., 2022). As the number of features increases, the dimensionality of the data grows, which can lead to the "curse of dimensionality" problem. This phenomenon makes it increasingly difficult for the model to accurately learn the decision boundaries, especially when the number of training instances is limited compared to the feature space (Das et al., 2022).

Irrelevant or redundant features can cause the model to overfit the training data by capturing noise or spurious correlations that do not generalize to new data. Selecting a minimal set of informative features helps prevent overfitting and improves the model's generalization ability (Das et al., 2022). Training and evaluating machine learning models on high-dimensional datasets can be computationally expensive and time-consuming. Reducing the number of features lowers the computational complexity, leading to faster training times and more efficient model deployment (Ni, 2022).

When working with high-dimensional data, it becomes challenging to understand the influence and importance of each feature on the model's predictions. Feature selection can enhance interpretability by focusing on the most relevant features, making explaining the model's behaviour easier and gaining insights into the underlying patterns (Chen et al., 2020). In some cases, features may be highly correlated, leading to multicollinearity. This can cause instability and redundancy in the model's learned parameters. Feature selection techniques can identify and remove these redundant features, improving the model's robustness and interpretability (Chen et al., 2020).

Effective feature selection techniques, such as filter methods (e.g., chi-squared, information gain), wrapper methods (using the model itself to evaluate features), or embedded methods (performing selection during model training), can help identify the most discriminative and relevant features for the task at hand (López-Dorado et al., 2021). By focusing on these informative features, machine learning models can achieve higher accuracy, better generalization, improved efficiency, and enhanced interpretability, ultimately leading to more effective and reliable performance in real-world applications (Almomani, 2020).

### **2.5.1 Methods of Feature Selection**

Feature selection plays a vital role in machine learning by identifying the most relevant and informative features from the original feature set (Sosa-Cabrera et al., 2023). Various methods have been proposed to address this challenge, each with its strengths and weaknesses. These methods can be broadly categorized into filter, wrapper, and embedded methods (Sosa-Cabrera et al., 2023).

#### **2.5.1.1 Filter Methods**

Filter methods are a class of techniques that rely on statistical measures to evaluate the relevance of features independently of the machine learning model (Mayet et al., 2022). These methods are computationally efficient and can be applied as a preprocessing step before model training. One commonly used filter method is correlation analysis, which measures the strength of the relationship between each feature and the target variable. Features with high correlation coefficients are considered more relevant and selected for model inclusion (Mayet et al., 2022). Another popular filter method is information gain, which quantifies the reduction in entropy or uncertainty about the target variable when a specific feature is known (Riana & Mangkurat, 2023).

### **2.5.1.2 Wrapper Methods**

Wrapper methods, on the other hand, employ a machine learning model to evaluate feature subsets' usefulness. These methods typically employ a search strategy, such as forward selection, backward elimination, or recursive feature elimination, to iteratively evaluate different feature combinations. The feature subset that yields the best performance on a validation set is then selected for the final model. While wrapper methods can be computationally expensive, they often result in better feature subsets tailored to the specific model. (Faleiros et al., 2020).

### **2.5.1.1 Embedded Methods**

Embedded methods integrate feature selection as part of the model training process. These techniques simultaneously perform feature selection and model learning, effectively embedding the selection process within the model's objective function (Chen et al., 2020). One prominent example of an embedded method is L1 regularization, also known as the Lasso (Least Absolute Shrinkage and Selection Operator) technique. L1 regularization introduces a penalty term in the objective function that encourages sparse solutions, effectively driving the coefficients of irrelevant features to zero, thereby performing feature selection implicitly (Li et al., 2022).

Each method has its merits and drawbacks, and the choice often depends on the specific problem, the available computational resources, and the trade-off between computational complexity and model performance. Different feature selection techniques may be employed to leverage their complementary strengths and mitigate their weaknesses. Additionally, domain knowledge and expert insights can be incorporated into the feature selection process to enhance the quality of the selected features further and improve model interpretability (Sosa-Cabrera et al., 2023).

## **2.6 SPACE-SPECIFIC FEATURES TO IDENTIFY CYBER THREATS IN SATELLITE NETWORKS**

Identifying cyber threats in satellite networks requires carefully selecting features that capture relevant patterns and characteristics of network traffic, communication protocols, and signal characteristics. These features are crucial in enabling machine learning models to effectively detect and classify various cyber threats targeting satellite systems (Ronald et al., 2023).

Network traffic patterns are among the most important features to consider for cyber threat detection. These patterns can include statistics related to packet sizes, packet arrival rates, packet inter-arrival times, and the distribution of source and destination IP addresses and ports (Chen et al., 2020). Anomalies in these patterns may indicate the presence of denial-of-service attacks, network scans, or other malicious activities. Additionally, features related to the volume and frequency of specific types of network traffic, such as ICMP, TCP, or UDP packets, can provide valuable insights into potential threats (Chen et al., 2020).

Communication protocols used in satellite networks are another rich source of features for cyber threat detection (Ronald et al., 2023). Analyzing the characteristics of protocols like TCP, UDP, and specific application-layer protocols can reveal deviations from expected behaviour, which may indicate attacks or unauthorized access attempts. Features such as protocol flags, header fields, and payload content can be extracted and used to train machine-learning models for identifying potential threats (Jingyi Cai et al., 2023).

Signal characteristics are particularly relevant for detecting threats related to jamming, spoofing, or interference in satellite communications. Features derived from signal properties, such as signal strength, frequency, modulation, and signal-to-noise ratio, can be used to identify anomalies caused by malicious actors attempting to disrupt or manipulate satellite signals. Additionally,

features related to the spatial and temporal patterns of signal propagation can aid in detecting potential spoofing attacks or locating the sources of interference (Jingyi Cai et al., 2023).

In addition to these domain-specific features, general features related to system logs, user behaviour patterns, and network configurations can contribute to a comprehensive cyber threat detection system for satellite networks (Chen et al., 2020). Combining these diverse feature sets using appropriate feature engineering techniques can enhance the ability of machine learning models to detect and classify a wide range of cyber threats effectively (Cai et al., 2018).

It is important to note that selecting appropriate features is critical in building effective machine-learning models for cyber threat detection in satellite networks (Cai et al., 2018). Domain knowledge and expertise in satellite communications and cyber security are essential for identifying the most relevant and discriminative features. Furthermore, feature selection techniques, such as those discussed earlier, can be employed to identify the optimal subset of features, balancing model performance and computational efficiency (Cai et al., 2018).

## **2.7 RELATED WORKS**

Recent studies on intrusion detection systems show the improved performance of machine learning models. The growing network connection in satellite networks introduces additional risks and security challenges. DDoS is one of the most common attacks in satellite-terrestrial integrated networks and causes service delays. Many studies have been proposed for DDoS identification in satellite and terrestrial networks.

(Azar et al., 2023) This study proposed four hybrid intrusion detection systems for satellite-terrestrial communication systems (SAT-IDSs) to optimise the detection performance of malicious activities in network traffic. They utilised a sequential forward feature selection (SFS) method based on random forest (RF) to optimise detection performance and reduce execution time and



combine with Machine learning (ML) models: Random Forest (RF) and Deep learning (DL) models: Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU). The proposed models are evaluated and verified using the UNSW-NB15 and STIN datasets. The experimental results indicate that SFS-RF achieved 90.5% accuracy in the STIN dataset, and the RF-SFS-GRU model had the highest accuracy of 79% in the UNSW-NB15 dataset.

(Panigrahi et al., 2024) They proposed an ensemble-based trust model with the NSL-KDD+STIN+Exata-CDOS datasets, which implemented Ant Colony Optimization (ACO) to pick the shortest path while concurrently utilizing a to Detect DDoS in LEO Satellite-Terrestrial networks. The proposed model achieves 98% accuracy in detecting DDoS attacks, which is not comparable with existing protocols for performance evaluation. Researchers (Henry et al., 2023) proposed an approach that combined both CNN and GRU to optimize the network parameters. In this simulation, the authors used the CICIDS- 2017 benchmark dataset and metrics such as precision, recall, false-positive rate (FPR), and true-positive rate (TPR). The authors also performed a comparative analysis with other existing approaches, and the obtained results indicate the efficacy of the proposed IDS scheme in real-world cybersecurity setups.

(Ashraf et al., 2022) proposed a new approach-based intrusion detection method using data from satellite and terrestrial networks. The model combines random forest (RF) and multilayer perceptron (MLP) to increase the accuracy of intrusion detection compared to other machine learning models. They also analyse the efficiency of the proposed framework for the satellite and then use three datasets for experiments, namely NSL-KDD, KDD-CUP 99, and STIN. In addition, a performance comparison with state-of-the-art models is performed, which suggests that the RFMLP can detect intrusion attacks with higher accuracy than the existing approaches. Other researchers (Maseer et al., 2021) conducted a comprehensive analysis of the important features of

large traffic in networks to enhance the accuracy of the intrusion detection model and reduce the execution time. They use the Information Gain method as a feature selection method to select important features and then implement Bayes Net (BN), Random Forest (RF), Naive Bayes (NB), J48, and Random Tree (RT) classifiers. The results of experiments on the CICIDS-2017 dataset significantly improved the accuracy and execution time, with the Random Forest model (RF) achieving the highest accuracy of 99.86% based on 22 selected features. In comparison, the J48 model achieved an accuracy of 99.87% based on 52 selected features but with a longer execution time. These findings have practical implications for improving the efficiency of intrusion detection systems.

This study discusses machine learning methods using the NSLKDD dataset. Correlation analysis was employed as the FS method, which reduced the features into 5; a tree-based ML model was implemented on the reduced feature. Random Forst Decision Tree and XGBoost were employed in the SDN controllers as NIDS to monitor network traffic and detect malicious behaviour. Notably, XGBoost outperforms the other two methods, achieving a high F1-score of 95.95% in a multiclass classification task, while Random Forest and Decision Tree achieve slightly lower F1-scores of 94.6% and 94.5%, respectively. XGBoost also shows better precision and recall rates. The study indicates that the decision tree-based method can contribute significantly as an IDS in SDN Networking. Kevric et al. (2017) pointed out that combining two tree algorithm models can achieve better performance than separate tree classification models; the best combination they reported was a random tree and NB tree. The model was tested on the KDD dataset, and an accuracy of 89.24% was obtained. Al-Qatf et al. (2018) successfully combined upstream AE and downstream SVM, and the model obtained 84.96% binary classification accuracy when tested on the KDD dataset.

(Chohan et al., 2023) This study discusses solutions for cyber-attacks in intelligent electric vehicles and power systems using machine Learning-based IDS for network threat detection. This work presents a comparative analysis of various ML algorithms trained over the UNSW-NB15 dataset. ADA Boost, Linear Support Vector Machine (LSVM), Auto Encoder Classifier, Quadratic Support Vector Machine (QSVM), and Multi-Layer Perceptron algorithms are employed in the Python simulation. ADA Boost shows better results than other traditional techniques, with an excellent accuracy of 98.3%. Ingre and Yadav (2015) proposed an artificial neural network (ANN) model and a hybrid model that improves detection performance by combining different state-of-the-art algorithms. The latter achieved 81.2% accuracy for the NSL-KDD dataset. Sahu et al. (2022) proposed LSTM (Long Short-Term Memory) combined with FCN (Fully Connected Network) deep learning approaches to classify the normal and anomalous connections on intrusion datasets and specify the attack pattern more accurately. The proposed deep learning model achieved better classification accuracy using the KDDCup99, NSLKDD, GureKDD, KDDCorrected, Kyoto, and NITRIDS datasets.

(Jiang et al., 2020) proposed a robust UAV and satellite-based 5G network security model based on machine learning to bolster network security by effectively detecting vulnerabilities and cyberattacks. This approach is divided into two parts: creating the model using different machine learning algorithms and implementing the ML-based model using satellite or terrestrial gateways. The model achieves maximum accuracy with a 99.99% true negative rate and a 0% false negative rate using a decision tree algorithm, underscoring its reliability compared to other ML classifiers. Musaffer et al. (2020) designed a sparse autoencoder for an intrusion detection system on a reliable and updated network attacks dataset, CICIDS2017. The authors proposed a deep learning model, namely a memetic algorithm for abnormal traffic detection, and tested it on two well-known

datasets: NSLKDD and KDDCUP 99. Feature augmentation has been applied along with SVM to provide an effective intrusion detection framework and achieved robust results regarding training speed and faulty alarm rate (Gu et al., 2019). This research's reliability and robustness provide a solid foundation for further exploration and application in network security.

The study (Chandrashekar & Sahin, 2014) proposed a new approach-based intrusion detection. To classify normal and anomalous traffic, they compared it with different machine learning techniques, including SVM, AdaBoost, decision tree, and MLP. It depends on selected features based on the correlation between the features, and it is implemented using the UNSW-NB 15 dataset for network anomaly detection. The proposed approach achieves high accuracy in binary classification using Adaboost, which is 99.3%. These studies represent research efforts for devising suitable approaches for intrusion detection in satellite networks. A comparative analysis of the discussed research works is provided in Table 1.

Table 1: Comparative analysis of the existing approaches.

S/N	Author/Year	Title of paper	Contributions	Limitation
1	Panigrahi et al., 2024	A Smart Secure Model for Detection of DDoS Malicious Traces in Integrated LEO Satellite-Terrestrial Communications.	Ensemble-based trust model integration for Detection of DDoS malicious traces in LEO Satellite-Terrestrial networks.	Lack of comparison with existing protocols for performance evaluation.
2	Ashraf et al., 2022	A Deep Learning-Based Smart Framework for Cyber-Physical and Satellite System Security Threats Detection.	Integrates random forest (RF) and multilayer perceptron (MLP) to produce an RFMLP model and increase intrusion detection performance.	Low performance of the 'Syn_DDoS' class in the STIN dataset.
3	Maseer et al., 2021	Benchmarking of machine learning for anomaly-based intrusion detection systems in the CICIDS2017 dataset.	Information Gain method-based Bayes Net (BN), Random Forest (RF), Naive Bayes (NB), J48, and Random Tree (RT) classifiers Improving the accuracy of the intrusion detection model and minimize the execution time.	The proposed approach addressed a binary classification problem to detect attacks in the network traffic, regardless of the attack category.

4	Aburomman et al., 2016	A novel SVM-kNN-PSO ensemble method for an intrusion detection system.	Ensemble model SVM-KNN-PSO using a weighted algorithm for high accuracy.	Multi-class problem is not handled.
5	Gu et al. (2019)	Novel approach to intrusion detection using SVM ensemble with feature augmentation.	Feature augmentation with SVM to provide effective intrusion detection and achieved robust results in faulty alarm rate.	Configuration for different datasets is difficult.
6	Musafer et al., 2020	An enhanced design of sparse autoencoder for latent features extraction based on trigonometric simplexes for network intrusion detection systems.	Uses trigonometric simplexes.	Sparsity constraints.
7	Azar et al., 2023	Deep Learning-Based Hybrid Intrusion Detection Systems to Protect Satellite Networks.	Proposed four hybrid intrusion detection systems for satellite-terrestrial communication systems. And utilized the sequential forward feature selection method to optimize detection performance.	

---

## CHAPTER THREE

### MATERIALS AND METHODS

This section elaborates on the model's implementation, including the benchmark dataset and performance evaluation matrix.

#### 3.1 BENCHMARKING DATASETS.

Benchmark datasets are publicly available data over the Cloud. Various researchers have created them to improve intrusion detection research. Generally, each benchmark dataset consists of many features or attributes contributing to individual attacks, such as DDoS attacks. Also, each attack has unique features that contribute to its establishment. This study used publicly available datasets STIN, UNSW-NB15, and CICIDS2017.

##### 3.1.1 STIN Dataset

The STIN dataset (Li et al., 2020) represents a satellite dataset that includes attacks in modern satellite and terrestrial network environments. The authors (Li et al., 2020) simulate a real scenario for the terrestrial and satellite networks. This dataset contains two types of traffic, TER20 and SAT20, in CSV format. These two files contain 32 features with labels and nine different types of attacks. The distribution of the samples for types of attacks in training set for each category is mainly one for terrestrial network attacks, which include seven various types of attacks like Botnet with 14,622 records, Web Attack with 13,017 records, Backdoor with 12,762 records, LDAP\_DDos with 15,694 records, MSSQL\_DDos with 15,688 records, NetBIOS\_DDos with 11,530 records, and the last type is Portmap\_DDos with 14,380 records. Another one is for satellite network attacks, which include Syn\_DDos with 54,789 records and UDP\_DDos with 57,082

records. Flow-based features are considered when building the STIN dataset. Table 1 presents the characteristics of the dataset.

Table 2: Details of STIN dataset.

