NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A NETWORK INTRUSION DETECTION SYSTEM USING DECISION TREE MACHINE LEARNING ON AN ISTN ARCHITECTURE

by

Kok Siong J. Yap

March 2022

Thesis Advisor: Preetha Thulasiraman
Second Reader: Murali Tummala

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2022	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
A NETWORK INTRUSION DETECTION SYSTEM USING DECISION TREE MACHINE LEARNING ON AN ISTN ARCHITECTURE

5. FUNDING NUMBERS

6. AUTHOR(S) Kok Siong J. Yap

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

In recent years, the Navy has shown interest in an integrated satellite-terrestrial networking (ISTN) architecture for unmanned systems. With the development of satellite networks and growing numbers of unmanned system networks being connected, security and privacy are major concerns in an ISTN. In this thesis, we develop a network intrusion detection system (NIDS) specifically for an ISTN. We identify the critical location of the NIDS within the ISTN architecture and use the decision tree machine learning algorithm to perform cyber-attack detection against various threat vectors, including distributed denial of service. The decision tree algorithm is used to classify and segregate attack traffic from benign traffic. We use an open source ISTN data set available in the literature to train our algorithm. The decision tree is implemented using different split criteria, varying number of splits, and the use of principal component analysis (PCA). We manipulate the size of the training data and the number of data features to achieve reasonable false positive rates. We show that our NIDS framework based on decision tree learning can effectively detect and segregate different attack data classes.

14. SUBJECT TERMS
integrated satellite-terrestrial networking architecture, ISTN, network intrusion detection system, NIDS, decision tree, DT, principal component analysis, PCA

15. NUMBER OF PAGES
79

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

A NETWORK INTRUSION DETECTION SYSTEM USING DECISION TREE MACHINE LEARNING ON AN ISTN ARCHITECTURE

Kok Siong J. Yap
Civilian, DSO National Labs
BSEE, National University of Singapore, 2010

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2022

Approved by: Preetha Thulasiraman
Advisor

Murali Tummala
Second Reader

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In recent years, the Navy has shown interest in an integrated satellite-terrestrial networking (ISTN) architecture for unmanned systems. With the development of satellite networks and growing numbers of unmanned system networks being connected, security and privacy are major concerns in an ISTN. In this thesis, we develop a network intrusion detection system (NIDS) specifically for an ISTN. We identify the critical location of the NIDS within the ISTN architecture and use the decision tree machine learning algorithm to perform cyber-attack detection against various threat vectors, including distributed denial of service. The decision tree algorithm is used to classify and segregate attack traffic from benign traffic. We use an open source ISTN data set available in the literature to train our algorithm. The decision tree is implemented using different split criteria, varying number of splits, and the use of principal component analysis (PCA). We manipulate the size of the training data and the number of data features to achieve reasonable false positive rates. We show that our NIDS framework based on decision tree learning can effectively detect and segregate different attack data classes.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
ASM	Attribute Selection Measures
DDoS	Distributed Denial of Service
DoS	Denial of Service
DT	Decision Tree
DTN	Delay and Disruption Tolerant Networking
GDI	Gini's Diversity Index
GEO	Geostationary Earth Orbit
HF	High Frequency
HQ	Headquarters
IDS	Intrusion Detection System
IGPO	International Graduate Programs Office
ISTN	Integrated Satellite Terrestrial Networking
LEO	Low Earth Orbit
LPWAN	Low Power Wide Area Networks
MDRS	Maximum Deviance Reduction Selection
MITM	Man-in-the-Middle
ML	Machine Learning
MSEE	Master of Science in Electrical Engineering
NCC	Network Control Center
NIDS	Network Intrusion Detection System
NMC	Network Management Center
NPS	Naval Postgraduate School
PCA	Principal Component Analysis
RFID	Radio Frequency Identification
SDR	Software Defined Radio
SNA	Satellite Network Attack
SVD	Singular Value Decomposition
TNA	Terrestrial Network Attack
TR	Twoing Rule

UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UGV	Unmanned Ground Vehicle
USS	Unmanned Space Systems
USV	Unmanned Sea Vehicles
VLF	Very Low Frequency

ACKNOWLEDGMENTS

Throughout the writing of my thesis, I have received moral support and assistance from my family, advisor, colleagues, and great friends around me.

First and foremost, I want to convey my heartfelt appreciation to my advisor, Dr. Preetha Thulasiraman, who has been very patient guiding me throughout the thesis process with motivation and enthusiasm. This thesis would not have been feasible without her continual guidance and support. I would also like to thank my wife, Tingting, and my delightful son—Nicholas—for their support, love, and understanding while I pursued my postgraduate education overseas.

In addition, I would like thank Julie Samples for her great effort in helping me and fellow students meet administrative and thesis requirements.

I would also like to thank Erin Ferguson from the International Graduate Programs Office (IGPO) for her great hospitality towards the overseas students dealing with administrative work.

Lastly, I would like to thank all my fellow classmates, lecturers, staff, and all the faculty of the Electrical Engineering Department for making this such an enjoyable experience at NPS.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In recent years, there has been a surge in interest in Integrated Satellite-Terrestrial Networking (ISTN) architectures based on unmanned or autonomous systems. Over the last several years, developing interoperable networking protocols to connect satellite networks with terrestrial networks has been an ongoing topic of research. In addition, investigations on how to establish secure networking to prevent cyberattacks against ISTNs is also of interest.

A. WHAT IS AN INTEGRATED SATELLITE-TERRESTRIAL NETWORK?

An ISTN combines the satellite network and the terrestrial network, which are two separate systems. It has the potential to be a cornerstone in the implementation of a heterogeneous global system that provides seamless, omnipresent internet access while also improving the end user experience in both commercial and military applications [1]. One of the most efficient ways to achieve global connectivity is to establish an infrastructure for network access via satellite communication. Satellites can efficiently complement and develop dense terrestrial networks in both urban and rural locations, as well as provide mission-critical services with reliability, thanks to their large footprint [1].

However, the existing satellite networks and terrestrial networks are built independently. The different protocol designs of each network, in particular, makes it harder to integrate the two systems. Unlike terrestrial networks, which typically have steady transmission conditions, satellite networks incur significant propagation delays, substantial bit rate error, and sporadic connectivity. Therefore, the satellite and terrestrial communities are collaborating to develop a fully functional satellite-terrestrial architecture.

An example of an ISTN is shown in Figure 1. The networks support routing, adaptive access control, and spot-beam management. It consists of Geostationary Earth Orbit (GEO) satellites, Low Earth Orbit (LEO) satellite, ground stations (gateways), switches, and servers [1].

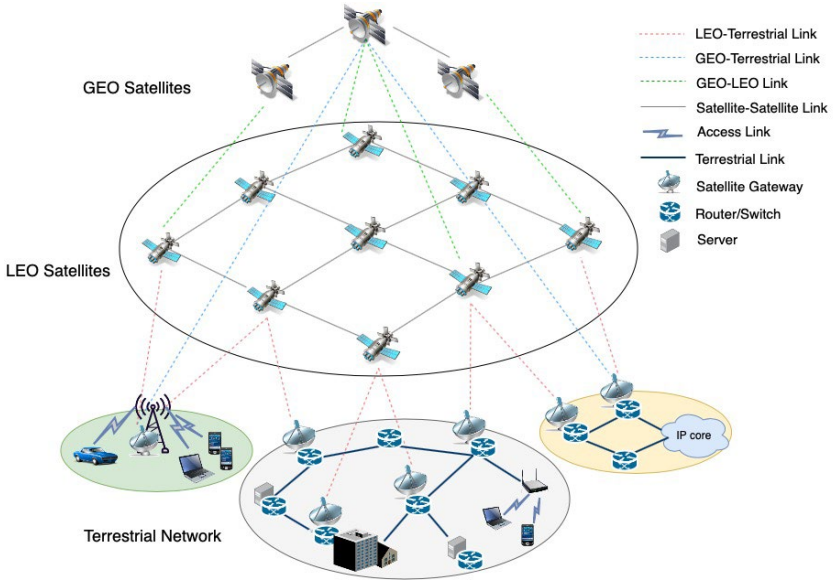

Figure 1. Example of an integrated satellite-terrestrial network. Source: [1].

Due to limited resources, insufficient attack resistance, and the high privacy requirements of satellite networks, existing ISTNs are vulnerable to security and privacy challenges. Therefore, in order to achieve a greater level of security against present and future network environments, a number of researchers have proposed developing and implementing a Network Intrusion Detection System (NIDS) for detecting both normal and malicious traffic in the ISTN.

B. RESEARCH MOTIVATIONS AND THESIS CONTRIBUTIONS

In the last decade, ISTNs have significantly increased the capacity of data transport between space and terrestrial networks [2]. However, with the development of satellite networks and growing numbers of unmanned system networks being connected, security and privacy are a major concern in ISTN. In the event of a cyberattack on the satellite network, the network resources will be depleted fast, making recovery difficult. Even with the installation of a firewall system, the security system is unable to identify contemporary attack settings and perform in-depth network packet analysis [2].

Therefore, it is still possible that the network can be penetrated. This leads to the need for a NIDS using machine learning for the ISTN. Existing research does not focus on one type of machine learning technique for the ISTN, and there is little established work on the placement of NIDS and the impact its location can have on ISTN networking.

The work presented in this thesis is built upon two published research efforts. First, Cheng [3] proposed a unique network architecture and design scenario that aligns with the needs of the PMW760 (Ship Integration Office). Second, Li et al. [2] implemented NIDS using Federated Learning and generated their own datasets for ISTNs.

In this thesis, we have proposed and adopted upon these works by 1) modifying the unique case architecture network presented in [3] into a simpler ISTN with various unmanned systems and 2) using the datasets generated for ISTN by [2] in order to implement the Decision Tree machine learning algorithm and identifying the placement of NIDS in the ISTN.

The contributions of this thesis are as follows:

- Identification of the placement of NIDS for a proposed cohesive ISTN architecture with various unmanned systems.
- Incorporation of NIDS with Network Control Center (NCC) and a Network Management Center (NMC), which is comprised of a data collector, feature extractor, classifier, anomaly detector, and response administrator to allow the network architecture to react according to the cyberattack threat.
- Use of valid integrated satellite-terrestrial communication network security datasets from an open-source thesis to perform cyberattack detection using Decision Tree machine learning.
- Proof that different attacks can be grouped together based on objectives, and that supervised machine learning using the decision tree algorithm can segregate the various classes of attack traffic from benign traffic.

C. THESIS ORGANIZATION

The rest of this thesis is organized as follows: In Chapter II, we provide an overview of unmanned systems. We also discuss the cybersecurity vulnerabilities of an ISTN and how NIDS can be used for cyberattack detection. Chapter III provides a brief overview on machine learning, how to choose a technique, and a more in-depth look at the Decision Tree (DT) classifier. Chapter IV describes our proposed integrated satellite-terrestrial network architecture with various unmanned systems, how cyberattacks impact the overall network architecture and how our proposed cybersecurity architecture is implemented. We also discuss how attacks are grouped for this study, and the reason for choosing Decision Tree algorithm

and the data set used. Chapter V presents the simulation results for the proposed ISTN architecture and provides analysis of the results. Chapter VI concludes the thesis and recommends possibilities for future work.

II. BACKGROUND AND RELATED WORK

In this chapter, we present an overview of an unmanned system, including their classification, abilities, and communication protocols. We then discuss the vulnerabilities of unmanned systems, satellite networks, and cyber threats against satellite networks. Finally, we discuss how an intrusion detection system works and which subset of network intrusion detection system types will be the most suitable for classification in an ISTN to detect a cyberattack.

A. OVERVIEW OF UNMANNED SYSTEMS

Unmanned systems are described as "electro-mechanical" systems that can conduct tasks independently in order to complete pre-determined missions [4]. Generally, they can be classified as unmanned aerial vehicle (UAV), unmanned ground vehicle (UGV), unmanned sea vehicles (USV) and unmanned space systems (USS). Typically there are five necessary abilities an unmanned system should have: 1) perception, 2) interaction, 3) information processing, 4) decision making, and 5) execution [5].

An unmanned system has a three-layer architecture, as shown in Figure 2. From right to left: physical layer, communication layer and application layer. From bottom to top: sea, ground, air and space platform.

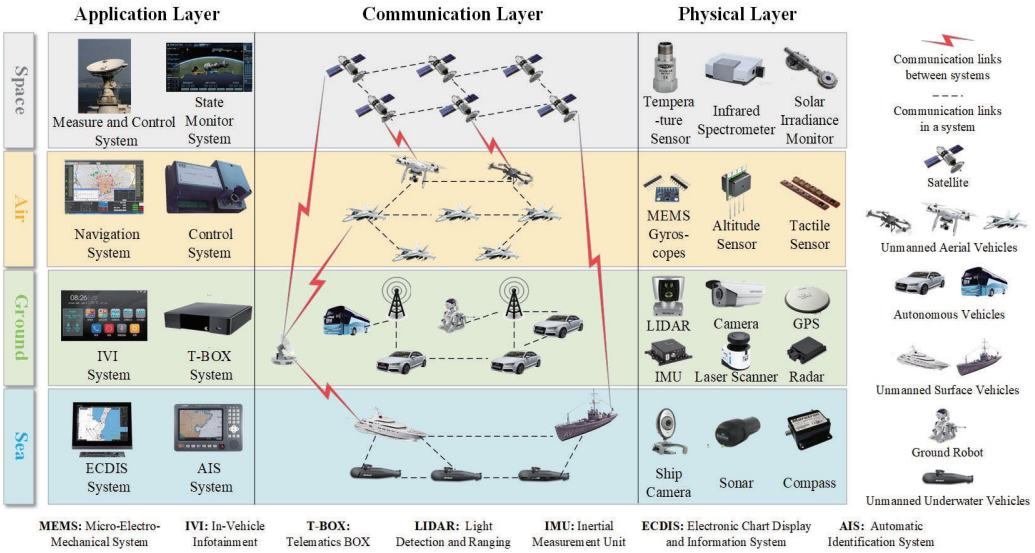

Figure 2. The three-layer architecture of unmanned systems. Source: [5].

Focusing on the communication layer in Figure 2, various wireless communication protocols can be used within the commercial and government sectors. Commonly used wireless commercial protocols include WiFi, Zigbee radio, cellular communication (2G, 3G, 4G, LTE and 5G), radio frequency identification (RFIDs), low-power wide-area networks (LPWANs), satellite, infrared and Bluetooth [6]. As for military or government wireless communication, the major focus is on ensuring network setup in a timely manner, operational dependability, speed of action, data security, and ease of equipment maintenance. Therefore, wireless military communication systems commonly use: very low-frequency (VLF), high-frequency (HF), terrestrial VHF, UHF and SHF tactical, satellite, wireless networking and software-defined radio (SDR) communications [6].

Although commercial and military wireless communication serve different applications using different technologies and frequency spectrum, there is one similarity between them which is the use of satellite communication. Satellites allow connections to many different devices and platforms to perform long distance or beyond visual line of sight communication. Therefore, satellite communication plays an important role in the establishment of an ISTN architecture, which is in the top level of the communication layer shown in Figure 2.

B. CYBERSECURITY VULNERABILITIES

1. Vulnerabilities of Unmanned System

Compared with manned systems, a typical unmanned system requires more sophisticated software to function. Thus, unmanned system software security requirements are likely to be more demanding and are more likely to face security problems. Data is a vital resource since the functioning of unmanned systems applications is highly reliant on networked systems that collect, store, analyze, and transfer data [7]. Hostile actors have also understood how valuable these resources are. As a result, cyber attackers take advantage of flaws in software, security policies, and communication technologies to obtain access to information and execute illegal acts in order to undermine cybersecurity.

2. Vulnerabilities of Satellite Network

In the current technological environment, we have seen satellite communication play an increasingly important part in a variety of 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 [8].

Additionally, having a legacy satellite communications platform in the space domain is more difficult to maintain compared to a terrestrial communication system. A terrestrial communication system allows quick upgrading and testing to assure communications, encryption, and increased cybersecurity easily but the same is not true 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.

Another point to note is that using botnets, ransomware, trojans, viruses, and other hacking tools, has the potential to disrupt the satellite network, and possibly bring it to a halt. Once the satellite communication (SATCOM) infrastructure is hacked, it is possible that 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 [9].

C. CYBER THREATS AGAINST SATELLITE NETWORKS

In the context of space systems, Figure 3 shows four segments of cyber threats, which are classified as space, user, link and ground [8]. There are two categories of cyber threats and attacks for satellite networks: passive and active.

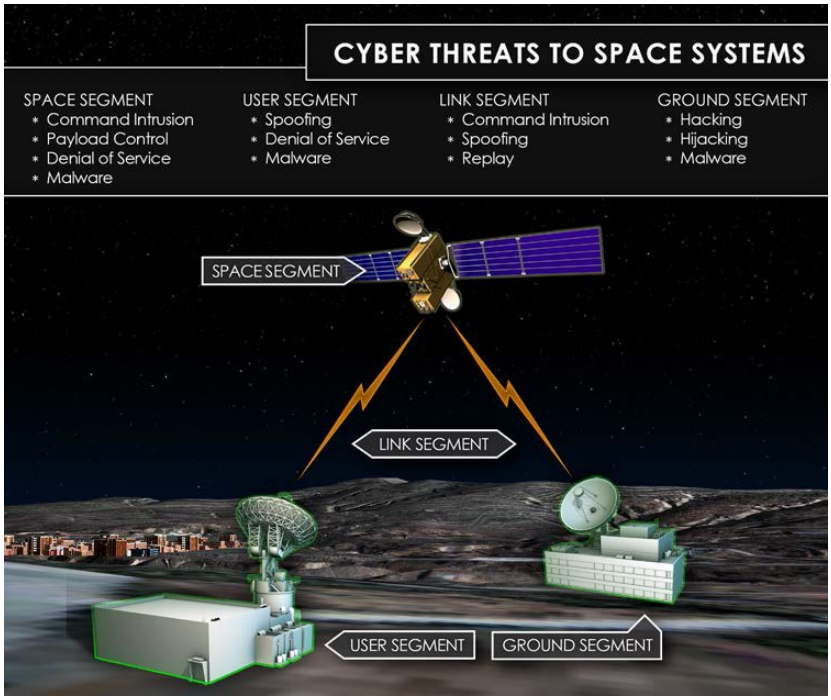

Figure 3. Cyber threats identified by NASIC. Source: [9].