Domain	Attack Type	Attack Time
Terrestrial attacks	Botnet	15:01 → 15:10
	Web attack	15:21 → 15:31
	Backdoor	15:41 → 15:52
	LDAP DDoS	16:01 → 16:11
	MSSQL DDoS	16:21 → 16:30
	NetBIOS DDoS	16:41 → 16:50
	Portmap DDoS	17:01 → 17:13
	Syn DDoS	17:21 → 17:32
	UDP DDoS	17:41 → 17:52
Satellite attacks	Syn DDoS	15:23 → 15:57
	UDP DDoS	16:52 → 17:20

### 3.1.2 UNSW-NB15 Dataset

Nour Mustafa and Jill Slay published the UNSW-NB15 dataset in 2015 to improve the NSLKDD dataset. This dataset comprises 47 features categorised into flow, basic, content, time, and general-

purpose features. The UNSW-NB15 dataset represented terrestrial traffic only, including various attacks in modern terrestrial network environments. This dataset contains two main files, mainly the training set and testing set files in CSV format, and it includes 45 features with labels and nine different types of attacks. The distribution of the samples for types of attacks in the training set is Analysis with 677 records; Backdoor with 583 records; DoS with 4089 records; Exploits with 11,132 records; Fuzzers with 6,062 records; Generic with 18,871 records; Reconnaissance with 3,496 records; Shellcode with 378 records; and Worms with 44 records. However, the distribution of the samples for types of attacks in the testing set for UNSWNB15 is: Backdoor with 1,746 records; Analysis with 2,000 records; DoS with 12,264 records; Exploits with 33,393 records; Fuzzers with 18,184 records; Generic with 40,000 records; Worms with 130 records; Shellcode with 1,133 records; and Reconnaissance with 10,491 records.

Table 3: Record distribution of UNSW-NB15 Dataset

Dataset	Total sample size	Normal	Attacks
Training set	175,341	56,000	119,341
Testing set	82,332	37,000	45,332

### 3.1.3 CICIDS2017 Dataset

CICIDS2017 Dataset was provided by the Canadian Institute of Cyber Security, which offers detailed real-world attacks. The team prioritised generating realistic network traffic using a benign profile system (Sharafaldin et al., 2016) that abstracts the behaviour of human interactions and generates naturalistic benign background traffic. This dataset collection, which included five days' worth of the Canadian Institute of Cybersecurity's normal and assault traffic statistics, was spread



across eight files. DoS/DDoS attacks were captured on Wednesday, July 5, and Friday, July 7, 2017. The data was created in a test bed infrastructure with two separate networks: victim-network and Attacker-Network. The research team included a victim-network router, firewall, switch, and equipment with Windows, Linux, and Mac Intosh operating systems. The network consists of a router, one switch, and four PCs. The CICIDS2017 Wednesday was used in this study and partitioned into training and testing datasets. Table 3 shows the statistical records of this dataset. The dataset used in this research contains normal network flows and flows with DDoS attacks.

Table 4: Record distribution of CICIDS2017 Wednesday Dataset

Dataset	Total sample size	Normal	Attacks
Training set	112,642	48,695	63,947
Testing set	113,069	48,991	64,078

### 3.2 CLASSIFIER COMPLEXITY.

The computational complexity of various learning algorithms is crucial to investigate. The empirical analysis might seem insignificant as it may vary when implemented in different environments. Meanwhile, theoretical analysis has been the significant focus of researchers, showing the worst case of each learning algorithm since it does not depend on input size and implementation environment.

#### 3.2.1 Theoretical Analysis of Decision Tree.

The idea behind DT is a top-down approach that follows divide and conquer. The basic operation of CART DT is Entropy gain.

Sample Complexity:  $O(n)$

(6)

Running – Time complexity:  $O(n \log_2 n)$  (7)

Time Complexity for Training:  $O(Vn \log_2 n)$  (8)

Where n: The number of points in the Training set and,

V: The dimension of the data.

The total running time complexity of the C5.0 decision tree is given by;

$$T(S, O, X) = O(n) + O(n \log_2 n) + O(Vn \log_2 n) \quad (9)$$

S is the training set, O is the features, and X is the target class.

### 3.2.2 Theoretical Analysis of KNN

KNN is a lazy learner algorithm that follows brute-force implementation. Its basic operation is the distance function.

Sample Complexity:  $O(n)$  (10)

Running Time complexity:  $O(Kn)$  (11)

Time Complexity for Training:  $O(Knd)$

Where n = number of samples in the Training set,

K = number of Neighbours and,

d = dimension of dataset

The total running time complexity of KNN is given by,

$$K(S, O, X) = O(n) + O(kn) + O(knd) \quad (12)$$

S is the training set, O is the features or sample, and X is the sample target.

### 3.2.3 Theoretical Analysis of SVM

SVM is a black box classifier; the basic operation of SVM is the kernel function.

$$\text{Sample Complexity: } O(n) \tag{13}$$

$$\text{Running Time Complexity: } O(n_{sv}d) \tag{14}$$

$$\text{The time complexity for Training: } O(n^2d + n^3)$$

Where n = number of Training samples,

$n_{sv}$  = number of support vectors and

d = dimension of dataset.

The total running time complexity of SVM is given by;

$$(SVM(S, O, X) = O(n) + O(n_{sv}d) + O(n^2d + n^3) \tag{15}$$

### 3.3 DATA PREPROCESSING

Preprocessing in machine learning is a critical stage that contributes to producing an efficient model in machine learning (ML). This process involves preparing the data for analysis by cleaning, transforming, and selecting relevant features. With the advancement of technology, cyberattacks have become increasingly sophisticated and challenging to detect. Therefore, it is crucial to preprocess the data effectively to ensure the accuracy and reliability of the machine learning model. This process helps give an equal preference to a dataset in terms of normalisation and dimensionality to aid model improvement. Not only that, but the dimensional reduction process also helps reduce the computational complexity of the model. Finally, it is essential to mention the

aspect of encoding since it is crucial and necessary for both SVM and KNN, which was used in this study.

One of the critical tasks in data preprocessing is data cleaning, which involves identifying and correcting errors and inconsistencies in the data. This process ensures that the model produces reliable and accurate predictions. Alrashdi et al. (2019) highlighted that using machine learning algorithms to detect cyberattacks in an intelligent city requires data cleaning to remove irrelevant data and improve the dataset's quality. Similarly, Snider et al. (2021) emphasised the importance of data cleaning in reducing the potential for bias and errors in cybersecurity policy-making. Therefore, data cleaning should be performed before any other preprocessing task to ensure the quality of the data.

Another essential task in data preprocessing is feature selection, which involves identifying and selecting the most relevant features for the model. Feature selection is crucial in reducing the dimensionality of the dataset and improving the performance of the machine learning model. Khazaei (2021) demonstrated the effectiveness of feature selection in detecting stealthy cyberattacks on smart grids. The study used a random forest algorithm to identify the most critical features and reduced the dimensionality of the dataset by 50%. This approach significantly improved the accuracy of the model. Therefore, feature selection is essential in data preprocessing to enhance the machine learning model's performance.

Data transformation is also a critical task in data preprocessing, which involves converting the data into a suitable format for analysis. This process requires scaling, normalisation, and encoding categorical variables. Song et al. (2021) applied data transformation techniques to develop a fuzzy control model for PDE systems under cyberattacks. The study used a sampled-data-based event-triggered approach to reduce the effects of cyberattacks on the control system. Furthermore,

Acharya et al. (2022) employed data transformation techniques to improve the accuracy of a machine learning model for cyber insurance against cyberattacks on electric vehicle charging stations. The study used a one-hot encoding technique to transform categorical variables and a logarithmic transformation to normalise the data. These approaches significantly improved the performance of the model. Therefore, data transformation is a crucial step in data preprocessing to ensure that the data is in a suitable format for analysis.

### **3.3.1 Data Cleaning**

Data cleaning is essential in developing machine learning models for predicting cyberattacks in the space industry. It involves removing and resolving inconsistencies, errors, and missing data from the available dataset to improve the accuracy and reliability of the model. The data collected for this research is prone to cyberattacks, and hence, it is crucial to ensure the data is free from any cyberattack before feeding it into the model. There are 45, 79, and 32 features in the UNSW-NB15, CIC-IDS 2017 (Wednesday), and STIN datasets. In the UNSW-NB15 dataset, two features are the attack's class designations, and 43 are important features. While "label" is a binary class label, "attack cat" is a multi-class label. The term "attack cat" was eliminated because the proposed ML models are built to conduct binary classification for the UNSW-NB15 dataset and CICIDS2017. In the STIN dataset, one feature is class designations, and 31 are important features. While "label" is a multi-class label, the ML models are built for multi-classification for STIN.

### **3.3.2 Minority Removal**

Extremely unbalanced datasets might negatively impact machine learning performance. In the STIN dataset, four minority classes are merged into one DDoS class, and another three minority classes are merged into one Botnet class because the minority classes are a subset of the main

class, and the main target in this dataset set is satellite attacks as shown in the code in Appendix A. Finally, a balanced training dataset for STIN was produced that included Botnet, DDoS, Syn\_DDoS, and UDP\_DDoS with 40,401, 57,292, 54,789, and 57,082 records, respectively.

### **3.3.3 Encoding**

Encoding involves converting string or categorical features into numerical values, simplifying the data for machine learning. In this process, each category in a feature is represented as binary values (1 for the category present, 0 for others). The UNSW-NB15 datasets include categorical features that need encoding for KNN and SVM classifiers. However, encoding increases the dataset's dimensionality and the computational complexity of the training model. In the case of the UNSW-NB15 dataset, it initially had 45 features and 1 class label, but after encoding, the dimensionality increased to 193 features. The UNSW-NB15 dataset includes three categorical characteristics that each contain categorical values: “service”, “proto”, “state” and “attack\_cat”. Using a label encoder, these features were converted from string values to integers, but the STIN dataset has only one categorical characteristic in the “Label” feature.

### **3.3.4 Normalization**

Normalisation techniques are essential when dealing with datasets with a significant range of values to ensure each element has a uniform range. Normalisation prevents the model from favouring any individual data element. In this study, both z-score and log scaling were employed for normalisation. Z-score and logscale normalisation are methods to scale the data, and they aim to bring the data within specific ranges, typically between 0 and 1 or -1 and +1, as the code shown in Appendix A.

Z-score normalization: The method is standardised based on the original data's average value (mean) and standard deviation. The average data after processing is 0, and the square difference is 1, which meets the standard normal distribution. The primary purpose is to unify different data dimensions into the same order of magnitude and measure the calculated Z-score value uniformly to ensure comparability between data. The formula is as follows.

$$X'_i = \frac{X_i - X_\mu}{X_\sigma} \quad (13)$$

Where  $X_\mu$  and  $X_\sigma$  Are the mean and standard deviation for individual values of x, respectively.

$$X'_{i=\log(X_i+1)} \quad (14)$$

Where i ranges from 1 to n in any n-dimensional feature set.

Appendix A provides the R scripts for the initial dataset loading and pre-processing steps. These scripts facilitate the integration of multiple datasets and ensure that the data is appropriately formatted for subsequent processing, and scripts for data normalization, feature selection, and resampling outline the procedures for splitting the data into training, validation, and testing sets.

### 3.4 TRAINING AND TESTING SET PREPARATION

The training set is used to fit the model, and the test set is used to verify the model's performance. The UNSW-NB15 dataset has 82,332 records for the training set and 175,341 records for the testing set, but the STIN dataset has 209,564 records for the training set and 41,913 records for the testing set, as shown in the code in Appendix A.

### 3.5 FEATURE SELECTION

Feature selection is essential in developing a machine-learning model for predicting cyberattacks in satellite networks. Feature selection aims to identify the most relevant features in the data that can be used to train the model. This helps to reduce the dimensionality of the data, improve the model's accuracy, and reduce the risk of overfitting. Several methods have been proposed for feature selection in the context of cyberattacks, including filter, wrapper, and embedded methods.

In the context of the space industry, feature selection is essential due to the complexity and high dimensionality of the data. Oyama et al. (2021) developed a method for handling stealthy cyberattacks on evolving nonlinear process systems. They used filter and wrapper methods to select the most relevant features and improve the model's performance. Zografopoulos et al. (2021) assessed the security of integrated transmission and distribution power systems and analysed the impact of cyberattacks on these systems. They used a filter method to select the most relevant features and evaluate the model's performance.

In this study, features are manually selected for each of the datasets. The features relevant to the attacks are selected to achieve more accurate results and reduce time complexity.

For the STIN dataset, seven (7) features that contributed most to the attacks were selected manually: "syn\_cnt", "pkt\_len\_min", "pkt\_len\_max", "down\_up\_ratio", "fl\_dur", "bw\_win\_byt", and "l\_bw\_pkt". For the UNSWB-15 dataset, six (6) features that contributed most to the attacks were selected manually, which are "Sttl," "ct\_state\_ttl," "dload," "dmean," "dbytes", "dpkts". Moreover, for the CIC-IDS 2017 (Wednesday) dataset, Sixteen (16) features that contributed most to the attacks were selected manually, which are "Total.Length.of.Fwd.Packets", "Subflow.Fwd.Bytes", "Active. Max", "Flow.IAT.Min", "min\_seg\_size\_forward", "Init\_Win\_bytes\_forward", "Destination.Port", "Fwd.Packet.Length.Max", "Flow.IAT.Mean",



"Active.Std", "Fwd.Packet.Length.Mean", "Avg.Fwd.Segment.Size", "Active.Mean",  
"Fwd.Packet.Length.Std", "Down.Up.Ratio", "URG.Flag.Count".

### **3.6 EXPERIMENTAL PROCEDURE.**

The experiments used R programming (Version 4.2.3) and R Studio on a Desktop AMD 3400G with 3.70GHz, 8GB RAM, 222GB SSD, and Microsoft Windows 10 Pro x64. Decision tree, SVM, and KNN classifiers were implemented using the R programming packages rpart, C5.0, e1071, and class. The experiments focused on data normalisation, encoding, and model training time evaluation. This research compares the performance and training time complexity for CART, SVM, and KNN classifiers. The model implementation was tested on STIN, CIC-IDS2017, and UNSW-NB15 intrusion datasets, and each was split into training and evaluation sets. The hold-out method was used. Log-Scaling Normalization was applied to STIN and UNSW-NB15 datasets, while Z-score Normalization was used for CIC-IDS2017 due to its negative values. However, values obtained from the training dataset were used for the testing dataset to prevent normalisation bias. UNSW-NB15 contained string features, and encoding was implemented, expanding the dimensions of the datasets to 193.

It is essential to state that the Decision tree as a supervised ML does not make any prediction with an unknown pattern; in this experiment, UNSW-NB15 contains a feature in which some levels in the Testing phase were absent in the Training phase. For this reason, the state feature was removed during the model evaluation in this experiment since it contains some levels that are absent in the training phase. Finally, each model's training and running time complexity were measured and recorded by setting the timing. Finally, the performance evaluation metrics were calculated to assess the machine learning models. Figure 10 shows the flowchart representation of the experiment.

Appendix B provides the detailed code for training and evaluating the CART model, including hyperparameter tuning and performance assessment. The implementation details for the CART model, including code snippets for decision tree construction and evaluation, can be found in Appendix C. Appendix D includes the R scripts for training and evaluating the KNN model, emphasising distance metric selection and optimization.

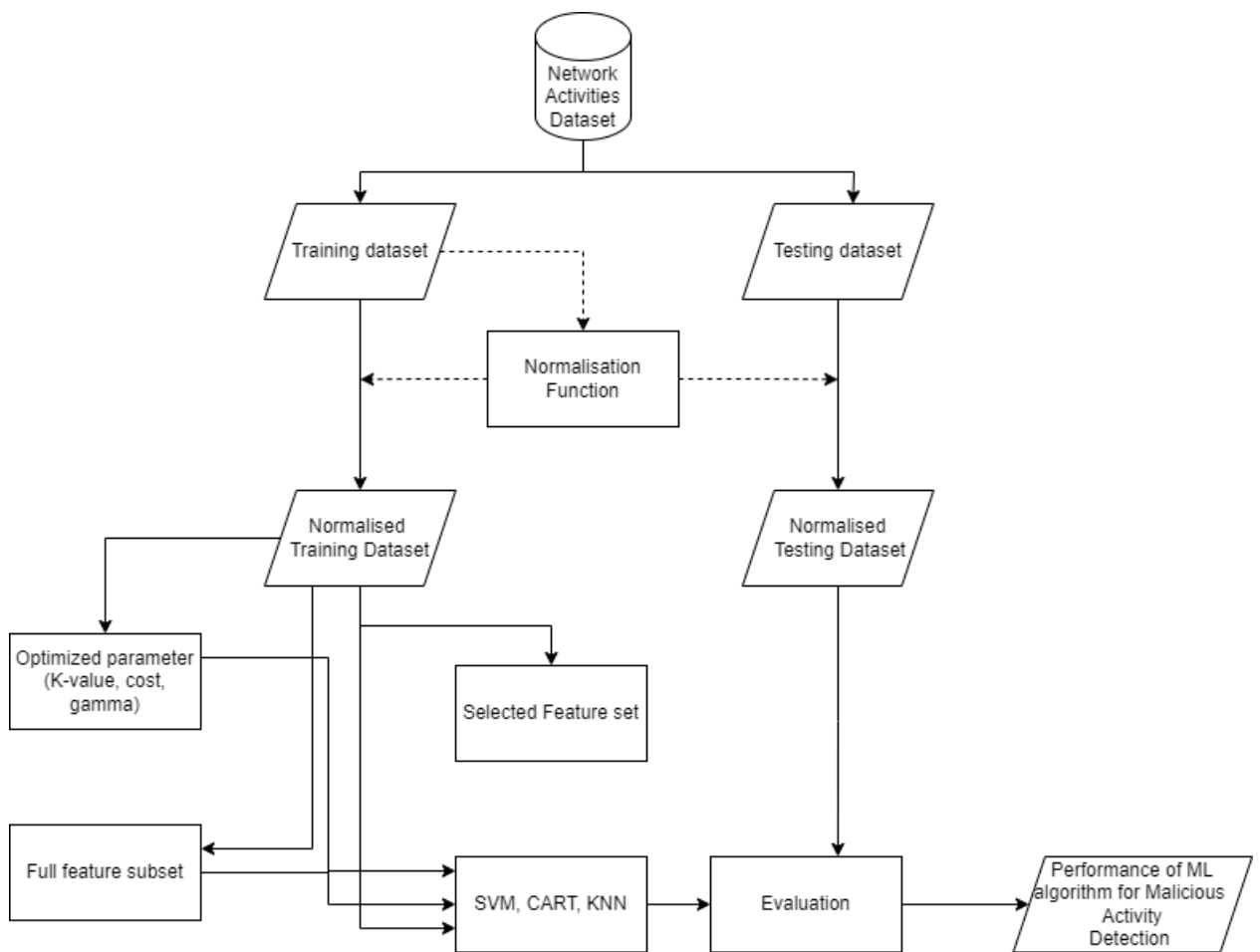


Figure 11: Experimental Flowchart.