1. Passive Attacks

Using a satellite terminal and fundamental understanding of the communication protocols, a passive attacker analyzes and observes communications meant for other terminals. [10]. Eavesdropping and spoofing attacks are a type of passive attack in which data is stolen between two devices connected to the internet [11]. However, it has no effect on system resources and is difficult to detect owing to the lack of change or alteration of data.

2. Active Attacks

An attacker can exploit a network by modifying or altering the content which can impact the system resource. Therefore, integrity and accessibility of the system will be threatened by an active attack. It includes a vast gamut of alternatives:

a. Denial-of-Service attacks (DoS / DDoS)

Once an attack occurs users are unable to access information systems, devices, or other network resources. Also, the system will not perform proper functions for a limited period of time due to the malicious cyber threat actions by flooding the targeted host or network with heavy traffic [11].

b. Masquerading attacks

According to [10], masquerading attacks refer to an attacker that impersonates or develops a fake identity in order to fool trusted communication parties and get unauthorized access data.

c. Trojan horse attacks

Trojan horse attacks refer to a cyberattack on a computer network. A backdoor trojan is an example of a malware that allows attackers with no authorization to get access to a system, network, or code application [11].

d. Replay attacks

This type of attack intercepts and replays traffic between two entities [10].

e. Man-in-the-Middle (MITM) attacks

Active network communications information sent between lawful communicating entities can be gathered, delayed or manipulated, and faked by malicious attackers [10].

D. INTRUSION DETECTION SYSTEM

In [13], an intrusion detection system (IDS) is defined as a hardware or software program that scans a network for harmful activities or policy infractions. Using a security information and event management system, any potentially risky conduct or violation is generally reported or gathered centrally. One of the most frequent forms of IDS classifications is the NIDS.

NIDS are used to analyze incoming network traffic or subnets continuously. According to [10], the NIDS uses machine learning, designed as classifiers to distinguish regular traffic from abnormal traffic. The installation of a NIDS at a critical junctures inside a network to monitor traffic to and from all network devices is critical [13]. NIDS allows to analyze passing traffic and compare it to a library of known assaults [4].

Once an attack or abnormality is detected, NIDS can alert the administrator immediately. Figure 4 illustrates an example of a simple NIDS system where The IDS is installed at the network edge or between the server and the network [12].

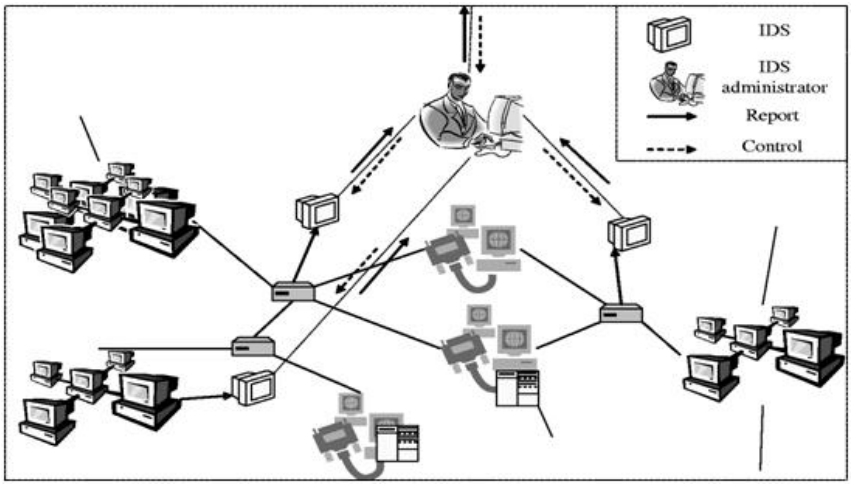

Figure 4. Network intrusion detection system. Source: [12].

Therefore, determining the position of the NIDS in an ISTN is crucial. In this thesis, we will be implementing and determining the position of a NIDS in a proposed ISTN architecture. Further details will be provided in Chapter IV.

1. Subset of IDS Types

IDS can also be categorized in two types: signature-based or anomaly-based.

a. Signature-based IDS

"A signature-based IDS detects possible threats by looking for specific patterns, such as byte sequences in network traffic, or known malicious instruction sequences used by malware" [13]. However, detecting new attacks for which no pattern exists is impossible.

b. Anomaly-based IDS

An anomaly based IDS identifies "deviations from normal traffic behavior and adapts to unknown attacks, primarily due to the explosion of malware" [13]. It compares new behavior to a defined model of trustworthy activity created via machine learning. This enables for the detection of previously unknown attacks, but it is prone to false positives, since previously undetected lawful behavior might be incorrectly classified as harmful [13].

2. Selection of IDS Classification

As mentioned in Chapter I, this thesis focuses on the detection of cybersecurity events for ISTN architecture. Anomaly-based NIDS will be the most suitable classification for an ISTN as it provides a more holistic approach to cyber threat detection.

In this thesis we will use supervised machine learning to implement NIDS in an ISTN. Supervised machine learning is an effective, foundational approach to detect anomalous traffic. This will help identify malicious cyberattacks in order to guarantee the privacy and security of its communications and data. We will also enable appropriate security responses to specific attacks.

THIS PAGE INTENTIONALLY LEFT BLANK

III. MACHINE LEARNING

In this chapter, we discuss further details about machine learning (ML) based on different criteria decision tree (DT) algorithms and how DT is chosen and implemented in this thesis.

A. MACHINE LEARNING METHODS

Machine learning (ML) is an artificial intelligence (AI) subfield. Machine learning methods enable computers to learn based on data inputs and then utilize statistical analysis to produce results that are within a given range. As a result, ML facilitates the creation of models from sample data by computers, allowing them to use data inputs to automate decision-making processes.

Generally, machine learning can be classified into the two most widely adopted methods: supervised and unsupervised learning.

1. Supervised Learning

Supervised learning is defined by "use of labeled datasets to train algorithms that are used to classify data or predict outcomes accurately" [14]. Supervised learning uses patterns to predict label values on unlabeled data. There are two types of supervised learning issues: (a) Classification and (b) Regression.

a. Classification

As explained by [15], a classification algorithm is used to correctly categorize test data and identify the dataset, and make certain predictions about how those items should be labeled. The common classification algorithms are random forest, K-nearest neighbor, support vector machines, liner classifiers, naïve bayes, discriminant analysis and decision tree.

b. Regression

Regression algorithms are used to project output values depending on data input attributes [15] . Its models can be applied for uses such as financial, forecasting and time series prediction. The common classification algorithms are liner regression, ensemble methods, decision tree and neural networks.

2. Unsupervised Learning

Machine learning algorithms that evaluate and cluster unlabeled datasets are known as unsupervised machine learning algorithms. Without the requirement for human participation, attacks are identified as hidden patterns or data groupings [14].

B. SELECTION OF A MACHINE LEARNING ALGORITHM

1. Overall Machine Learning Techniques

The fact is, there is no best method or one size fits all to find the right algorithm. However, we can take into consideration the size and types of data we are dealing with as well as the insights we want to get out of the data. Figure 5 shows a summary of ML techniques from MathWorks [16]. Our focus in this thesis is to show the feasibility of machine learning for anomaly detection in an ISTN. Our focus is on supervised machine learning. Supervised learning methods are simple, yet effective and provide foundational research in this area.

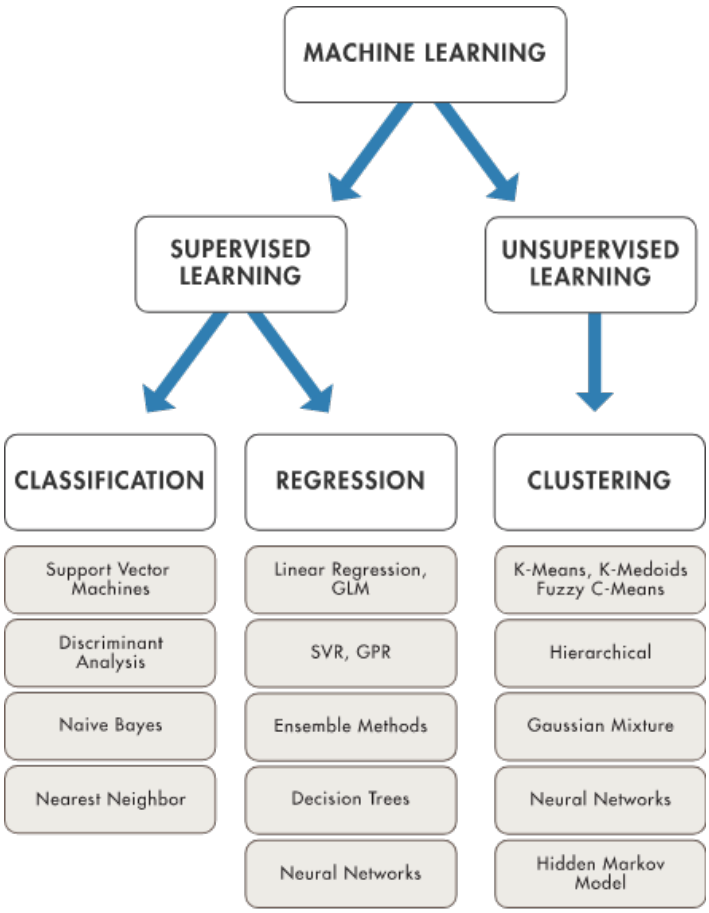

Figure 5. Machine learning techniques. Source: [16].

2. Decision Tree

For anomaly detection and prediction, we chose to use the Decision Tree (DT) supervised machine learning algorithm. Before we state the rationale for why this algorithm was chosen, we first discuss the methodology of the DT algorithm.

In [17], DT is represented as a tree-like layout with a tree branch indicating internal nodes to describe a test of a certain feature, reflecting the outcome of the test. The leaves of the tree refer to the nodes as a class. It is a non-parametric classifier and supervised learning system that uses conditional control statements to represent data characteristics as a tree shown in Figure 6 [18]. The goal of the DT classification technique is to anticipate a response and then follow the decisions down the tree from the root node to the leaf node, where the answer is stored.

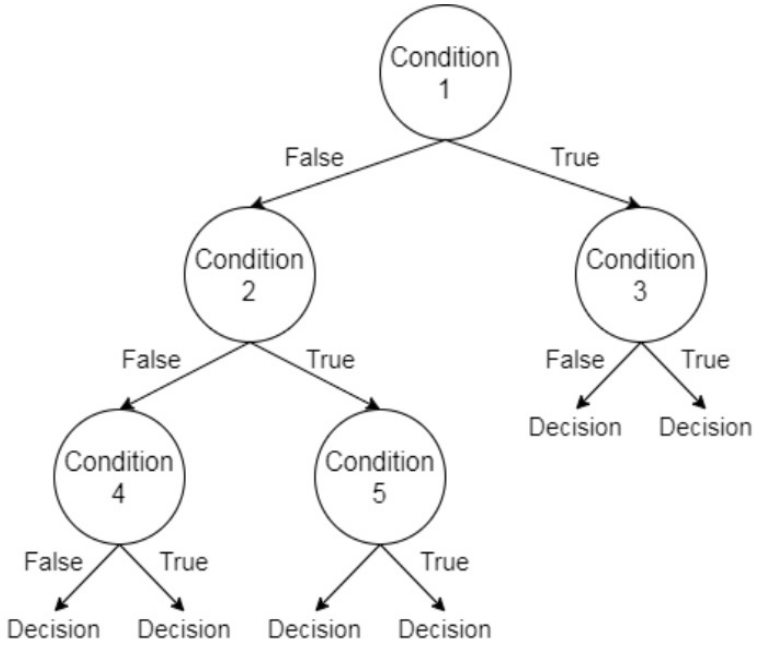

Figure 6. Illustration of the decision tree, conditional control. Source: [18].

3. Decision Tree Algorithm

In [19], the primary purpose of employing a decision tree is to use simple decision rules derived from the training data and develop a training model that can be used to forecast the target variable class or value.

The terminology related to decision trees are:

a. Root Node

Root Node represents the whole population or sample and is split into two or more homogenous groups [19].

b. Splitting

The division of a node into two or more sub-nodes.

c. Decision Node

The point at which a sub-node is split into further sub-nodes.

d. Pruning

Removal of unwanted branches from sub-nodes of a decision node. It reduces the size of the decision trees.

e. Branch / Sub-tree

A subsection of the entire tree.

f. Parent & Child Node

"A parent node of sub-nodes is a node that has been split into sub-nodes, whereas sub-nodes are the child of a parent node" [19].

Figure 7 shows an example of a DT, where each internal node splits down the tree from the root node into two or more sub-nodes based on the values of an input attribute discrete function. In each decision node a single attribute is partitioned according to the attribute values. Therefore, each terminal node (B and C), also known as leaf nodes, are allocated to a single class that represents the best suitable goal values or responses. This is a recursive procedure that is repeated for each subtree rooted at the new node [19].

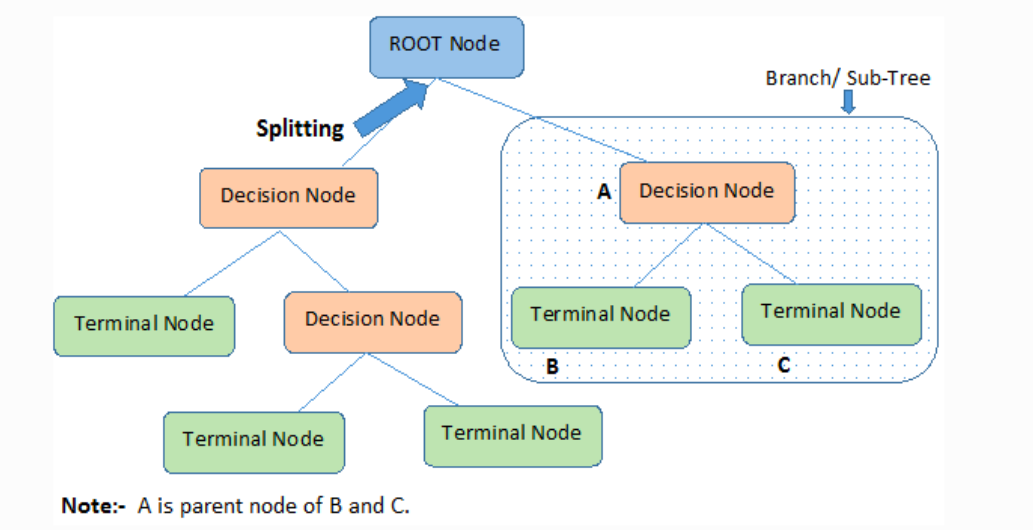

Figure 7. Example of DT algorithm. Source: [19].

4. Training a Decision Tree

Previously we mentioned that DTs are built by recursively splitting training samples using features from the data that are used in a specific task. This means that making the number of splits and strategic split criteria will heavily affects the accuracy of the tree. Figure 8 shows a basic idea of a DT algorithm that works with the given dataset, also known as "training set," used to train the model as well as to evaluate the performance of measurement e.g., accuracy, precision and recall. This process will be repeated recursively for all child nodes until one of the conditions is met.

However, the decision criteria are different for classification and regression trees. Nominal responses come from classification trees, while numeric responses come from regression trees. In this thesis, we will be focusing on using classification trees.

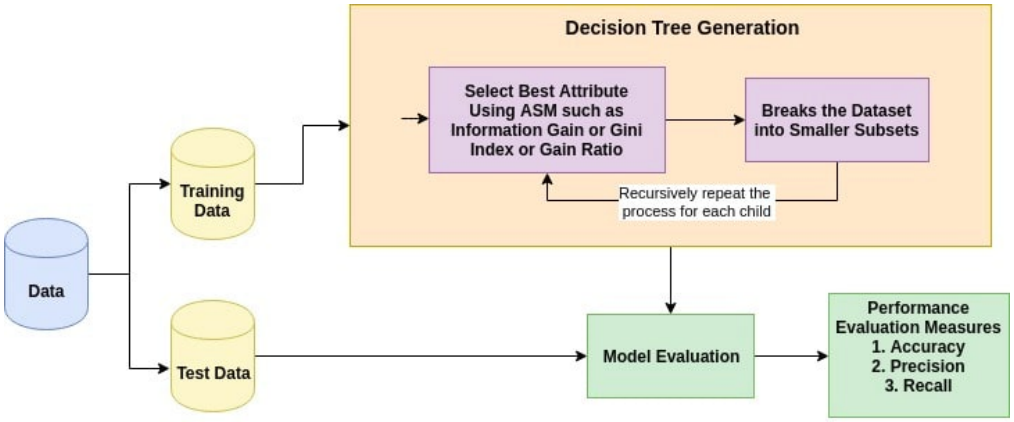

Figure 8. Basic block diagram of DT algorithm in data training. Source: [20].

Some parameters of note are:

a. Maximum number of splits

It is the number of splits in a tree model that must occur before a forecast can be made. This allows you to control for the maximum depth of the tree, which reduces over fitting. In [21], over fitting occurs when the learning algorithm keeps developing hypotheses that minimize training set error but raise test set error. Therefore, by having a higher number of splits, you will get higher accuracy on the training sets.

a. Split criterion

A split criterion is a measure for deciding when to split the nodes. Purity is used to determine whether or not to divide at each node. When a node data is divided evenly 50/50, it is 100 percent impure, and when all of its data belongs to a single class, it is 100 percent pure [22].

The purpose is to split a node into several sub-nodes in order to generate reasonably pure nodes [23]. Below are some of the DT split nodes based on impurity, node error and surrogate:

(1) Gini's Diversity Index (GDI)

GDI cost function is used to evaluate splits in a dataset. It assesses the likelihood of a given feature being wrongly categorized when randomly chosen [24].

Mathematically, it can be expressed as:

$$Gini's\ Index = 1 - \sum_{i=1}^{K}(Pi)^2, \qquad [1]$$

where, as explained by "K is the number of subsets generated by the split and Pi denotes the probability of an element being classified for a distinct class" [24].

(2) Maximum Deviance Reduction Selection (MDRS)

In [25], "Pi defined the same as for the Gini's index, the deviance of a node is":

$$E(S) = -\sum_{i=1}^{c} Pi\ log_2\ Pi \qquad [2]$$

where, as explained by "S represents the = current state and, P*i* is the probability of an event *i* of state S or percentage of class *i* in a node of state S" [25].

(3) Twoing rule (TR)

Twoing is a distinct method of determining how a node should be divided. It is not a purity measure of a node. According to [25], "let L(*i*) denote the fraction of members of class *i* in the left child node after a split, and R(*i*) denote the fraction of members of class *i* in the right child node after a split."