### 3.7 MODEL EVALUATION

Model evaluation is a crucial aspect of machine learning, ensuring that the model developed is effective and efficient in predicting cyberattacks in the space industry. A well-developed machine

learning model is expected to accurately classify a cyberattack as a genuine or false alarm, with high precision and recall rates. The evaluation process involves assessing the model's performance using various metrics such as accuracy, precision, recall, and F1-score. Researchers have utilised different techniques to evaluate the efficacy of their models in predicting cyberattacks within the space industry.

The study by S. P. P et al. (2020) proposed a machine learning model for predicting cyberattacks in industrial control systems using enhanced principal component analysis and hypergraph-based convolution neural network (EPCA-HG-CNN). The study evaluated the model's efficacy using various metrics such as accuracy, precision, recall, F1-score, and area under the curve (AUC). The results showed that the proposed model outperformed existing models in terms of accuracy, precision, and recall rates. Additionally, the study showed that EPCA-HG-CNN has a high AUC score, indicating a reliable model for predicting cyberattacks in the space industry.

## CHAPTER FOUR

### RESULTS AND DISCUSSION

The research study investigates an intrusion detection system for predicting cyberattacks on space systems (Satellite Network). Intrusion, conversely, is the process of getting false privileges over the networks. Intrusion detection detects anomaly behaviour over a networking environment. As stated earlier, Intrusion detection for Satellite networks is critical because it is vital to global connectivity, safety, and reliable services. ML is considered here because it can learn from data without thorough programming. It uses three supervised machine learning algorithms: Random Forest (CART), KNN, and SVM. These algorithms are tested on UNSWB-15, CICIDS2017, and STIN. They will provide high security to satellite and terrestrial networks. The dataset was split into a train set and a test set in the ratios of 0.8 and 0.2, respectively. Each model is assessed using the accuracy, precision, recall, and F1 score evaluation metrics.

Feature selection is essential in ML problems as it increases the ML performance while reducing the complexity of the model. Investigating the feature selection property is critical as it is crucial in identifying features contributing to various attack types, which can also increase the time efficiency of the ML model. Categorizing various attacks in an IDS is vital as it enables the broad generalization of the ML model without redesigning itself. Decision trees, lazy learners (KNN) and Black-box (SVM) were investigated and compared. SVM is a supervised ML classifier with an efficient classification method, making it suitable for selecting relevant features with the help of its Kernel function. KNN is a lazy learner using the distance method to classify data. Each mentioned classifier will be implemented on the STIN dataset, the UNSW-NB dataset, and the CIC-IDS 2017 Wednesday. The STIN and UNSW-NB15 datasets were selected due to their broad uses for satellite network attacks. Likewise, CIC-IDS 2017 Wednesday was chosen since they both

contain DoS and DDoS attacks, which are essential for this study as DoS and DDoS are major attacks for satellite and space networks.

#### 4.1 PERFORMANCE METRICS

Assessing performance is paramount in machine learning when dealing with intrusion detection systems. The critical focus is optimizing True Negatives and minimizing False Positives to achieve high Accuracy, low false alarms, and high detection rates. Therefore, the evaluation of performance metrics involved measuring Accuracy, Recall, False Alarm Rate, and Precision, as the code is shown in Appendix E.

True Positive (TP): Truly classify of normal activities.

True Negative (TN): Truly classify of intrusion.

False Positive (FP): false classification of normal activities.

False Negative (FN): False classification of intrusion

Accuracy: The number of correct predictions that were correct.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (15)$$

Precision: The amount of correctly predicted intrusion.

$$Precision = \frac{TP}{TP+FP} \quad (16)$$

Recall: The number of relevant instances responsible for intrusion.

$$Recall = \frac{TP}{TP+FN} \quad (17)$$

False alarm rate: The number of false predictions of intrusion.

$$\text{False alarm rate} = \frac{FP}{FP+TN} \quad (18)$$

## 4.2 PERFORMANCE ANALYSIS

To prove the efficiency of the machine learning algorithms, the approaches were tested on three datasets: UNSW-NB15, CIC-IDS 2017(Wednesday), and STIN. Classifier performance evaluation is commonly computed using evaluation metrics such as accuracy, precision, recall, and the False alarm rate. These metrics are measured using a confusion matrix in Eqs. 15, 16, 17, and 18. To test the performance of individual attack labels on the STIN dataset, a confusion matrix for the Decision Tree (CART) and SVM multiclass classification model was computed with the code in Appendix E, as shown in Fig. 11, Fig. 12, Fig. 13, and Fig. 14. Figure 11 illustrates the confusion matrix for the STIN dataset's Decision Tree multiclass classification Model for the full feature set. The Decision Tree classifier performs well for terrestrial attacks, which include Botnet and DDoS categories. Still, it performs better for the UDP\_DDoS class than the syn\_DDoS class in satellite attacks, which include Syn\_DDoS and UDP\_DDoS attacks. Figure 12 illustrates the confusion matrix for the STIN dataset's Decision Tree multiclass classification Model for the reduced feature set. The Decision Tree classifier performs well for terrestrial attacks, which include Botnet and DDoS categories. Still, it performs better for the UDP\_DDoS class than the syn\_DDoS class in satellite attacks, which include Syn\_DDoS and UDP\_DDoS attacks. Figure 13 illustrates the confusion matrix for the STIN dataset's SVM multiclass classification Model for the full feature set. Based on the darker squares in the confusion matrix, the Decision Tree classifier performs well for terrestrial attacks, which include Botnet and DDoS categories. Still, it performs better for the UDP\_DDoS class than the syn\_DDoS class in satellite attacks, which include Syn\_DDoS and

UDP\_DDos attacks. Figure 14 illustrates the confusion matrix for the STIN dataset's SVM multiclass classification Model for the reduced feature sets.

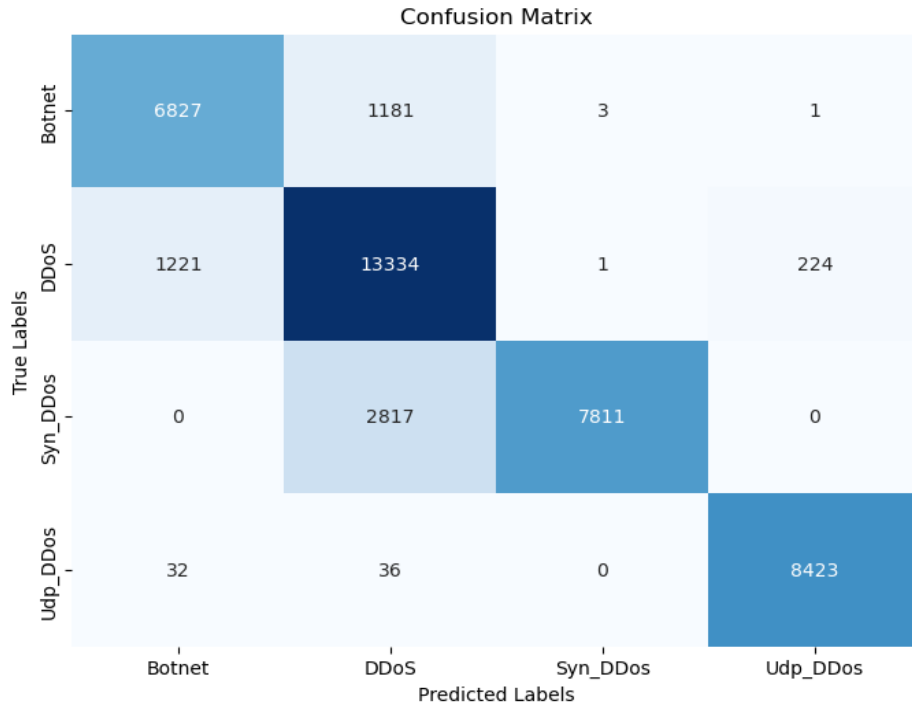


Figure 12: Confusion matrix of a full feature set for Decision Trees (STIN)

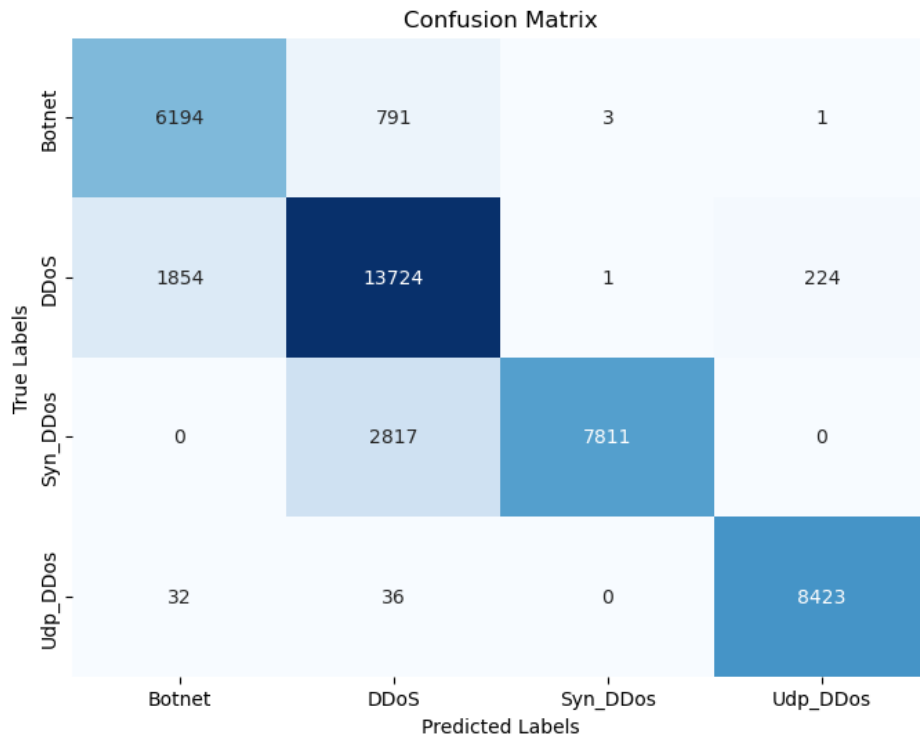


Figure 13: Confusion matrix of a Reduced feature set for Decision Trees (STIN)

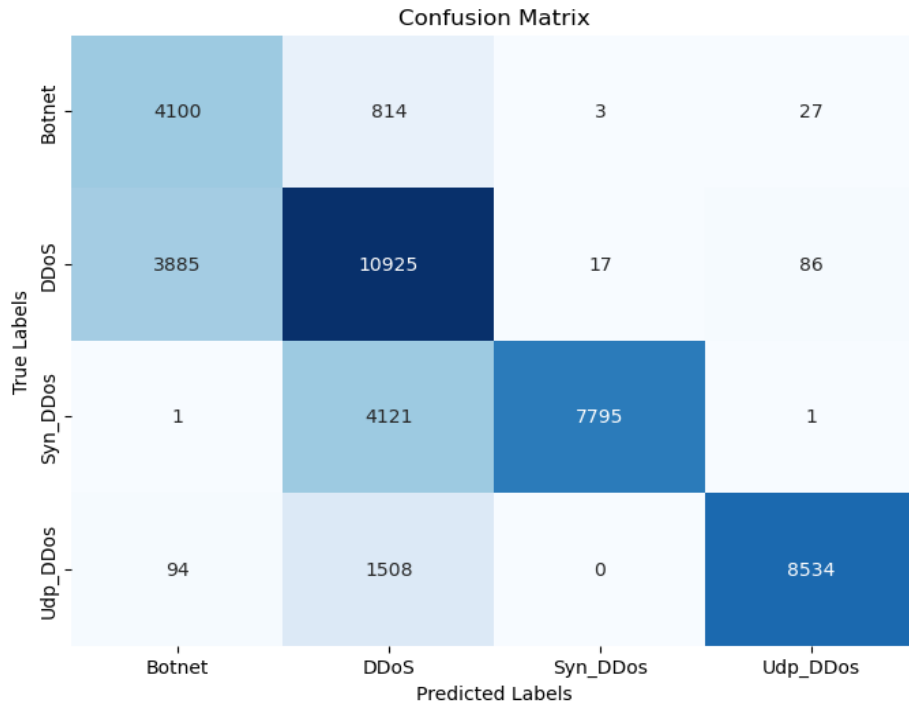


Figure 14: Confusion matrix of a Full feature set for SVM (STIN)

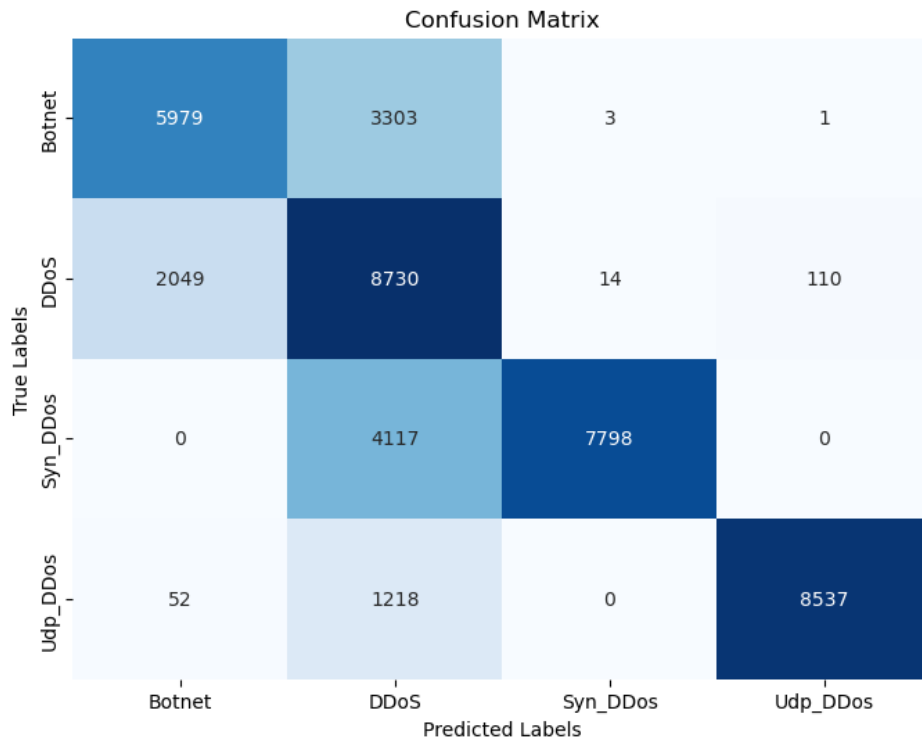


Figure 15: Confusion matrix of a Reduced feature set for SVM (STIN)



To test the performance of the UNSWB-15 dataset, This dataset has nine types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms, which are grouped into two individual attack labels named normal and abnormal attacks which the attack that is relevant to attack satellite network are classified as abnormal and others as normal, a confusion matrix for the Decision Tree (CART), SVM, and KNN classification model was computed, as shown in Fig. 15, Fig. 16, Fig. 17, Fig. 18, and Fig. 19. Figure 15 illustrates the confusion matrix for the UNSWB-15 dataset's Decision Tree classification Model for the full feature set. Figure 16 illustrates the confusion matrix for the UNSWB-15 dataset's Decision Tree classification Model for the reduced feature set. Figure 17 illustrates the confusion matrix for the UNSWB-15 dataset's SVM classification Model for the full feature set. Figure 18 illustrates the confusion matrix for the UNSWB-15 dataset's SVM classification Model for the reduced feature set by manually selecting the features for the decision tree. Figure 19 illustrates the confusion matrix for the UNSWB-15 dataset's KNN classification Model for the full feature set.