The twoing splitting rule will maximize the following impurity measure [25]:

$$P(L)P(R)(\sum_i | L(i) - R(i)|)^2 \qquad [3]$$

where, as explained by "*P(L)* and *P(R)* are the fractions of observations that split to the left and right respectively" [25].

(4) Node error

In [25], the node error is defined "as the fraction of misclassified classes at a node while letting j be the class with largest number of training samples at a node."

The node error is:

$$1 - P(j) \qquad [4]$$

b. Surrogate decision splits

A surrogate decision split is used to help deal with missing data and to identify variable importance. It tries to predict your actual split and improve the accuracy of the predictions [25].

C. PRINCIPAL COMPONENT ANALYSIS (PCA)

While dealing with a large dataset, we will face many predictor variables or features that make the computation heavy. This results in longer training time, which is undesirable. To overcome this issue, the dimensionality reduction techniques are very useful to use in reducing the number of features in the dataset. A popular technique for dimension reduction is called Principal Component Analysis (PCA). PCA is a mathematically rigorous statistical approach for reducing the dimension of a dataset by removing the variables with the least information about something we have anticipated and leaving the variables with the least information to accomplish simplicity [26]. It also increases interpretability, minimizing information loss as well as it takes a shorter time to train the model.

However, there are still some limitations with using PCA [27]: First, the datasets with no or poor feature correlation, or that do not fulfil the linearity assumptions, might result in a decrease in model performance. Second, the framework based on variance ignores the classes' distinguishing features, and low variance components may include information that separates one class from another; information may be ignored. Third, the outliers also have an impact; thus, data normalization should be a key part of any workflow. Fourth, each major component is made up of a variety of unique traits that make it impossible to determine the relative importance of individual aspects.

PCA is available in the MATLAB machine learning toolbox. It uses a Singular Value Decomposition (SVD) algorithm [28], which is a matrix factorizing technique where it can achieve dimension reduction through matrix decomposition and reduce the number of variables for computation.

D. DECISION TREE IMPLEMENTATION

1. Benefits of Decision Tree

DT is one of the most popular algorithms in ML. The benefits of DT are that it can quickly learn from a dataset by studying information about the system crucial features that reveal malicious activity. It uses a non-parametric approach that is not dependent on probability distribution assumptions and is distribution-free [29]. Moreover, it can promote to check on any advancement of attack signatures and different activities that have occurred while perceiving the data patterns. As a result, the value of various security frameworks is increased by examining the layout of intrusion detection information [17].

Unlike other classification algorithms, the decision tree uses non-linear relationships between parameters that does not influence the performance results. It also gives a rich arrangement of rules, which are simple, straightforward and require a shorter training period. The DT can also be easily integrated with the technologies in real-time applications.

2. Choice of Machine Learning Technique

In [30], a study based on a UAV and satellite-based 5G network security model experimented with multiple ML classifiers. The authors showed that for all types of attacks that were evaluated, the decision tree classifier achieved a minimum false negative rate of 0% and a maximum accuracy of 99.99%, better than other ML classifiers such as K-nearest Neighbors (KNN), K-means (K-M), Stochastic Gradient Descent (SGD), Logistic regression (LD) and Linear Discriminant Analysis (LDA) [30].

Also, in [31] Imam and Dovis, proposed the use of DT, “to distinguish between ionospheric scintillation and multi-path in GNSS scintillation data.” Their approach, which labels data as scintillated, multi-path impacted, or clean GNSS signal, was shown to be 96% accurate using DT.

Therefore, DT has several benefits. From an experimental viewpoint, it is preferred to use a linear, non-computationally heavy method like the DT as the foundation for cyberattack detection in an ISTN.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. PROPOSED ARCHITECTURE AND RESEARCH METHODOLOGY

In this chapter, we propose an ISTN architecture with various unmanned systems. We identify cyberattacks and their overall impact on the network architecture. We deploy NIDS at two key locations in the network. Our approach is built upon the use of DT algorithm to analyze traffic patterns, detect network anomalies and alert the network administrator when cyberattacks arise.

A. PROPOSED ISTN ARCHITECTURE AND NETWORK SETUP

The network architecture used in this thesis is based on a unique case network architecture presented in [3] by LTC Denny Cheng. We modify this architecture into a simpler ISTN with various unmanned systems. The ISTN architecture used in this thesis is shown in Figure 9.

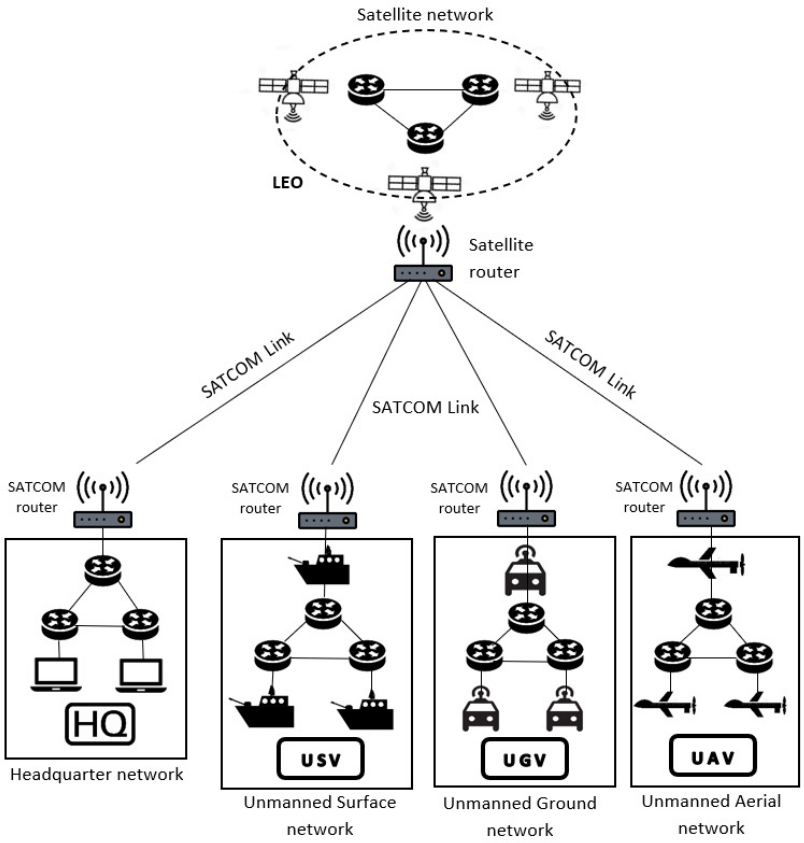

Figure 9. Proposed integrated satellite-terrestrial networking. Adapted from [3].

The network architecture is connected by the tree topology, which is often utilized in the building of satellite-terrestrial networks, as illustrated in Figure 9. It consists of a satellite network, headquarter network, USV network, UGV network, and UAV network. We can classify the headquarter network and different unmanned system networks as terrestrial domains, whereas the satellite network domain exists independently in space. Therefore, the network setup can be broken into two parts as follows:

1. Terrestrial Network

a. Headquarter Network

- The Mission Commander is stationed at headquarters (HQ) and is linked to the rest of the network via SATCOM router link then to a satellite router. This allows communications with the different unmanned systems at the mission site.
- It enables the Mission Commander to command and manage the different autonomous system platforms while at the same time monitoring their status at the mission site.
- A basic ring topology network consisting of one gateway, two routers, and two clients is used to demonstrate HQ On-site network communication and data sharing using TCP/IP protocol stack.

b. Unmanned Systems Networks

- For simple illustration, all three unmanned systems: (1) USV, (2) UGV, and (3) UAV are implemented by having a decentralized communication architecture based on the leader-followers formation architecture [32].
- As for real-time interactive communication, among the unmanned vehicles, a ring topology network is deployed.
- All leaders of the unmanned systems are connected to the satellite router via direct SATCOM Link communication to the satellite router and back to the HQ.

2. Satellite Network

- In a ring topology, three LEO satellites are positioned in the same orbit and networked. They must be visible at all times in order to communicate with the HQ and unmanned systems at the mission site.
- Through SATCOM Link, one satellite gateway is connected to the terrestrial network domain, serving as the HQ and the leaders of the unmanned systems at the mission site.

B. CYBERATTACK ON PROPOSED ISTN ARCHITECTURE

Before proceeding to introduce cyberattacks, we replaced all of the routers in Figure 10 with switches (From S1 to S5) in order to assess traffic flow over the whole network.