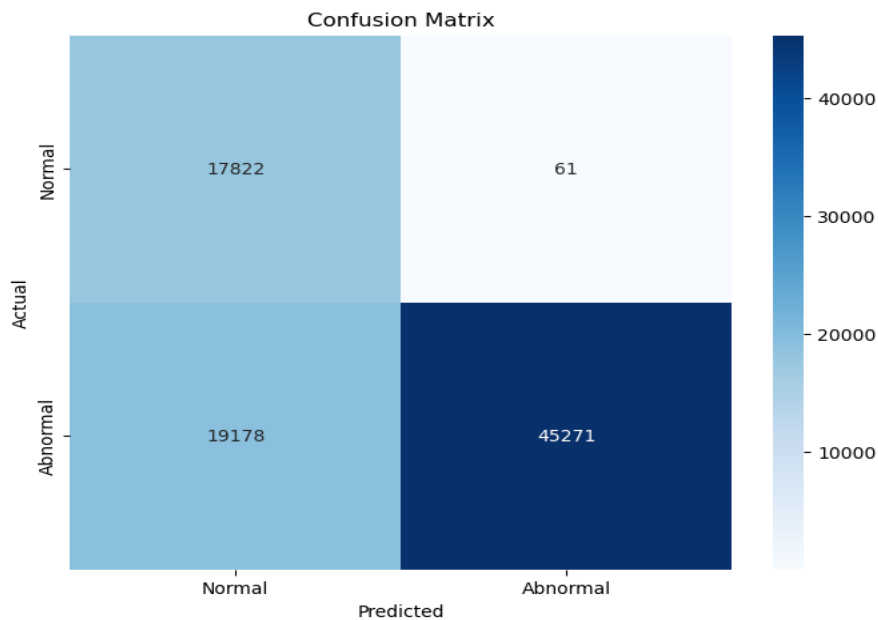


Figure 16: Confusion matrix of a Full feature set for Decision Tree (CART) for UNSWB-15

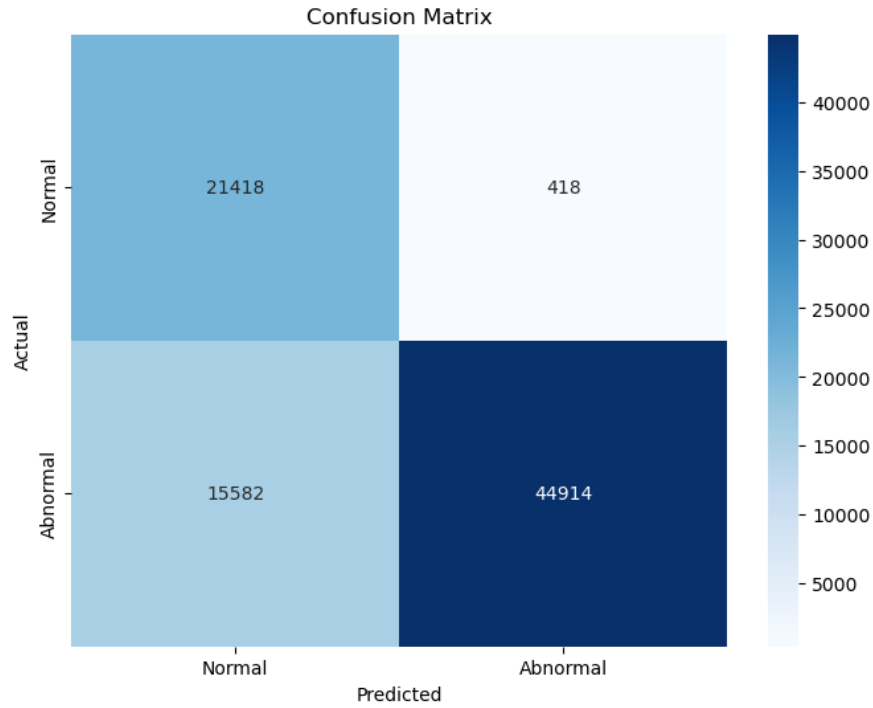


Figure 17: Confusion matrix of a Reduced feature set for Decision Tree (CART) for UNSWB-15

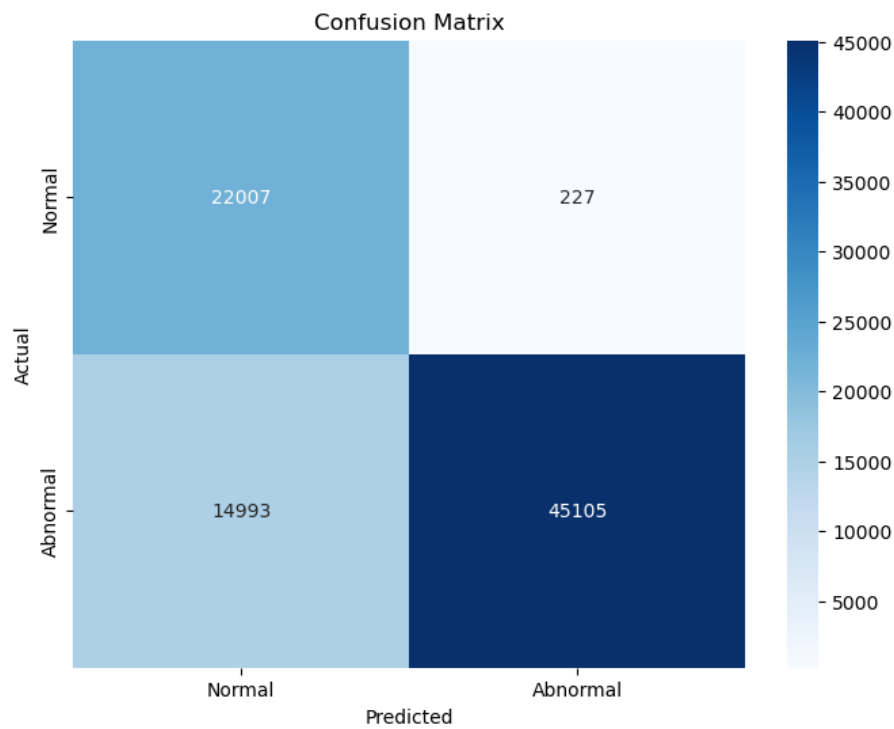


Figure 18: Confusion matrix of a Full feature set for SVM for UNSWB-15

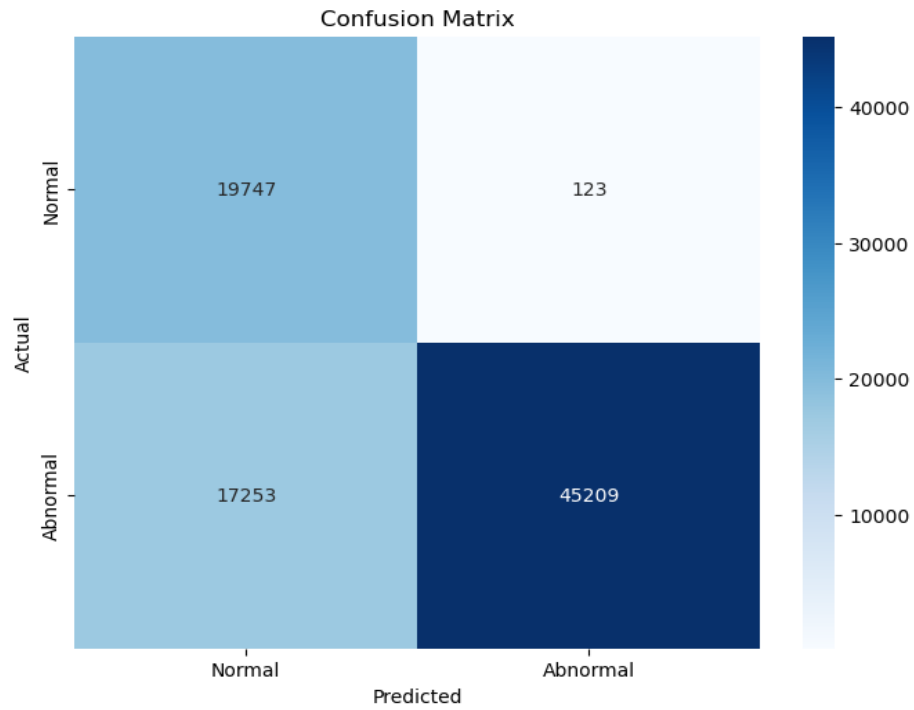


Figure 19: Confusion matrix of a Reduced feature set for SVM for UNSWB-15

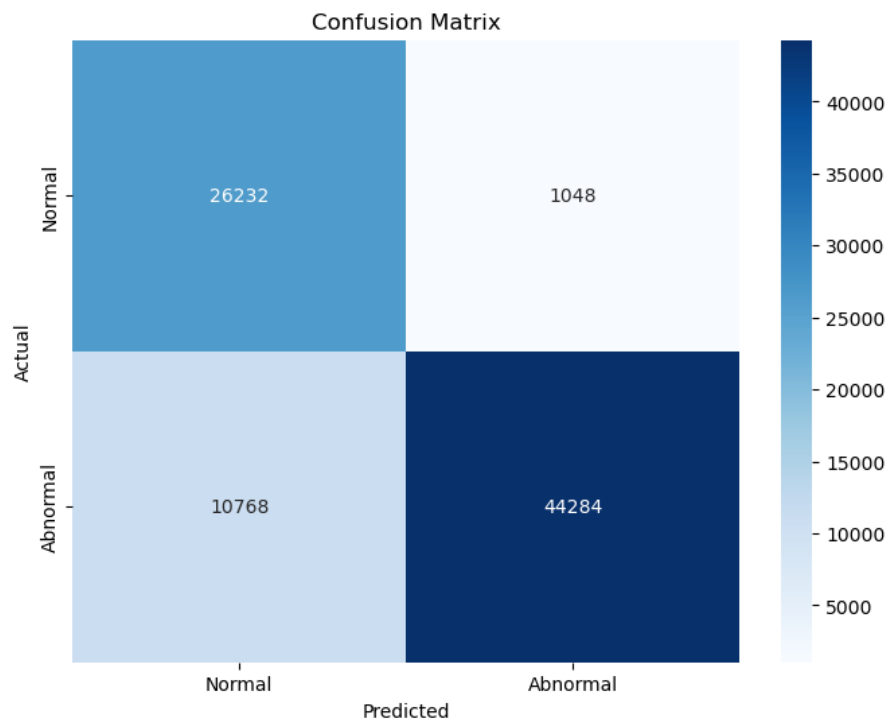


Figure 20: Confusion matrix of a Full feature set for KNN for UNSWB-15

To test the performance of the CIC-IDS2017 (Wednesday) dataset, the attack labels, which are like Five (5), are grouped into two individual attack labels, which are normal and abnormal attacks, a confusion matrix for the Decision Tree (CART) and SVM classification model was computed, as shown in Fig. 20, Fig. 21, Fig. 22, and Fig. 23. Figure 20 illustrates the confusion matrix for the CIC-IDS2017 (Wednesday) dataset's Decision Tree classification Model for the full feature set. Figure 21 illustrates the confusion matrix for the CIC-IDS2017 (Wednesday) dataset's Decision Tree multiclass classification Model for the reduced feature set. Figure 22 illustrates the confusion matrix for the CIC-IDS2017 (Wednesday) dataset's SVM classification Model for the full feature set. Figure 23 illustrates the confusion matrix for the CIC-IDS2017 (Wednesday) dataset's SVM classification Model for the reduced feature set.

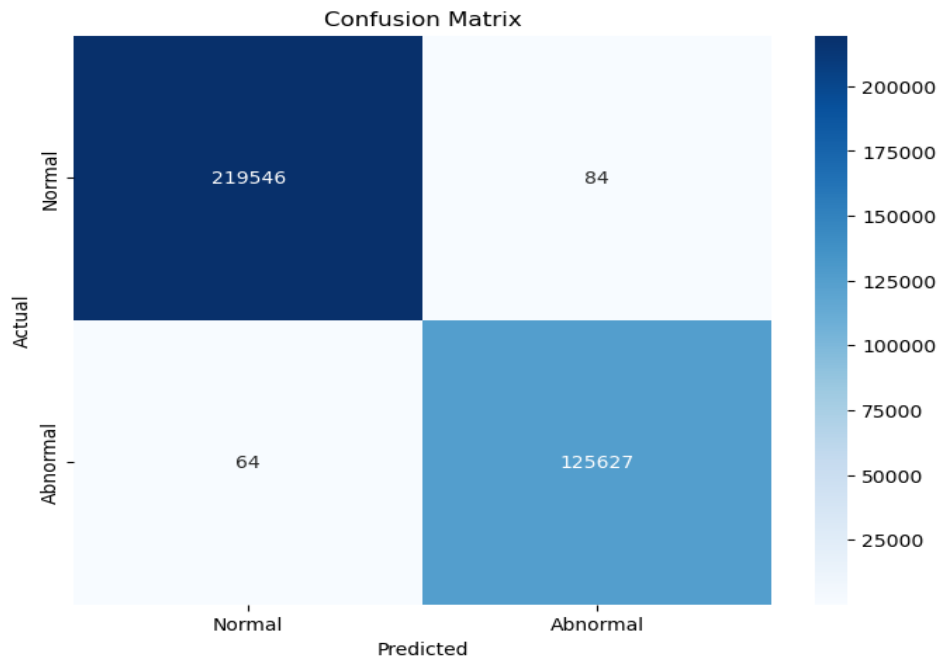


Figure 21: Confusion matrix of a Full feature set for Decision Tree (CART) for CICIDS2017-Wednesday

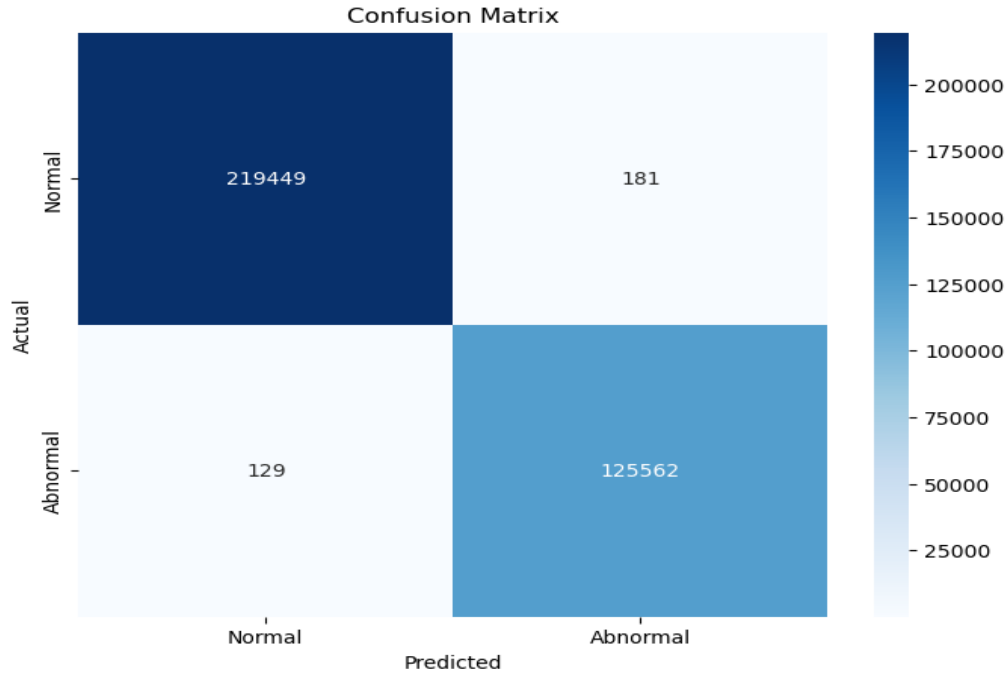


Figure 22: Confusion matrix of a Reduced feature set for Decision Tree (CART) for CICIDS2017-Wednesday

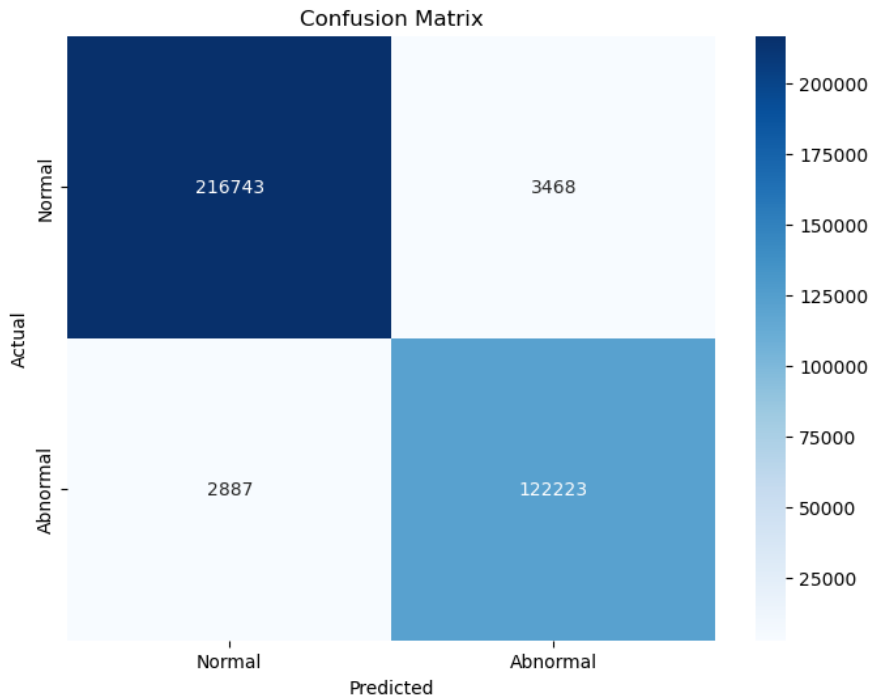


Figure 23: Confusion matrix of a Full feature set for SVM for CICIDS2017-Wednesday

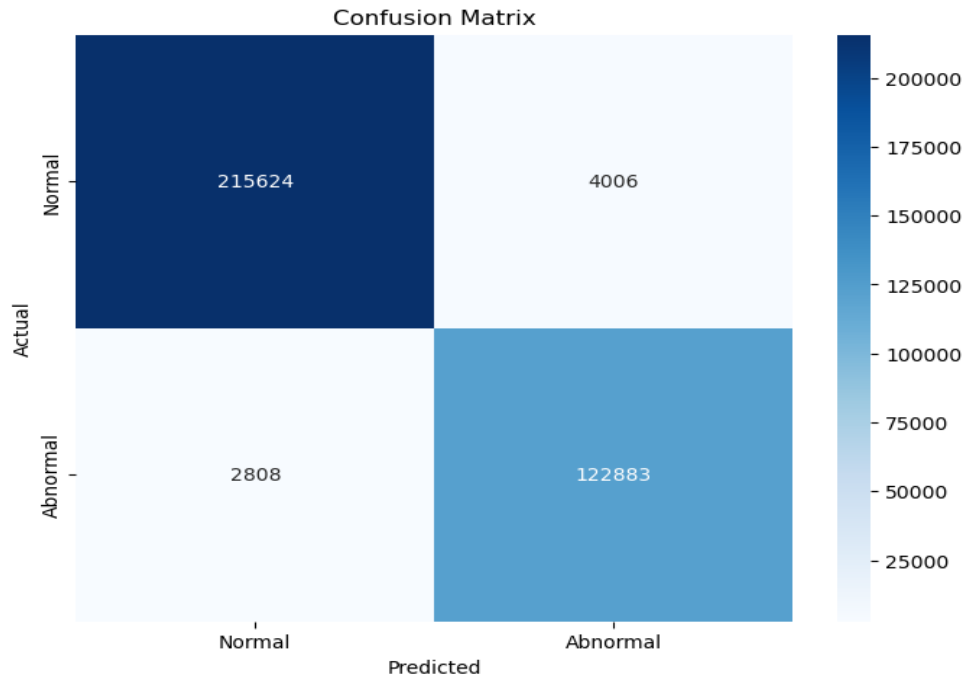


Figure 24: Confusion matrix of a Reduced feature set for SVM for CICIDS2017-Wednesday

### 4.3 COMPARISON OF CART DT WITH OTHER CLASSIFIERS.

This section investigates and compares the classifier performance of Different ML categories on the STIN, UNSW-NB15, and CIC-IDS 2017 Wednesday datasets. The Black Box (SVM) and the lazy learner KNN were compared with the CART decision tree classifier. Due to the distance approach in Lazy learner KNN, the classifier caused too many tie issues on the STIN and CIC-IDS 2017 Wednesday datasets. This limitation was encountered because of the experimental procedure, a Hold-out for this study. As a result, KNN was eliminated along the lines of this section. The K value used for KNN implementation in this study was the square root of the feature list, and the parameter used for SVM implementation was (kernel = radial and cost = 1/feature list).

This result demonstrates that SVM and CART are classification efficient on the Intrusion dataset used in this study. However, due to too many tie problems in KNN, the following experiment will not consider implementing KNN.

On the STIN dataset, CART achieved the highest accuracy of 93.42% compared with SVM, having 87.41% accuracy as shown in Table 5 and Table 7, and performed well for the reduced set, having 93.13% and 87.04% as the CART and SVM, respectively, as shown in Table 6 and Table 8. CART achieves the best precision and false alarm rate (FAR) on UNSWB-15, having 99.87% and 0.34%, respectively. Compared with SVM, it achieved 99.50%, 1.02% precision, and FAR, respectively, as shown in Table 5. Furthermore, on the CIC-IDS 2017 Wednesday full feature dataset, CART achieved the best performance across all evaluated metrics. CART achieved 99.87% accuracy, SVM achieved 98.48% accuracy, and for the 16 reduced feature lists of the CIC-IDS 2017 Wednesday dataset, as shown in Table 9 and Table 10, respectively, the SVM achieved the best performance across all evaluated metrics. Likewise, the time complexity of CART is more efficient than that of SVMs.

These results show that CART is time-efficient compared to SVM. These findings also reveal the reliability of the CART decision tree classifier’s time efficiency compared to other categories of supervised ML classifiers (Black Box, lazy learner, and Decision trees) Tables 11 & 12. As a result of the above experimental results, CART has demonstrated an excellent candidacy in developing a time-efficient ML model as an IDS for Satellite networking.

Table 5: Accuracy of Decision Tree classifier on STIN full feature set dataset.

Feature subset size	Decision Tree performance of the Satellite full feature set				
	Attack type	Accuracy	Precision	Recall	False Alarm Rate
31	Botnet	94.18%	84.49%	85.21%	3.70%
31	DDoS	86.93%	76.77%	90.22%	14.87%
31	Syn_DDoS	93.27%	99.95%	73.49%	0.01%
31	UDP_DDoS	99.30%	97.40%	99.20%	0.67%

Table 6: Accuracy of Decision Tree classifier on STIN reduced feature set dataset.

Feature subset size	Decision Tree performance of Satellite Reduced feature set				
	Attack Type	Accuracy	Precision	Recall	False Alarm Rate
8	Botnet	93.60%	76.66%	88.63%	5.40%
8	DDoS	86.35%	79.02%	86.84%	13.96%
8	Syn_DDoS	93.27%	99.95%	73.49%	0.01%
8	UDP_DDoS	99.30%	97.40%	99.20%	0.67%

Table 7: Accuracy of SVM classifier on STIN full feature set dataset.

Feature subset size	SVM performance of the Satellite full feature set				
	Attack type	Accuracy	Precision	Recall	False Alarm Rate
31	Botnet	88.49%	50.74%	82.93%	10.77%
31	DDoS	75.11%	62.90%	73.26%	23.87%
31	UDP_DDoS	90.12%	99.74%	65.41%	0.07%
31	Syn_DDoS	95.91%	98.68%	84.20%	0.36%

Table 8: Accuracy of SVM classifier on STIN reduced feature set dataset.

Feature subset size	SVM performance of Satellite Reduced feature set				
	Attack Type	Accuracy	Precision	Recall	False Alarm Rate
8	Botnet	87.10%	74.01%	64.39%	6.44%
8	DDoS	74.21%	50.27%	80.07%	27.86%
8	Syn_DDoS	90.14%	99.78%	65.45%	0.06%
8	UDP_DDoS	96.71%	98.72%	87.05%	0.35%



Table 9: Accuracy of classifiers on CIC-IDS 2017 Wednesday full feature set dataset.

Feature	Model performance of CIC-IDS 2017 Wednesday full feature set				
subset size	Algorithm	Accuracy	Precision	Recall	False Alarm Rate
68	KNN	—	—	—	—
68	SVM	98.48%	98.58%	97.27%	0.81%
68	CART	99.87%	99.84%	99.79%	0.09%

Table 10: Accuracy of classifiers on CIC-IDS 2017 Wednesday reduced feature set dataset.

Feature	Model performance of CIC-IDS 2017 Wednesday Reduced set				
subset size	Algorithm	Accuracy	Precision	Recall	False Alarm Rate
16	KNN	—	—	—	—
16	SVM	96.92%	96.69%	94.92%	1.91%
16	CART	95.80%	89.37%	98.96%	5.74%

Table 11: Accuracy of classifiers on UNSW-NB15 full feature set dataset.

Feature	Model performance of UNSW-NB15 full feature set				
subset size	Algorithm	Accuracy	Precision	Recall	False Alarm Rate
194	KNN	86.28%	96.71%	81.72%	5.20%
194	SVM	81.51%	99.50%	75.05%	1.02%
42	CART	76.63%	99.87%	70.24%	0.34%

Table 12: Accuracy of classifiers on UNSW-NB15 reduced feature set dataset.

Feature	Model performance of UNSW-NB15 Reduced feature set				
subset size	Algorithm	Accuracy	Precision	Recall	False Alarm Rate
6	KNN	—	—	—	—
6	SVM	78.90%	99.73%	72.38%	0.62%

6	CART	76.63%	99.87%	70.24%	0.34%
---	------	--------	--------	--------	-------

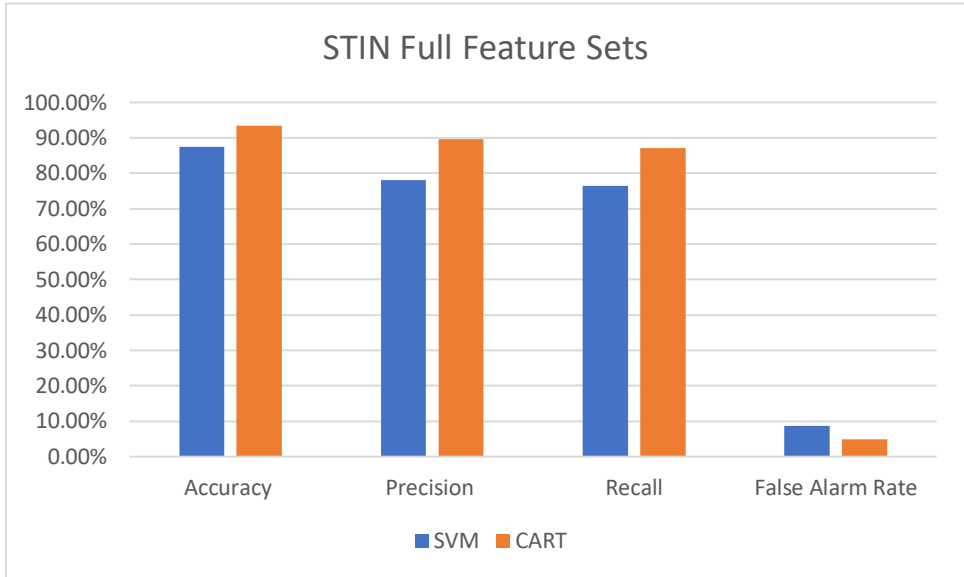


Figure 25: Performance comparison of all models on the STIN Full feature dataset

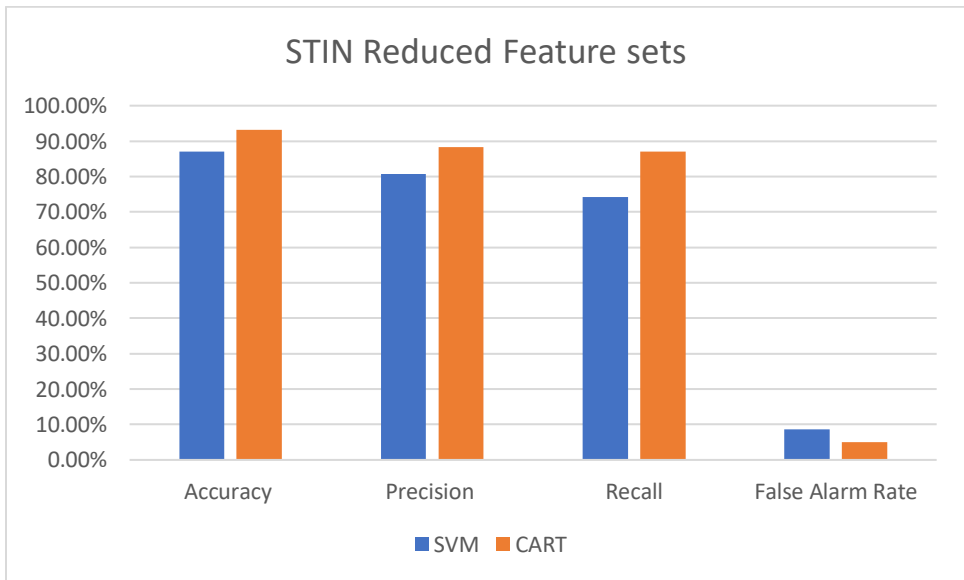


Figure 26: Performance comparison of all models on the STIN Reduced feature dataset

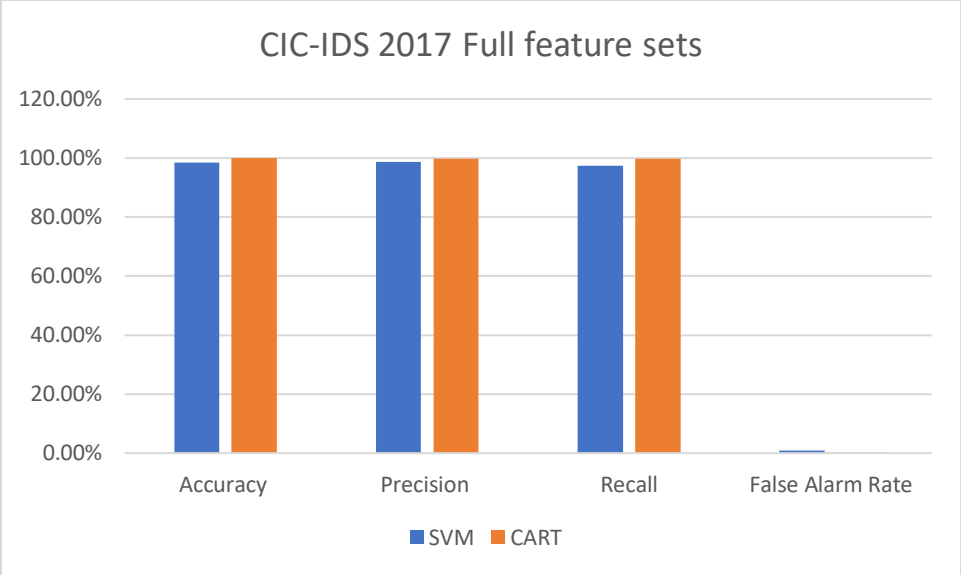


Figure 27: Performance comparison of all models on the CIC-IDS 2017 (Wednesday) Full feature dataset

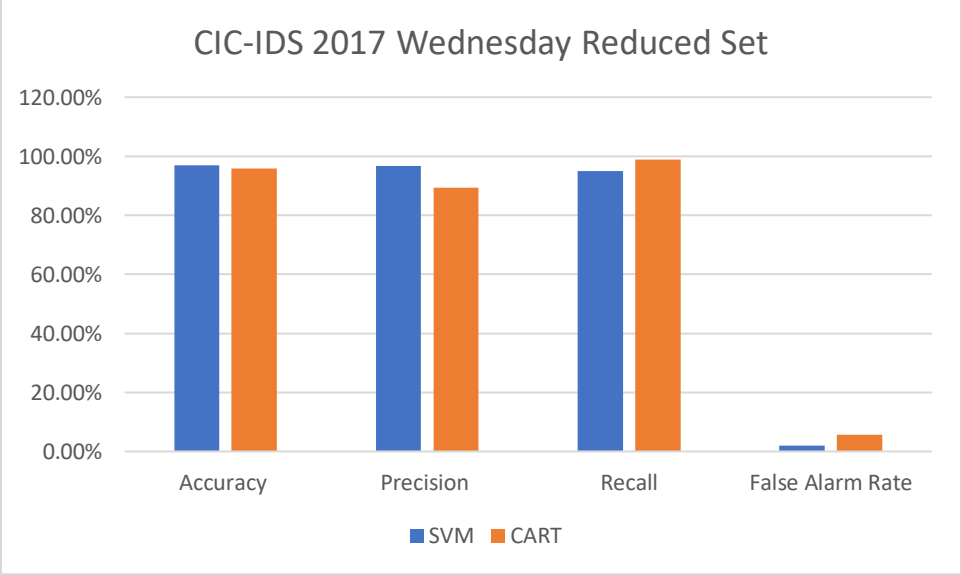


Figure 28: Performance comparison of all models on the CIC-IDS 2017 (Wednesday) Reduced feature dataset