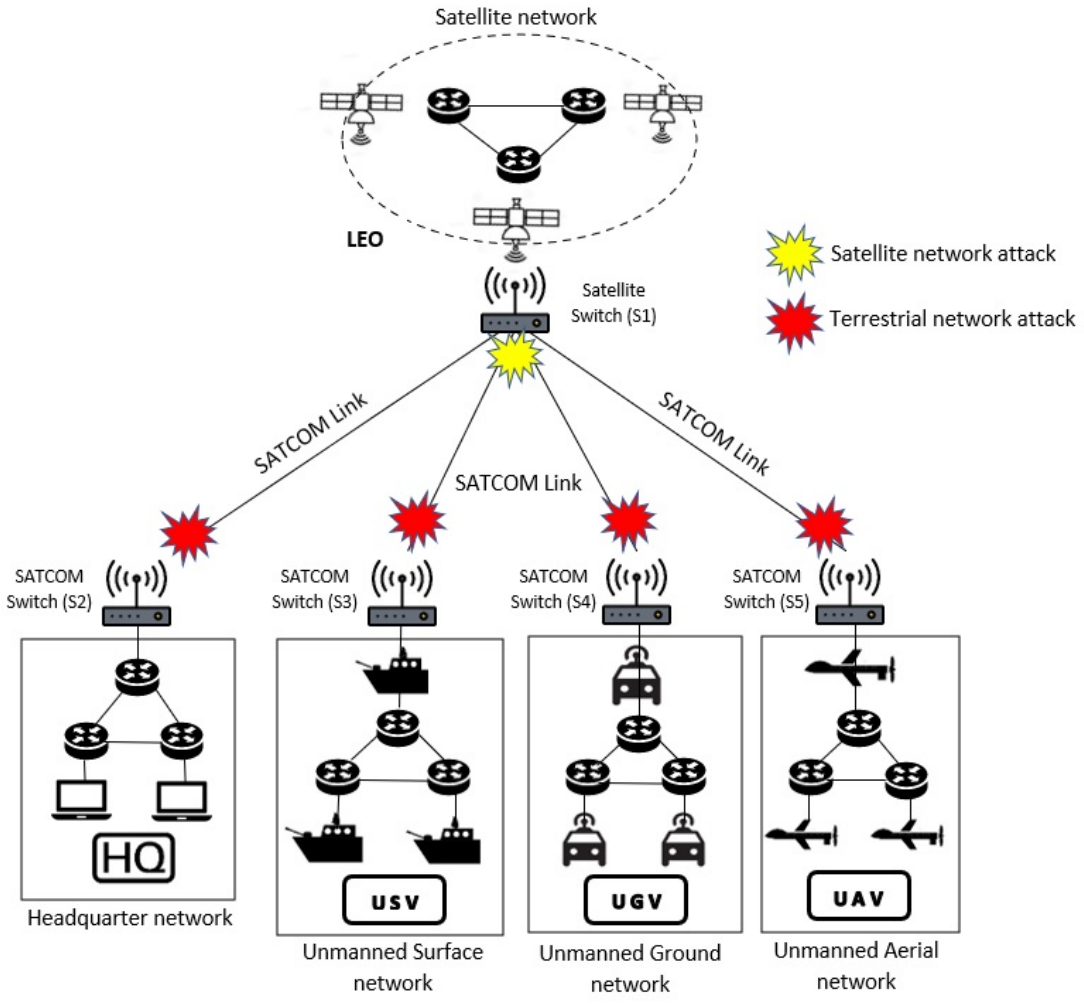

Figure 10. Network attacks on proposed ISTN

To accurately simulate the modern cyber threat environment, two attack hosts are setup in a different network segment as shown in Figure 14. One of the attack hosts will focus attacks on the terrestrial network, while the other will attack the satellite network. In Chapter II, we discussed the cyber security weaknesses of unmanned systems and satellite systems. As mentioned, DDoS attacks are one of the most significant security issues that affect satellite and terrestrial networks alike.

We have also incorporated various forms of network attacks as shown in Table 1 to replicate a real-world environment. These network attacks were discussed in [2]. The data set that we use for training our DT algorithm is also adapted from [2].

Table 1 shows that three hours of datasets were collected starting from 15:00 to 18:00 The attack times indicate the duration of the specific attacks during the simulation.

Table 1. Types of cyberattacks for ISTN. Source: [2].

Domain	Attacks	Attack Times
Terrestrial Network	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 Network	Syn_DDoS	15:23 → 15:57
	UDP_DDoS	16:52 → 17:20

C. PROPOSED SECURITY ARCHITECTURE

Our proposed security system includes a classifier and an anomaly detector which can identify both known and unknown threats quickly. This system should be able to initiate particular reactions in order to safeguard the network integrity, isolate the regions under attack, and promptly notify the server authorities. Figure 11 describes the proposed security architecture as a flow chart. The NIDS is identified within the flow chart.

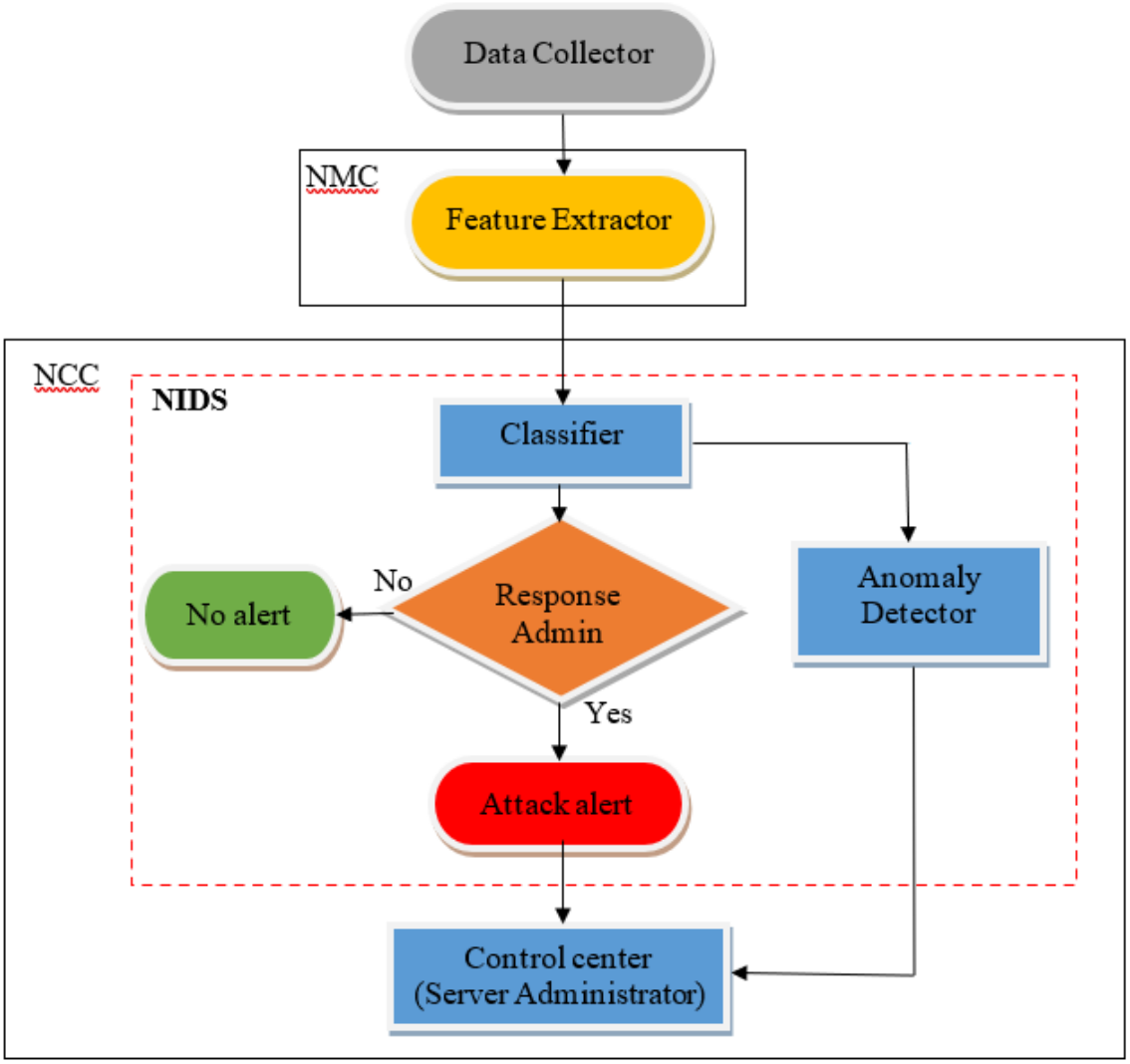

Figure 11. Flow chart of proposed security system

The data collector is designed to gather all types of traffic data from network units on a regular basis. The Network Control Center (NCC) is in charge of capacity allocation and terminal connections, while the Network Management Center (NMC) is in charge of traffic network management [10]. NIDS is then installed in the NCC software, which is responsible for monitoring and controlling the network status and behavior.

To implement the NIDS, we add an extra router to the network to act as a gateway. This router connects to the NCC and the NMC that receives data from the wireless communication network. The feature extractor in the NMC then collects gathered data, extracts NIDS-relevant characteristics, and delivers them to the classifier as structured vector data. The classifier module then examines if a particular vector is associated with a malicious attack or legal traffic.

However, the classifier is not perfect as it cannot pre-determine a new type of cyberattacks also known as zero-day attacks. Therefore, an anomaly detector is introduced to deal with these issues. Once it detects an anomaly it will update the control center for further investigation.

The DT algorithm will be used to make this categorization. If the classifier detects an attack, the system will alert the response administrator. The response administrator will then respond with various pre-planned responses and send an attack alert to notify the control center and to perform isolation quickly before infecting other networks.

Hence, NIDS is deployed at two key locations in the network. One is in the HQ, while the other is at the satellite network shown in Figure 12. We understand that if the HQ or satellite network system fails, we will be unable to establish connection with the mission many unmanned systems. Hence, these two locations are essential and critical to the entire network.