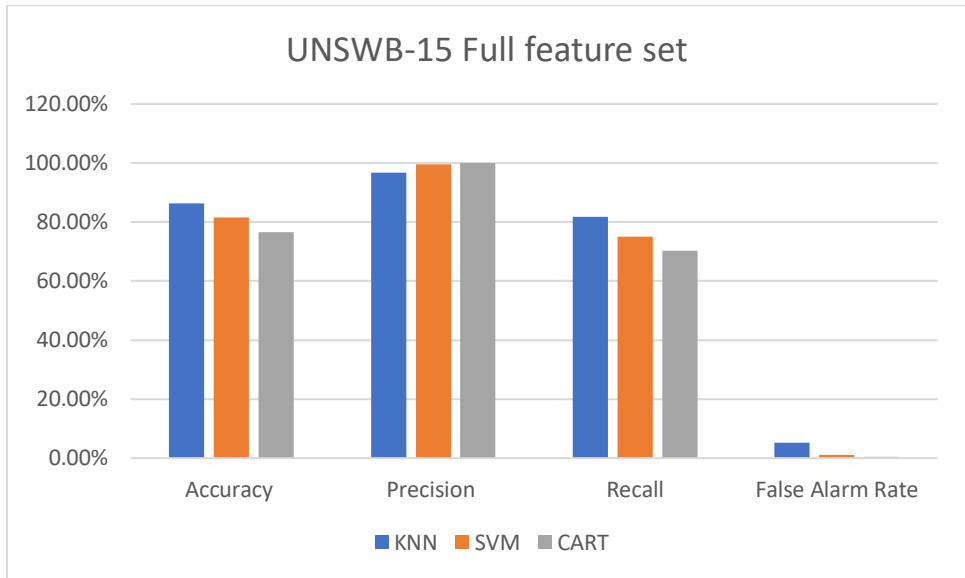


Figure 29: Performance comparison of all models on the UNSWB-15 Full feature dataset

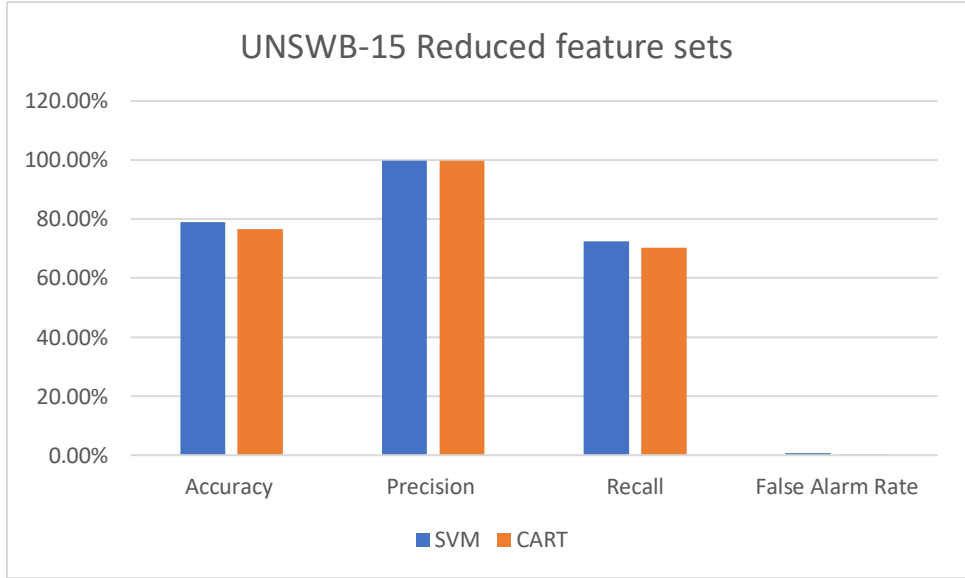


Figure 30: Performance comparison of all models on the UNSWB-15 Reduced feature dataset

**4.4. COMPUTATIONAL COMPLEXITY OF THE ALGORITHMS**

The computational complexity of the SVM model is estimated using the execution time on all three datasets, and results are given in Table 13 and Table 14. The execution time of the SVM is higher than that of the Decision Tree and KNN. Given that the model takes a slightly longer time

for training and testing, the performance of the SVM is significantly higher than that of the machine learning models.

Table 13: Estimated execution time of all classifiers on all full features of the three datasets.

Model	Dataset	Estimated Time
Decision Tree (CART)	STIN	68.31 secs
	CIC-IDS 2017 Wednesday	59.29 secs
	UNSWB-15	48.5 secs
SVM	STIN	559.51 secs
	CIC-IDS 2017 Wednesday	505.13 secs
	UNSWB-15	421.06 secs
KNN	STIN	—
	CIC-IDS 2017 Wednesday	—
	UNSWB-15	236.09 secs

Table 14: Estimated execution time of all classifiers on the reduced features set of the three datasets.

Model	Dataset	Estimated Time
Decision Tree (CART)	STIN	6.22 secs
	CIC-IDS 2017 Wednesday	4.96 secs
	UNSWB-15	3.71 secs
SVM	STIN	286.78 secs
	CIC-IDS 2017 Wednesday	209.45 secs
	UNSWB-15	176.05 secs
KNN	STIN	—

	CIC-IDS 2017 Wednesday	—
	UNSWB-15	—

#### 4.5 DISCUSSION OF RESULTS

This research experiment presented a performance analysis, comparison, and time complexity of the three ML algorithm techniques. The CART decision tree, KNN, and SVM techniques are exploited to significantly reduce the execution time of training and testing data by minimising the number of selected features with a manual selection of relevant features for the attack while enhancing the accuracy of intrusion detection. It selects 8, 6, and 16 features among 31 features, 194 features, and 68 features from the STIN, UNSWNB15, and CIC-IDS 2017 Wednesday datasets, respectively.

When applying to the STIN dataset, the accuracy of the Decision Tree classifier maintained the same accuracy with the reduced feature sets at 99.30%, as shown in table 5 and 6, and the accuracy of the SVM classifier increased from 95.91% to 96.71% as shown in table 7 and 8. The experimental results demonstrate that the CART Decision Tree classifier achieved a higher accuracy score in the UDP\_DDoS attack than the SVM classifier, as shown in Table 5. In addition, SVM classification recorded higher detection accuracy for Syn\_DDoS than Decision Tree, as shown in Table 7. When applied to the UNSW-NB15 dataset, the KNN achieved the best result for the full feature set, as shown in Table 11, but it did not work for the reduced feature set because of the number of ties. The accuracy of the Decision Tree classifier achieved the same result as the reduced feature sets with 76.63% accuracy, as shown in Tables 5 and 6, and the accuracy of the SVM classifier Reduced from 81.51% to 78.90%, as shown in Tables 7 and 8. Also, when applied to the CIC-IDS 2017 Wednesday dataset, the CART Decision Tree achieved the best result with

99.87% and almost the same accuracy as the Reduced feature sets, as shown in Table 9. The accuracy of the Decision Tree classifier achieved the same result as the reduced feature sets with 76.63% accuracy, as shown in Table 11, and the accuracy of the SVM classifier Reduced from 98.48% to 96.92%, as shown in Table 12. However, CART is this experiment's most time-efficient DT model, demonstrating the best overall results across all datasets. The support of the categorical features of the DT model contributes significantly to its lower complexity (Training time).

Finally, comparing the performance of KNN, SVM, and CART, the results show that SVM and CART Decision Trees demonstrate good classification performance across all evaluated matrices used in this study, as shown in Figure 25-30. As a result, CART achieved the best results in a short period compared with SVM, as shown in Tables 13 and 14.

## **4.6 OPERATIONAL PROCEDURE OF THE CODES**

This section provides a step-by-step guide on how to set up and run the R codes used for data collection, pre-processing, and model evaluation. The objective is to ensure that future readers can reproduce the results and conduct further experiments without additional assistance.

### **4.6.1 Prerequisites**

Before running the codes, ensure you have the following prerequisites installed on your system: R, RStudio, and Install the necessary R packages by running the following commands in the R console.

- a) Download and install R from CRAN.
- b) Download and install RStudio, an integrated development environment (IDE) for R, from RStudio's website.
- c) `install.packages(c("dplyr", "ggplot2", "caret", "e1071", "rpart", "class", "reshape2"))`

## 4.6.2 Setting Up the Project

Created the project files, including the datasets and R scripts

Create a new directory on your computer and organize the files as follows:

MSC/

```
|— data/
|   |— UNSWB15.csv
|   |— STIN.csv
|   └— CICIDS2017(Wednesday).csv
|— scripts/
|   |— data_preprocessing.R
|   |— svm_model.R
|   |— cart_model.R
|   └— knn_model.R
|   └— eval_metrics.R
|— results/
|   └— (empty, to store output files)
```

## 4.6.3 Running the Codes

Follow these steps to run each script and perform the tasks outlined in the thesis:



For data pre-processing, open `data_preprocessing.R` located in the `scripts/` directory. Run the script to clean, normalize, and preprocess the data. The preprocessed data will be saved for use in model training with this command `“source(scripts/data_preprocessing.R)”`

For training and evaluating the three models for each dataset, open `svm_model.R`, `cart_model.R`, and `knn_model.R` are located in the `scripts/` directory. Run the script to train and evaluate all three models with these scripts: `“source("scripts/svm_model.R")”`, `“source("scripts/cart_model.R”)”`, and `“source("scripts/knn_model.R”)”`. The script outputs performance metrics and saves the model to the `results/` directory.

#### **4.6.4 Interpreting the Results**

After running each model script, navigate to the `results/` directory to review the output files, which include performance metrics and model summaries. Use these outputs to compare model performance and draw conclusions, as the thesis discusses.

## CHAPTER FIVE

### SUMMARY OF FINDINGS, CONCLUSION AND RECOMMENDATIONS

#### 5.1 SUMMARY OF FINDINGS

This study employed three datasets, UNSWB15, STIN, and CIC-IDS2017 (Wednesday), for intrusion detection systems (IDS) to predict cyberattacks on satellite networks. It also compared the classification performance of three categories of ML classifiers: Support Vector Machine (SVM), Decision Tree (CART), and K-nearest neighbours (KNN) and manual selection of feature importance to reduce the time complexity and performance. Support Vector Machine (SVM) demonstrated high accuracy and robustness in distinguishing between benign and malicious activities, especially in scenarios with complex decision boundaries. SVM showed superior precision and recall, making it highly effective in minimizing false positives and false negatives, which show 99.87% and 70.24%, respectively, for UNSWB-15 as shown in Table 11 and 99.84% and 99.79%, respectively, for CIC-IDS 2017 (Wednesday) as shown in table 9. The decision tree (CART) showed a good accuracy performance of 99.87% for CIC-IDS 2017 (Wednesday) and 93.42% for STIN, as shown in Tables 9 and 5, respectively, because of its internal feature selection features. KNN achieved commendable detection accuracy. It performed well in recognizing patterns in the data, but its high computational cost, causing too many ties, and slower prediction time were noted as significant drawbacks. The study employed manual feature selection techniques to reduce dimensionality and improve computational efficiency, leading to more accurate and reliable predictions. The models achieved higher accuracy and better overall performance by identifying and utilising the most relevant features.

Each dataset presented unique challenges and opportunities for the models. The UNSWB15 dataset was instrumental in testing the models' ability to handle diverse types of cyberattacks. The STIN

and CICIDS2017 (Wednesday) datasets, which contain accurate network data, provided valuable insights into the models' performance in different network environments and attack scenarios.

## **5.2 RESEARCH LIMITATIONS**

The datasets (UNSWB15, STIN, and CICIDS2017) do not represent all possible types of cyberattacks or the diversity of satellite network configurations. Real-world satellite networks can have unique characteristics and threat landscapes not fully captured in these datasets. The models were trained and tested on specific datasets, and their performance may vary when applied to different satellite networks or newer types of cyberattacks not present in the datasets. So, a real-time satellite dataset is necessary. Satellite networks are highly specialized and can vary widely in architecture and usage. The models developed in this study may need further customization and adaptation to be effectively deployed in various real-world satellite network environments.

The study did not extensively address potential security and privacy concerns related to deploying machine learning models in satellite networks. Ensuring the models themselves are secure from adversarial attacks and protecting the privacy of the data used are critical aspects that require further investigation. In conclusion, while this research provides an understanding of the application of machine learning models for IDS in satellite networks, addressing these limitations in future work is essential to enhancing the solutions' practical applicability and robustness.

## **5.3 CONCLUSION**

The requirement for cybersecurity solutions to prevent attacks in the modern network environment has increased along with the number of network intrusion attacks on satellites because of the importance of weather forecasting and communication. The increase in network intrusion attacks has also increased the need for an intuitive cybersecurity system to cope with the attacks in the modern network environment. Three ML-based IDSs are used in this research to provide a

high level of security for both satellite and terrestrial networks. The experiment of this study provides insight into the comparison of the three categories of ML Classifiers in which Decision tree performance is more accurate than the others due to its Multiclass classification, internal feature selection, various optimised versions and support of categorical features. Knowing that feature selection is significant when implementing an ML model for IDS; this study compared the accuracy and time complexity by manually selecting the important features for the attacks. The machine learning algorithm is evaluated and verified using the UNSW-NB15, CIC-IDS 2017 Wednesday, and STIN datasets. The feature selection technique improves the classification results and reduces the execution time.

In conclusion, this experiment provides valuable insight into how ML researchers can visualise the theoretical properties mentioned above in the scope of intrusion detection systems.

#### **5.4 RECOMMENDATIONS**

By demonstrating the effectiveness of models like Support Vector Machine (SVM), Decision Tree (CART), and K-Nearest Neighbors (KNN), the research provides a foundation for further exploration and development of advanced IDS technologies. The findings' practical implementation can lead to deploying more sophisticated IDS solutions in real-world satellite networks, thereby improving their resilience against cyber threats and reducing the risk of service disruptions. Investing in hardware and software infrastructure that supports real-time data processing and analysis is critical. Optimizing the computational efficiency of models, particularly for algorithms like KNN, can help achieve timely threat detection. The simplicity and low resource requirements of this study's three supervised machine learning algorithms make them highly suitable for satellite networks.

Promoting collaboration and knowledge exchange between government, business, and academia can lead to more creative and reliable IDS solutions. Establishing forums and working groups devoted to satellite network security can facilitate this communication. Engaging in cybersecurity contests and challenges can spur innovation in IDS technologies and yield insightful information.

## **5.5 SUGGESTIONS FOR FUTURE RESEARCH**

One future work of this study can be to implement a well-feature selection technique such as wrappers and embedded, which can improve the generalization of this model by optimizing the classifiers as there is still room for improvement across all the datasets used in this work. Another future work of this study is an empirical analysis of deep learning such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variants, which have shown promise in other cybersecurity domains. These models can capture more complex patterns and temporal dependencies in network traffic data.

Conduct extensive testing and validation of IDS models in realistic satellite network environments using simulated and real-world data. Collaboration with satellite network operators can provide valuable insights and data for more accurate evaluation. By pursuing these research directions, future studies can address the current limitations and push the boundaries of what is possible in protecting satellite networks from cyber threats.

## APPENDIX A: DATA PREPROCESSING

```
# Combine four minority classes into one 'DDoS' class
combined_data$Label[combined_data$Label %in% c("Syn_DDoS", "UDP_DDoS",
"LDAP_DDoS", "MSSQL_DDoS")] <- "DDoS"

# Combine another three minority classes into one 'Botnet' class
combined_data$Label[combined_data$Label %in% c("Web Attack", "Backdoor",
"NetBIOS_DDoS")] <- "Botnet"

combined_data <- rbind(TestDatasetNEG, Train)
combined_data <- na.omit(combined_data)

library(caret)

# Set seed for reproducibility
set.seed(123)

# Create an index for splitting the data (80% training, 20% testing)
splitIndex <- createDataPartition(combined_data$Label, p = 0.8, list = FALSE)

# Create training and testing sets
training_data <- combined_data[splitIndex, ]
testing_data <- combined_data[-splitIndex, ]

# Check the dimensions of the datasets
dim(training_data)
dim(testing_data)

Testdataset = Testdataset[,c(-15)]
Traindataset = Traindataset[,c(-15)]
```

```
save(TestDatasetNorm, file = "SateliteTestzscore.Rda")
```

```
save(TrainDatasetNorm, file = "SateliteTrainzscore.Rda")
```

```
#-----zscore
```

```
TrainDatasetNorm <- Traindataset
```

```
TrainDatasetNoNorm <- Traindataset
```

```
TestDatasetNorm <- Testdataset
```

```
TrainDatasetMeanList <- vector(length = (ncol(Traindataset)-1))
```

```
TrainDatasetSdList <- vector(length = (ncol(Traindataset)-1))
```

```
for(j in 1:ncol(Traindataset))
```

```
{
```

```
  TrainDatasetMeanList[j] <- mean(TrainDatasetNoNorm[,j])
```

```
  TrainDatasetSdList[j] <- sd(TrainDatasetNoNorm[,j])
```

```
  for(i in 1:nrow(Traindataset))
```

```
  {
```

```
    TrainDatasetNorm[i,j] <- ((Traindataset[i,j] - TrainDatasetMeanList[i]) /  
(TrainDatasetSdList[i]))
```

```
  }
```

```
}
```

### **Logscalling:**

```
columns_to_log2 <- c("dur", "spkts",  
"dpkts", "sbytes", "dbytes", "rate", "sttl", "dttl", "sload", "dload", "sloss",  
"dloss", "sinpkt", "dinpkt", "sjit", "djit", "swin", "stcpb",
```

```

        "dtepb", "dwin", "tcprtt", "synack", "ackdat", "smean", "dmean",
        "trans_depth", "response_body_len",
"ct_srv_src", "ct_state_ttl", "ct_dst_ltm", "ct_src_dport_ltm", "ct_dst_sport_ltm",

"ct_dst_src_ltm", "is_ftp_login", "ct_ftp_cmd", "ct_flw_http_mthd", "ct_src_ltm", "ct_srv_dst"
)