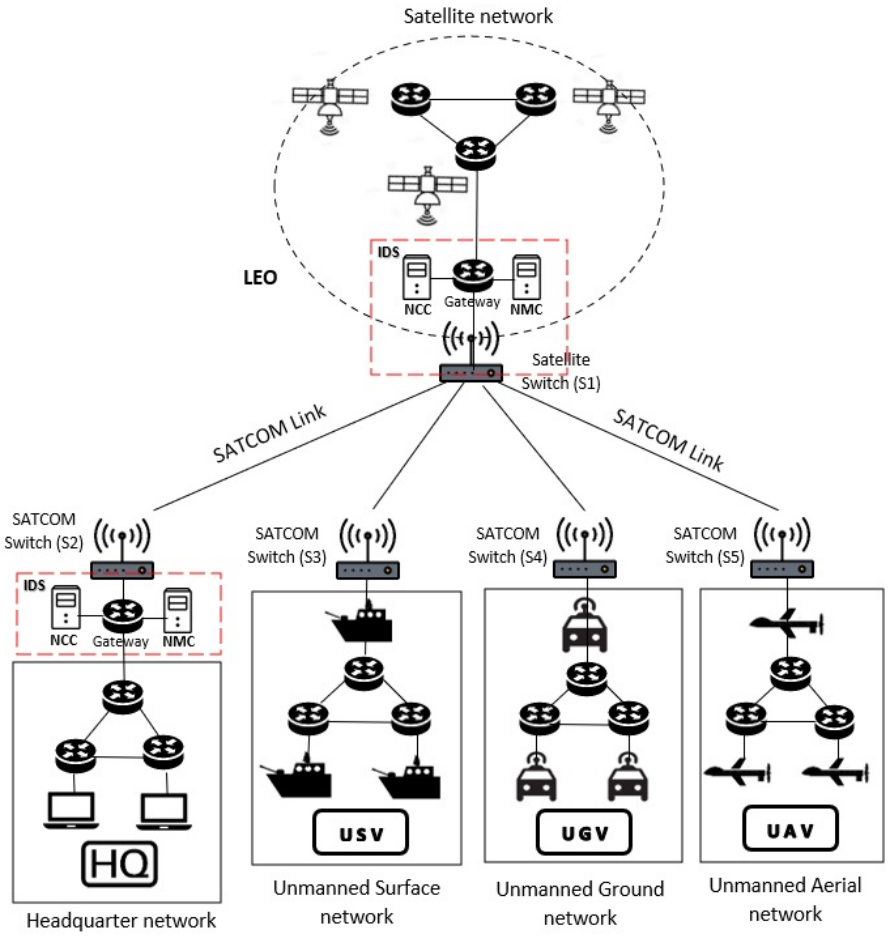

Figure 12. Two key locations of NIDS implemented on proposed ISTN

The remainder of this thesis focuses on the DT approach for realizing the classifier block and addressing the feature extractor criteria. One thing to keep in mind is that false triggering is highly disruptive and takes a long time to resolve, which is undesirable. As a result, the classifier false alert rate needs to be kept low.

D. CLASSIFICATION OF ATTACKS

Various cyberattacks should be categorized according to the expected reaction, since the classifier primary goal is to trigger suitable responses. Generally, there are three classes of attack proposed for analysis:

1. Terrestrial Network Attack

A terrestrial attack main goal is to go beyond a network normal security requirement while gaining access to sensitive data from network devices in order to inject malware into the network and take control of it as a whole. This allows attackers to launch cyberattacks by exploiting their security network. For this thesis, a total of nine various attack scenarios are created. This includes six different DDoS attacks, botnets, backdoors, and web attacks, as shown in Table 1.

2. Satellite Network Attack

In comparison to terrestrial networks, satellite networks provide a higher degree of security and a unique transmission protocol. This makes it harder for cyber attackers to enter and cause harm to the satellite network [2]. DDoS attacks on satellite networks only use Syn and UDP The cyberattacks simulated for the satellite network are shown in Table 1.

3. Benign

Benign is defined as the opposite of malicious cyberattack in networking. Any traffic that does not fit into one of the classifications is categorized as benign. Other than that, zero-day attacks that can’t be classified will be sent to the anomaly detector.

E. DATA SET USED

The efficacy of NIDS is determined by its ability to detect attacks, which necessitates a comprehensive data collection containing both normal and abnormal activities. Obtaining adequate datasets is challenging, especially because datasets tend to be proprietary. Many

studies use various open-source data sets such as CSE-CIC-IDS2018 [33], UNSW-NB15 [34], NSLKDD [35] etc.

However, due to the specific features of ISTN, these datasets and NIDS are often deployed for terrestrial networks and are hard to apply to satellite communications.

In [2], the authors generated two datasets named TER20 (177,244 data points) and SAT20 (132,320 data points). They utilized Tcpdump [36] "to collect network traffic in the form of packets and the Argus [37] and CICFlowMeter [38] tools to produce reliable features from the PCAP files," and they developed a delay and interruption tolerant network in the satellite networks.

F. DATA FEATURE SELECTION AND PREPARATION

Features of network flow packets may be retrieved in a PCAP file using Arugs [37] and CICFlowMeter [38]. Packet-based features and flow-based features were among them. While most attacks are DDoS, DDoS will have a substantial influence on the processing efficiency in the real analysis of a network [2]. Therefore, flow-level features from the database were used.

Fifteen of the best flow-level characteristics for intrusion detection were selected to characterize the various forms of benign and hostile traffic simulated in the prototype. These features are shown in Table 2.

Table 2. ISTN data set features. Source: [2].

#	Name	Description
1	fl_dur	Flow duration
2	fw_pk	Total packets in the forward direction
3	l_fw_pkt	Total length of forward packets
4	l_bw_pkt	Total length of backward packets
5	pkt_len_min	Minimum length of a flow
6	pkt_len_max	Maximum length of a flow
7	pkt_len_std	Standard deviation length of a flow
8	fl_byt_s	Packet bytes transmitted per second
9	bw_iat_tot	Total time between of two backward packets
10	bw_iat_min	Minimum time between of two backward packets
11	fw_hdr_len	Number of bytes used in forward packet header
12	bw_pkt_s	Number of backward packets per second
13	syn_cnt	Number of packets with SYN
14	urg_cnt	Number of packets with URG
15	bw_win_byt	Number of backward bytes in the initial window

Despite the fact that the dataset is immediately accessible, further preparation is required. To prepare the data for simulation, the following procedures were followed:

1. The data was scaled down into training and test sets. There should be no data overlap between the two sets, and the set size was chosen to optimize accuracy while minimizing training time. Eighty percent of the data was used for the training set and 20% of the data for the test set. Also, an estimation of 2:1 ratio of benign to malicious data was established. For every one attack entry, there are two entries of benign data.
2. To meet the requirements of each simulation specified in the next section, the data was combined into training set files.

THIS PAGE INTENTIONALLY LEFT BLANK

V. EXPERIMENT AND EVALUATION

To implement our NIDS, the MATLAB Classification Learner App is used to train and test the DT. This chapter discusses and summarizes the results that were obtained by using these DT split criteria algorithms.

A. SIMULATION

1. General MATLAB Settings

The reader can utilize a comparable simulation environment by following the instructions in the Appendix on how to set up and run using the MATLAB Classification Learner App. Table 3 contains a list of general parameters.

Table 3. General MATLAB parameters used

Model Type Preset	Decision Tree-Fine
Validation	Holdout-Percent held out: 20
Number of splits	100
Split criterion	Varied base on simulation run
Surrogate decision splits	Off
PCA	On
PCA Variance	99%

2. Simulation Runs

For each simulation run, the model type is fixed using Decision Tree (Fine). The number of splits remains 100, all validation sets will be selected at a 20% of the data due to large datasets being used, and the PCA feature will be enabled for all runs. However, three different split criterions will be used separately to identify which one gives the best performance to recognize various malicious attacks.

To determine and compare the effectiveness and efficiency of the various setting, we consider the following measures:

1. Overall Accuracy: Percentage of observations that are correctly classified made to the size of datasets.

2. True Positive Rate (TPR): The proportion of correctly classified observations per dataset to determine as an intrusion or normal packet.

3. False Negative Rate (FNR): The proportion of incorrectly classified observations per dataset to determine as an intrusion or normal packet.

The simulations used in this thesis are as follows:

a. Benign-Terrestrial Network Attack (TNA)

This set of simulations is to ensure a TNA type of attack can be discerned from the benign traffic.

b. Benign-Satellite Network Attack (SNA)

This set of simulations is to ensure an SNA type of attack can be discerned from the benign traffic.

c. Benign-Combined (TNA and SNA)

The goal of the combined simulation is to demonstrate that a multi-attack classifier with adequate accuracy may be used as part of the NIDS.

B. RESULTS AND APPLICATION

We use a confusion matrix to depict our experimental findings. An example of a confusion matrix is given in Figure 13.

True positives indicate that the attack packet was properly anticipated as attack, whereas false positives indicate that the benign packet was mistakenly forecasted as an attack packet. A true negative indicates that the benign packet is accurately anticipated as such, whereas a false negative indicates that the actual attack packet is wrongly predicted as benign.