```

```

for (column in columns_to_log2) {
  TestDataset[[column]] <- log(TestDataset[[column]] + 1)
}

```

```

columns_to_log1 <- c(
"dur", "spkts", "dpkts", "sbytes", "dbytes", "rate", "sttl", "dttl", "sload", "dload", "sloss",
  "dloss", "sinpkt", "dinpkt", "sjit", "djit", "swin", "stcpb",
  "dtepb", "dwin", "tcprtt", "synack", "ackdat", "smean", "dmean",
  "trans_depth", "response_body_len",
"ct_srv_src", "ct_state_ttl", "ct_dst_ltm", "ct_src_dport_ltm", "ct_dst_sport_ltm",

"ct_dst_src_ltm", "is_ftp_login", "ct_ftp_cmd", "ct_flw_http_mthd", "ct_src_ltm", "ct_srv_dst"
)

```

```

for (column in columns_to_log1) {
  TrainDataset[[column]] <- log(TrainDataset[[column]] + 1)
}

```



## APPENDIX B: DECISION TREE (CART) MODEL IMPLEMENTATION

```
library(rpart)
FsTrain$Label = as.factor(FsTrain$Label )
start.time <- Sys.time()
Train_Cart = rpart( Label ~ ., data=FsTrain, method = "class" )

end.time <- Sys.time()
time.taken <- round(end.time - start.time,2)
time.taken

start.time <- Sys.time()

predicted.classes <- Train_Cart %>%
  predict(FsTest, type = "class")

end.time <- Sys.time()
time.taken <- round(end.time - start.time,2)
time.taken

table(pred = predicted.classes,FsTest$Label )
```

## APPENDIX C: SVM MODEL IMPLEMENTATION

```
library(e1071)
start.time <- Sys.time()
svm_model <- svm(x = FsTrain[,-7], y = FsTrain[,7], type = "C-classification", kernel = "radial",
cost = 1, gamma = 1/6)

end.time <- Sys.time()
time.taken <- round(end.time - start.time,2)
time.taken

#-----svm Prediction on Train
start.time <- Sys.time()

pred_train <- predict(svm_model,FsTest[,-7])

end.time <- Sys.time()
time.taken <- round(end.time - start.time,2)
time. taken

table(pred = pred_train , FsTest[,7])
```

## APPENDIX D: KNN MODEL IMPLEMENTATION

```
library(class)
FsTrain$label = as.factor(FsTrain$label)

start.time <- Sys.time()

pred <- knn(FsTrain[, -26],FsTest[,-26],FsTrain$label , k=13)

end.time <- Sys.time()
time.taken <- round(end.time - start.time,2)
time.taken

table(pred = pred,FsTest$label)
```

## APPENDIX E: EVALUATION METRICS IMPLEMENTATION

```
# Load necessary libraries
library(ggplot2)
library(reshape2)

# Define the confusion matrix
conf_matrix <- matrix(c(TN, FP, FN, TP), nrow=2, byrow=TRUE)

# Assign names to the rows and columns for readability
rownames(conf_matrix) <- c("Actual Normal", "Actual Abnormal")
colnames(conf_matrix) <- c("Predicted Normal", "Predicted Abnormal")

# Calculate metrics
TN <- conf_matrix[1, 1]
FP <- conf_matrix[1, 2]
FN <- conf_matrix[2, 1]
TP <- conf_matrix[2, 2]

# Calculate accuracy
accuracy <- round((TP + TN) / (TP + TN + FP + FN) * 100, 2)

# Calculate precision
precision <- round(TP / (TP + FP) * 100, 2)

# Calculate recall
recall <- round(TP / (TP + FN) * 100, 2)

# Calculate false alarm rate
false_alarm_rate <- round(FP / (FP + TN) * 100, 2)
```

```
# Print metrics
cat("Accuracy:", accuracy, "%\n")
cat("Precision:", precision, "%\n")
cat("Recall:", recall, "%\n")
cat("False Alarm Rate:", false_alarm_rate, "%\n")

# Plot heatmap
conf_matrix_melt <- melt(conf_matrix)
colnames(conf_matrix_melt) <- c("Actual", "Predicted", "Count")

ggplot(data = conf_matrix_melt, aes(x = Predicted, y = Actual, fill = Count)) +
  geom_tile(color = "white") +
  geom_text(aes(label = Count), vjust = 1) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(title = "Confusion Matrix", x = "Predicted", y = "Actual") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

## REFERENCES

- Abaimov, S., & Martellini, M. (2022). Understanding Machine Learning. In *Machine Learning for Cyber Agents* (pp. 15–89). Springer International Publishing.
- Acharya, S., Mieth, R., Konstantinou, C., Karri, R., & Dvorkin, Y. (2022, March). Cyber Insurance Against Cyberattacks on Electric Vehicle Charging Stations. *IEEE Transactions on Smart Grid*, 13(2), 1529–1541. <https://doi.org/10.1109/tsg.2021.3133536>.
- Aerospace Corporation, "Protecting Space Systems from Cyber Attack". <https://medium.com/the-aerospace-corporation/protecting-space-systems-from-cyber-attack-3db773aff368>. Accessed: 2022-03-31.
- Ahmed, M., Byreddy, S., Nutakki, A., Sikos, L. F., & Haskell-Dowland, P. (2021, November). ECU-IoHT: A dataset for analysing cyberattacks in the Internet of Health Things. *Ad Hoc Networks*, 122, 102621. <https://doi.org/10.1016/j.adhoc.2021.102621> \*
- Ahmed, L.A.H., Hamad, Y.A.M.: Machine learning techniques for network-based intrusion detection system: a survey paper. In: National Computing Colleges Conference (NCCC). IEEE, 2021.
- Ahsan, M., Nygard, K. E., Gomes, R., Chowdhury, M. M., Rifat, N., & Connolly, J. F. (2022). Cybersecurity threats and their mitigation approaches using machine learning—A review. *Journal of Cybersecurity and Privacy*, 2(3), 527–555. <https://doi.org/10.3390/jcp2030027>.
- Alvarez, J., & Walls, B. (2016). Constellations, clusters, and communication technology: Expanding small satellite access to space. *2016 IEEE Aerospace Conference*.
- Almomani, O. (2020). A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. *Symmetry*, 12(6), 1046. <https://doi.org/10.3390/sym12061046>.
- Alrashdi, I., Alqazzaz, A., Aloufi, E., Alharthi, R., Zohdy, M., & Ming, H. (2019). *AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning*. <https://doi.org/10.1109/ccwc.2019.8666450>.
- Amir, Feizi., Ali, Nazemi. (2022). Classifying random variables based on support vector machine and a neural network scheme. *Journal of Experimental & Theoretical Artificial Intelligence*.36(5):1-24. doi: 10.1080/0952813x.2022.2104385.
- Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A. V., & Di Franco, F. (2023). The role of machine learning in cybersecurity. *Digital Threats: Research and Practice*, 4(1), 1–38. <https://doi.org/10.1145/3545574>.

- Ashraf, I., Narra, M., Umer, M., Majeed, R., Sadiq, S., Javaid, F., Rasool, N.: A deep learning based smart framework for cyber-physical and satellite system security threats detection. *Electronics* 11(4), 667 (2022).
- Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, 12(6), 1333. <https://doi.org/10.3390/electronics12061333>.
- Awuor, O. G. (2023). Assessment of existing cyber-attack detection models for web-based systems. *Global Journal of Engineering and Technology Advances*, 15(1), 070–089. <https://doi.org/10.30574/gjeta.2023.15.1.0075>.
- Azar, A. T., Shehab, E., Mattar, A. M., Hameed, I. A., & Elsaid, S. A. (2023). Deep Learning Based Hybrid Intrusion Detection Systems to Protect Satellite Networks. *Journal of Network and Systems Management*, 31(4). <https://doi.org/10.1007/s10922-023-09767-8>.
- B. Bailey, Establishing space cybersecurity policy, standards, and risk management practices. Aerospace Corporation El Segundo, CA, 2020.
- B. Ingre and A. Yadav, “Performance analysis of NSL-KDD dataset using ANN,” in 2015 international conference on signal processing and communication engineering systems, Guntur, India, 2015.
- Berry, H. S. (2023). The importance of cybersecurity in supply chain. *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*.
- Bongers, A., & Torres, J. L. (2023). Star wars: Anti-satellite weapons and orbital debris. *Defence and Peace Economics*, 1–20. <https://doi.org/10.1080/10242694.2023.2208020>
- Bunn, M. (2023). Insider threats to nuclear security. In *The Oxford Handbook of Nuclear Security*. Oxford University Press.
- Binitta, Sunny., Leema, G. (2022). Comparative Study Between KNN & SVM. *International Journal of Advanced Research in Science, Communication and Technology*, doi: 10.48175/ijarsct-4908.
- Cai, J., Song, S., Zhang, H., Song, R., Zhang, B., & Zheng, X. (2023). Satellite network traffic prediction based on LSTM and GAN. *2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*.
- Cai, Jie, Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300, 70–79. <https://doi.org/10.1016/j.neucom.2017.11.077>.
- Chandrashekar, G., Sahin, F.: A survey on feature selection methods. *Comput. Electr. Eng.* 40(1), 16–28 (2014).

- Chen, R.-C., Dewi, C., Huang, S.-W., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, 7(1). <https://doi.org/10.1186/s40537-020-00327-4>.
- Chi, Y., Cheng, Y., & Ji, X. (2022). Terminal security monitoring based on power consumption information. *Journal of Physics. Conference Series*, 2242(1), 012038. <https://doi.org/10.1088/1742-6596/2242/1/012038>.
- Chohan, M. N., Haider, U., Ayub, N. M. Y., Shoukat, N. H., Bhatia, N. T. K., & Hassan, N. M. F. U. (2023). Detection of Cyber Attacks using Machine Learning based Intrusion Detection System for IoT Based Smart Cities. *EAI Endorsed Transactions on Smart Cities*, 7(1). <https://doi.org/10.4108/eetsc.3222>.
- C. P. Pfleeger, Security in computing / Charles P. Pfleeger, Shari Lawrence Pfleeger. Upper Saddle River, NJ: Prentice Hall, 5th ed. ed., 2015. ISBN: 9780134085043.
- CCSDS, "Security threats against space missions," Tech. Rep. CCSDS 350.1-G-2, The Consultative Committee for Space Data Systems, December 2015.
- Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A. Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364, 2017.
- David, W. (2023). On the Philosophy of Unsupervised Learning. *Philosophy & Technology*, doi: 10.1007/s13347-023-00635-6.
- Dewan, M., Farid., Nabila, Sabrin, Sworna., Ruhul, Amin., Nazifa, Sadia., Moshiur, Rahman., Nazmul, Khan, Liton., Md., Saddam, Hossain, Mukta., Swakkhar, Shatabda. (2022). Boosting K-Nearest Neighbour (KNN) Classification using Clustering and AdaBoost Methods. doi: 10.1109/TENSYMP54529.2022.9864503.
- Dey, A. Machine learning algorithms: a review. *International Journal of Computer Science and Information Technologies*, 2016, 7(3): 1174-1179.
- D. G. Hennecken, "Beyza unal: Cybersecurity of nato's space-based strategic assets. london: Chatham house, juli 2019," SIRIUS : Zeitschrift fur Strategische Analysen, vol. 4, no. 2, pp. 227-228, 2020.
- D. Li, "Cyber-attacks on space activities: Revisiting the responsibility regime of article vi of the outer space treaty," *Space Policy*, vol. 63, p. 101522, 2023.
- Das, A., Pramod, & B, S. (2022). An efficient feature selection approach for intrusion detection system using decision tree. *International Journal of Advanced Computer Science and Applications : IJACSA*, 13(2). <https://doi.org/10.14569/ijacsa.2022.0130276>



- Diro, A., Kaisar, S., Vasilakos, A. V., Anwar, A., Nasirian, A., & Olani, G. (2024). Anomaly detection for space information networks: A survey of challenges, techniques, and future directions. *Computers & Security*, *139*, 103705. <https://doi.org/10.1016/j.cose.2024.103705>
- Dubosq, R., Schneider, D., Rogowitz, A., & Gault, B. (2022). *Unraveling the secrets of the Earth through nanogeology: A correlative microscopy approach*. <https://doi.org/10.5194/egusphere-egu22-1186>.
- Eder-Neuhauser, P., Zseby, T., Fabini, J., & Vormayr, G. (2017). Cyber attack models for smart grid environments. *Sustainable Energy Grids and Networks*, *12*, 10–29. <https://doi.org/10.1016/j.segan.2017.08.002>
- Ellerbeck, S. (2022, October 19). *The space economy is booming. What benefits can it bring to Earth?* World Economic Forum. <https://www.weforum.org/agenda/2022/10/space-economy-industry-benefits/>.
- Elsayed, R., Hamada, R., Hammoudeh, M., Abdalla, M., Elsaid, S.A.: A hierarchical deep learning based intrusion detection architecture for clustered Internet of Things. *J. Sens. Actuator Network*. 12(1), 3 (2022). \*
- Eshakagdy, M., Matter, A.H.M.E.D., Hussin, S., Hassan, D., Elsaid, S.: A Comparative study of intrusion detection systems applied to NSL-KDD Dataset. *Egypt. Int. J. Eng. Sci. Technol.* (2022). <https://doi.org/10.21608/eijest.2022.137441.1156>.
- Falco, G., Henry, W., Aliberti, M., Bailey, B., Bailly, M., Bonnart, S., Boschetti, N., Bottarelli, M., Byerly, A., Brule, J., Carlo, A., Rossi, G. D., Epiphaniou, G., Fetrow, M., Floreani, D., Gordon, N. G., Greaves, D., Jackson, B., Jones, G., ... Wallen, M. (2022). An international technical standard for commercial space system cybersecurity - A call to action. *ASCEND 2022*.
- Faleiros, M. C., Nogueira-Barbosa, M. H., Dalto, V. F., Júnior, J. R. F., Tenório, A. P. M., Luppino-Assad, R., Louzada-Junior, P., Rangayyan, R. M., & de Azevedo-Marques, P. M. (2020). Machine learning techniques for computer-aided classification of active inflammatory sacroiliitis in magnetic resonance imaging. *Advances in Rheumatology (London, England)*, *60*(1). <https://doi.org/10.1186/s42358-020-00126-8>.
- Fu, Z. (2022). Computer cyberspace security mechanism supported by cloud computing. *PloS One*, *17*(10), e0271546. <https://doi.org/10.1371/journal.pone.0271546>
- Gao, X., Deng, F., Zeng, P., & Zhang, H. (2023, March). Adaptive Neural Event-Triggered Control of Networked Markov Jump Systems Under Hybrid Cyberattacks. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(3), 1502–1512. <https://doi.org/10.1109/tnnls.2021.3105532>.

- Greenberg, A. The Untold Story of NotPetya, the Most Devastating Cyber-attack in History [online] Available at: <https://www.wired.com/story/notpetya-cyber-attack-ukraine-russia-code-crashed-the-world/> [Accessed 18 March 2022].
- Gu, J.; Wang, L.; Wang, H.; Wang, S. A novel approach to intrusion detection using SVM ensemble with feature augmentation. *Comput. Secur.* 2019, 86, 53–62.
- Gu, Z., Shi, P., Yue, D., Yan, S., & Xie, X. (2021, October). Memory-Based Continuous Event-Triggered Control for Networked T–S Fuzzy Systems Against Cyberattacks. *IEEE Transactions on Fuzzy Systems*, 29(10), 3118–3129. <https://doi.org/10.1109/tfuzz.2020.3012771>.
- Hammi, B., & Zeadally, S. (2023). Software supply-chain security: Issues and countermeasures. *Computer*, 56(7), 54–66. <https://doi.org/10.1109/mc.2023.3273491>
- Haque, A., Chowdhury, M. N., Soliman, H., Hossen, M. S., Fatima, T., & Ahmed, I. (2023). Wireless Sensor Networks anomaly detection using Machine Learning: A Survey. *ArXiv*. /abs/2303.08823.
- H. Al-Hraishawi, H. Chougrani, S. Kisseleff, E. Lagunas and S. Chatzinotas, "A Survey on Nongeostationary Satellite Systems: The Communication Perspective," in *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 101–132, First quarter 2023, doi: 10.1109/COMST.2022.3197695.
- Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature selection for intrusion detection using random forest. *Journal of Information Security*, 07(03), 129–140. <https://doi.org/10.4236/jis.2016.73009>.
- Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., Sharma, B., Chowdhury, S.: Composition of hybrid deep learning model and feature optimization for intrusion detection system. *Sensors* 23(2), 890 (2023).
- Ho, S., Jufout, S. A., Dajani, K., & Mozumdar, M. (2021). A Novel Intrusion Detection Model for Detecting Known and Innovative Cyberattacks Using Convolutional Neural Network. *IEEE Open Journal of the Computer Society*, 2, 14–25. <https://doi.org/10.1109/ojcs.2021.3050917>.
- Idris, I., & Damilola, A. N. (2023). Systematic literature review and metadata analysis of insider threat detection mechanism. *International Journal of Computer Science and Mobile Computing*, 12(4), 60–88. <https://doi.org/10.47760/ijcsmc.2023.v12i04.007>
- Iman Sharafaldin, Arash Habibi Lashkari, and Ali A.Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International
- ITGovernance, available under <https://www.itgovernance.co.uk/blog/list-of-data-breaches-and-cyber-attacks-in-february-2022-5-1-million-records-breached> [Accessed 20 March 2022].

- J., Gajda., J., Kwiecień., Wojciech, Chmiel. (2022). Machine learning methods for anomaly detection in computer networks. 276-281. doi: 10.1109/MMAR55195.2022.9874341.
- J. Kevric, S. Jukic, and A. Subasi, “An effective combining classifier approach using tree algorithms for network intrusion detection,” *Neural Computing and Applications*, vol. 28, Supplement1, pp. 1051–1058, 2017.
- J. Pavur and I. Martinovic, “The cyber-asat: on the impact of cyber weapons in outer space,” in 2019 11th International Conference on Cyber Conflict (CyCon), vol. 900. IEEE, 2019, pp. 1–18.
- J. Pearson, “Russia downed satellite internet in ukraine-western officials,” *Reuters*, de, vol. 10, 2022.
- Jiang, K., Wang, W., Wang, A., Wu, H.: Network intrusion detection combined hybrid sampling with deep hierarchical network. *IEEE Access* 8, 32464–32476 (2020).
- Jora, O.-D., Roşca, V. I., Iacob, M., Murea, M.-M., & Nedef, M.-Ştefan. (2023). Small and medium enterprises shooting for the stars: What matters, besides size, in outer space economy? *Management & Marketing*, 18(1), 20–35. <https://doi.org/10.2478/mmcks-2023-0002>.
- Jordan, M. I., & Mitchell, T. M. (2015, July 17). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255–260. <https://doi.org/10.1126/science.aaa8415>.
- K. Thangavel, J. J. Plotnek, A. Gardi, and R. Sabatini, “Understanding and investigating adversary threats and countermeasures in the context of space cybersecurity,” in 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC). IEEE, 2022, pp. 1–10.
- Khazaei, J. (2021, May). Stealthy Cyberattacks on Loads and Distributed Generation Aimed at Multi-Transmission Line Congestions in Smart Grids. *IEEE Transactions on Smart Grid*, 12(3), 2518–2528. <https://doi.org/10.1109/tsg.2020.3038045>
- Khazaei, J., & Asrari, A. (2022, September). Second-Order Cone Programming Relaxation of Stealthy Cyberattacks Resulting in Overvoltages in Cyber-Physical Power Systems. *IEEE Systems Journal*, 16(3), 4267–4278. <https://doi.org/10.1109/jsyst.2021.3108635>
- Kola, V. (2022). Practical Approach Of Implementing Artificial Intelligence. *Journal of Electronics, Computer Networking and Applied Mathematics*, doi: 10.55529/jecnam.22.21.24.
- Li, K., Zhou, H., Tu, Z., Wang, W., Zhang, H.: Distributed network intrusion detection system in satellite-terrestrial integrated networks using federated learning. *IEEE Access* 8, 214852–214865 (2020).

- Li, H., Li, X., Liu, X., Bu, X., Li, H., & Lyu, Q. (2022). Prediction of blast furnace parameters using feature engineering and Stacking algorithm. *Ironmaking and Steelmaking*, 49(3), 283–296. <https://doi.org/10.1080/03019233.2021.1992816>
- Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., Jurafsky, D. Deep reinforcement learning for dialogue generation. arXiv preprint arXiv:1606.01541, 2016.
- Liu, Z., & Wang, L. (2021). FlipIt Game Model-Based Defense Strategy Against Cyberattacks on SCADA Systems Considering Insider Assistance. *IEEE Transactions on Information Forensics and Security*, 16, 2791–2804. <https://doi.org/10.1109/tifs.2021.3065504>
- López-Dorado, A., Pérez, J., Rodrigo, M. J., Miguel-Jiménez, J. M., Ortiz, M., de Santiago, L., López-Guillén, E., Blanco, R., Cavalliere, C., Morla, E. M. S., Boquete, L., & Garcia-Martin, E. (2021). Diagnosis of multiple sclerosis using multifocal ERG data feature fusion. *An International Journal on Information Fusion*, 76, 157–167. <https://doi.org/10.1016/j.inffus.2021.05.006>.
- Machine learning (ML) in cybersecurity*. (2023, September 8). SailPoint; SailPoint Technologies. <https://www.sailpoint.com/identity-library/how-ai-and-machine-learning-are-improving-cybersecurity/>.
- M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, “Deep learning approach combining sparse autoencoder with SVM for network intrusion detection,” *IEEE Access*, vol. 6, pp. 52843–52856, 2018.
- Maseer, Z.K., Yusof, R., Bahaman, N., Mostafa, S.A., Foozy, C.F.M.: Benchmarking of machine Learning for anomaly-based intrusion detection systems in the CICIDS2017 dataset. *IEEE Access* 9, 22351–22370 (2021).
- Mayet, A. M., Alizadeh, S. M., Nurgalieva, K. S., Hanus, R., Nazemi, E., & Narozhnyy, I. M. (2022). Extraction of time-domain characteristics and selection of effective features using correlation analysis to increase the accuracy of petroleum fluid monitoring systems. *Energies*, 15(6), 1986. <https://doi.org/10.3390/en15061986>.
- McCarthy, A., Ghadafi, E., Andriotis, P., & Legg, P. (2022). Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: A survey. In *Preprints*. <https://doi.org/10.20944/preprints202202.0099.v1>.
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2022). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6), 1–35. <https://doi.org/10.1145/3457607>.
- Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Computing*, 17(3), 12–22. <https://doi.org/10.1109/mprv.2018.03367731>.

- Muhammad, H., Longe, O. B., Baale, A., & Antai, U.-O. E. (2022). Towards the development of a machine learning Enhanced Framework for Honeypot and CAPTCHA intrusion detection systems. *Advances in Multidisciplinary & Scientific Research Journal Publication*, 34, 43–50. <https://doi.org/10.22624/aims/acrabespoke2022/v34p4>.
- Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." *Military Communications and Information Systems Conference (MilCIS)*, 2015. IEEE, 2015.
- Mowla, N.I.; Tran, N.H.; Doh, I.; Chae, K. AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET. *J. Commun. Netw.* 2020, 22, 244–258.
- Musafer, H.; Abuzneid, A.; Faezipour, M.; Mahmood, A. An enhanced design of sparse autoencoder for latent features extraction based on trigonometric simplexes for network intrusion detection systems. *Electronics* 2020, 9, 259.
- Naveed, M., Arif, F., Usman, S. M., Anwar, A., Hadjouni, M., Elmannai, H., Hussain, S., Ullah, S. S., & Umar, F. (2022). A deep learning-based framework for feature extraction and classification of intrusion detection in networks. *Wireless Communications and Mobile Computing*, 2022, 1–11. <https://doi.org/10.1155/2022/2215852>.
- N., C., Thoutam., Mayur, Sonawane., Ghanshyam, R., Chaudhari., Om, Kathe., Prajwal, Sontakke. (2023). Machine Learning for the Identification of Network Anomalies. *Indian Scientific Journal Of Research In Engineering And Management*, 07(03) doi: 10.55041/ijsrem18082.
- Nguyen, N.T., Chang, C.C.: A biometric-based authenticated key agreement protocol for user-to-user communications in mobile satellite networks. *Wirel. Pers. Commun.* 107(4), 1727–1758 (2019).
- Ni, X. (2022). Research on image recognition technology based on machine learning. *Frontiers in Business, Economics and Management*, 6(2), 110–113. <https://doi.org/10.54097/fbem.v6i2.2819>.
- Orsini, H., Bao, H., Zhou, Y., Xu, X., Han, Y., Yi, L., Wang, W., Gao, X., & Zhang, X. (2022). AdvCat: Domain-agnostic robustness assessment for cybersecurity-critical applications with categorical inputs. *2022 IEEE International Conference on Big Data (Big Data)*.
- Oyama, H., Rangan, K. K., & Durand, H. (2021, June 26). Handling of stealthy sensor and actuator cyberattacks on evolving nonlinear process systems. *Journal of Advanced Manufacturing and Processing*, 3(3). <https://doi.org/10.1002/amp2.10099>.
- Panigrahi, L., Pattanayak, B. K., Mohanty, B., Pattnaik, S., & Habboush, A. K.. (2024). A Smart Secure model for Detection of DDoS Malicious Traces in Integrated LEO Satellite-Terrestrial Communications. 12(2). <https://doi.org/10.37391/ijeer-120223>.

Paul Maguire, "Satellites and the spectre of IoT attacks". <https://spacenews.com/satellites-specter-iot-attacks/>. Accessed: 2024-01-26.

Pavur, J. (2021). *Securing new space: on satellite cyber-security (Doctoral dissertation)*.

Prasad, M., Pal, P., Tripathi, S., & Dahal, K. (2022). AI/ML driven intrusion detection framework for IoT-enabled cold storage monitoring system. In *Research Square*. <https://doi.org/10.21203/rs.3.rs-2190363/v1>.

Radoglou-Grammatikis, P., Sarigiannidis, P., Efstathopoulos, G., Karypidis, P.-A., & Sarigiannidis, A. (2020). DIDEROT: An intrusion detection and prevention system for DNP3-based SCADA systems. *Proceedings of the 15th International Conference on Availability, Reliability and Security*.

Rajagopalan, R. (2019), Electronic and Cyber Warfare in Outer Space [online] Available at: <https://www.unidir.org/files/publications/pdfs/electronic-and-cyber-warfare-in-outer-space-en-784.pdf> [Accessed 10 July 2021].

Ronald, M., Wei, L., Slay, M., Wei, L., Slay, M., Wei, L., Slay, M., & Wei, L. (2023). *Deep-learning-based Intrusion Detection for Software-defined Networking Space Systems*. 22, 639–647. <https://doi.org/10.34190/eccws.22.1.1085>.

Rath, M., Mishra, S.: Security approaches in machine learning for satellite communication. In: *Machine Learning and Data Mining in Aerospace Technology*, pp. 189–204 (2020).

R.Bruckardt, "How will the space economy change the world?" <https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/how-will-the-space-economy-change-the-world>, 2022, [Online; accessed 20-June-2023].

R. Hutchins, "Cyber defense of space assets," Master's thesis, Tufts University Department of Computer Science, Boston, Massachusetts, 2016.

Riana, R., & Mangkurat, U. L. (2023). Implementation of information gain and particle swarm optimization upon covid-19 handling sentiment analysis by using k-nearest neighbor. *JIKO (Jurnal Informatika Dan Komputer)*, 6(1), 7–12. <https://doi.org/10.33387/jiko.v6i1.5260>.

Ribeiro, A.A.; Sachine, M. On the optimal separating hyperplane for arbitrary sets: A generalization of the SVM formulation and a convex hull approach. *Optimization* 2020, 71, 213–226. \*

S., P., Krithivasan, K., S., P., & Sriram V.S., S. (2020, July). Detection of Cyberattacks in Industrial Control Systems Using Enhanced Principal Component Analysis and Hypergraph-Based Convolution Neural Network (EPCA-HG-CNN). *IEEE Transactions on Industry Applications*, 56(4), 4394–4404. <https://doi.org/10.1109/tia.2020.2977872>.

- Sahu, S.K., Mohapatra, D.P., Rout, J.K., Sahoo, K.S., Pham, Q.V., Dao, N.N.: A LSTM-FCNN based multi-class intrusion detection using scalable framework. *Comput. Electr. Eng.* 99, 107720 (2022).
- Sarker, I. H. (2021). *Machine Learning: Algorithms, Real-World Applications and Research Directions*. *SN Computer Science*. 2.
- Shaveta (2023). A review on machine learning. *International Journal of Science and Research Archive*, 9(1):281-285. doi: 10.30574/ijrsra.2023.9.1.0410
- Sindhu, P., Menon. (2022). Challenges in KNN Classification. *IEEE Transactions on Knowledge and Data Engineering*, doi: 10.1109/tkde.2021.3049250.
- Sosa-Cabrera, G., Gómez-Guerrero, S., García-Torres, M., & Schaerer, C. E. (2023). *Feature Selection: A perspective on inter-attribute cooperation*. <https://doi.org/10.48550/ARXIV.2306.16559>.
- Stoudenmire, E., Schwab, D. J. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, 2016: 4799-4807.
- Suo, D., Moore, J., Boesch, M., Post, K., & Sarma, S. E. (2022). Location-based schemes for mitigating cyber threats on connected and automated vehicles: A survey and design framework. *IEEE Transactions on Intelligent Transportation Systems: A Publication of the IEEE Intelligent Transportation Systems Council*, 23(4), 2919–2937. <https://doi.org/10.1109/tits.2020.3038755>.
- Snider, K. L. G., Shandler, R., Zandani, S., & Canetti, D. (2021, January 1). Cyberattacks, cyber threats, and attitudes toward cybersecurity policies. *Journal of Cybersecurity*, 7(1). <https://doi.org/10.1093/cybsec/tyab019>
- Song, X., Zhang, Q., Song, S., & Ahn, C. K. (2022, July). Sampled-Data-Based Event-Triggered Fuzzy Control for PDE Systems Under Cyberattacks. *IEEE Transactions on Fuzzy Systems*, 30(7), 2693–2705. <https://doi.org/10.1109/tfuzz.2021.3092200>.
- Sylvester Kaczmarek, "Cybersecurity for satellites is a growing challenge". <https://www.thespacereview.com/article/4747/1>. Accessed: 2024-02-26.
- Szolucha, A. (2022). Space exploration and the imaginaries of living in a climate-changing world. In *SocArXiv*. <https://doi.org/10.31235/osf.io/pr538>.
- Tahri, R., Balouki, Y., Jarrar, A., & Lasbahani, A.. (2022). Intrusion Detection System Using Machine Learning Algorithms. 46. <https://doi.org/10.1051/itmconf/20224602003>.

- Takyi, K., Bagga, A., & Goopta, P. (2018). Clustering techniques for traffic classification: A comprehensive review. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*.
- Tedeschi, P., Sciancalepore, S., & Di Pietro, R. (2022). Satellite-based communications security: A survey of threats, solutions, and research challenges. *Computer Networks*, *216*, 109246. <https://doi.org/10.1016/j.comnet.2022.109246>.
- “Types of Satellites and Applications.” <https://satellite.insightconferences.com/events-list/types-of-satellites-and-applications>. Accessed: 2021-05-20.
- Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K.-L. A., Elkhatib, Y., Hussain, A., & Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 65579–65615. <https://doi.org/10.1109/access.2019.2916648>.
- V. Varadharajan, “Security Challenges when Space Merges with Cyberspace,” <https://arxiv.org/ftp/arxiv/papers/2207/2207.10798.pdf>, 2022, [Online; accessed 20-June-2023].
- Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. *IEEE Access* *7*, 41525 -41550 (2019).
- Wahyono, T., & Heryadi, Y. (2019). Machine learning applications for anomaly detection. In *Computational Intelligence in the Internet of Things* (pp. 49–83). IGI Global.
- Wang, W. (2023). SPACE: a new modeling tool for supporting layout design of military command and control spaces. *The Journal of Defense Modeling and Simulation Applications Methodology Technology*, 154851292311653. <https://doi.org/10.1177/15485129231165310>.
- Wang, F., Li, C., Niu, S., Wang, P., Wu, H., & Li, B. (2022). Design and analysis of a spherical robot with rolling and jumping modes for deep space exploration. *Machines*, *10*(2), 126. <https://doi.org/10.3390/machines10020126>.
- “What Is a Satellite.” <https://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-a-satellite-58.html>. Accessed: 2021-05-18.
- Xiao, W. (2022). *Analysis and Detection of GPS Spoofing Attacks in Cyber Physical Systems (Doctoral dissertation)*.
- Xu, B.; Shirani, A.; Lo, D.; Alipour, M.A. Prediction of relatedness in stack overflow: Deep learning vs. SVM: A reproducibility study. In Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, Oulu, Finland, 11–12 October 2018; pp. 1–10. \*



Y. Abe, H. Tsuji, A. Miura, S. Adachi, Frequency Resource Management Based on Model Predictive Control for Satellite Communications System, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E101.A (12) (2018) 2434-2445.

Zografopoulos, I., Ospina, J., Liu, X., & Konstantinou, C. (2021). Cyber-Physical Energy Systems Security: Threat Modeling, Risk Assessment, Resources, Metrics, and Case Studies. *IEEE Access*, 9, 29775–29818. <https://doi.org/10.1109/access.2021.3058403>

Z. A. Baig, S. Sanguanpong, S. N. Firdous, T. G. Nguyen, C. So-In, Averaged dependence estimators for DOS attack detection in iot networks, *Future Generation Computer Systems* 102 (2019), pp. 198–209. \*

Zoph, B., Le, Q. V. Neural architecture search with reinforcement learning. arXiv preprint arXiv: 1611.01578,2016.