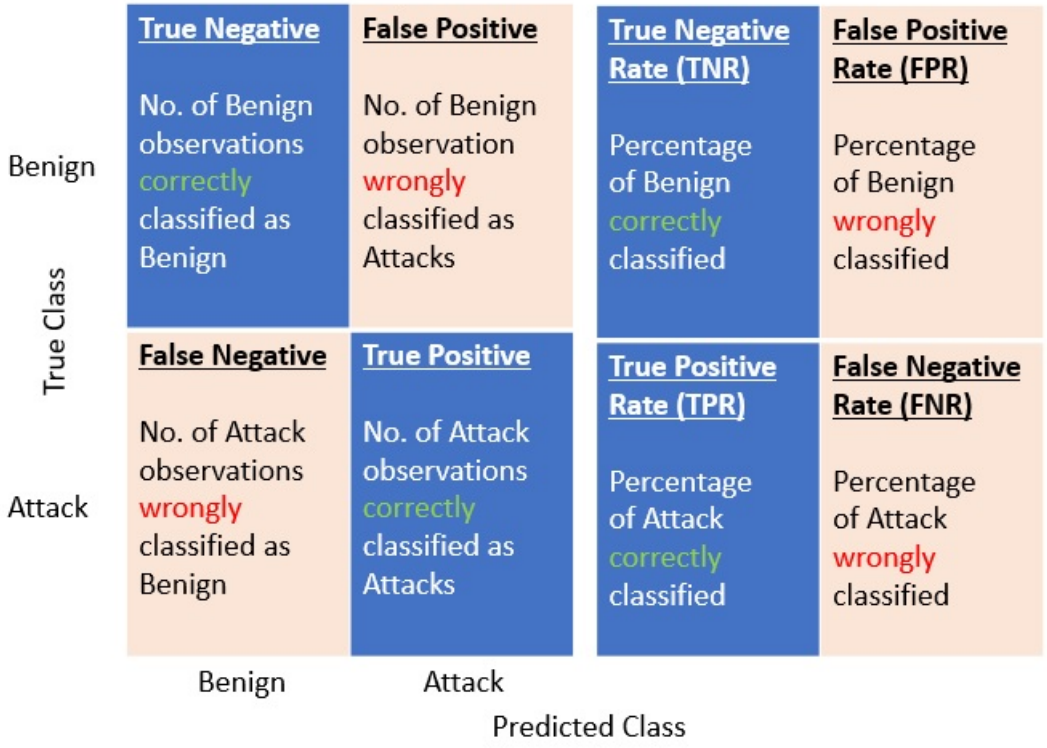

Figure 13. Example of a confusion matrix

1. Benign-Terrestrial Network Attack (TNA)

Three runs were performed based on the general settings except the split criterion have been changed. We used the following split criterion: (1) Gini's diversity index (GDI) [24], (2) Twoing rule (TR) [25] and (3) Maximum deviance reduction selection (MDRS) [25].

Figure 14, using GDI, shows that TNA can be discerned from Benign data with an overall accuracy of 98.6% probability of detection and a false positive rate of 1.4% for Benign and 4.1% for TNA. Figure 15, using TR, shows the same results as the GDI split criterion. Figure 16, using MDRS, shows that TNA can be discerned from Benign data with an overall accuracy of 99.6% probability of detection and a false positive rate of 1.0% for Benign and 5.0% for TNA. The results show that using MDRS gives us a better overall accuracy probability of 1% detection but the false positive rate for Benign is 0.4% lower and for TNA the false positive rate is 0.9% higher than the other two split criterion.

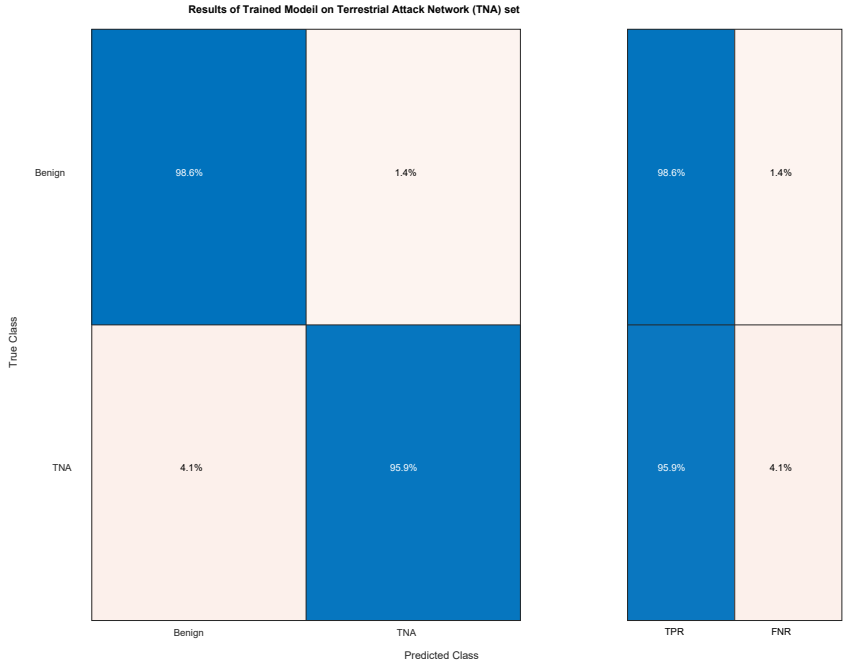

Figure 14. Using GDI: Results from benign-terrestrial network attack model

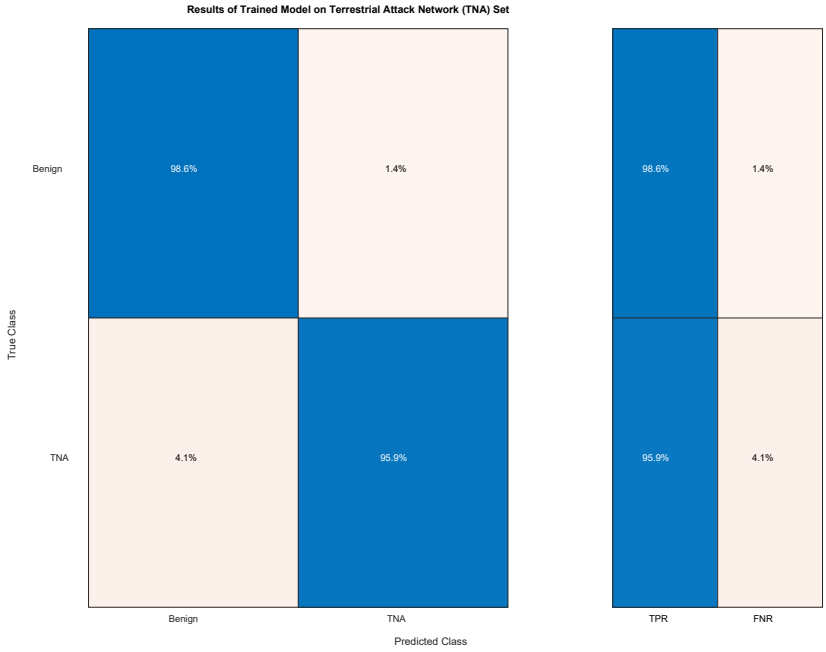

Figure 15. Using TR: Results from benign-terrestrial network attack model

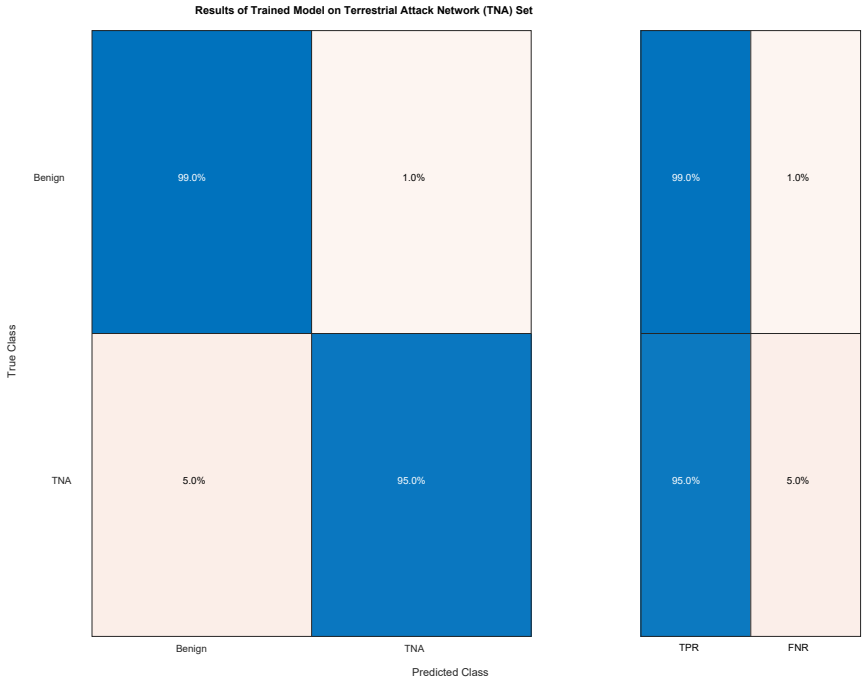

Figure 16. Using MDRS: Results from benign-terrestrial network attack model

2. Benign-Satellite Network Attack (SNA)

Figure 17, using GDI, shows that SNA can be discerned from Benign data with an overall accuracy of 98.1% probability of detection and a false positive rate of 1.9% for Benign and 4.9% for SNA. Figure 18, using TR, shows the same results as GDI split criterion. Figure 19, using MDRS, shows that SNA can be discerned from Benign data with an overall accuracy of 98.3% probability of detection and a false positive rate of 1.7% for Benign and 5.2% for TNA. The results show that using MDRS give us a better overall accuracy of probability of detection by 0.2% but the false positive rate for Benign is 0.2% lower and for SNA the false positive rate is 0.3% higher than the other two split criterion.

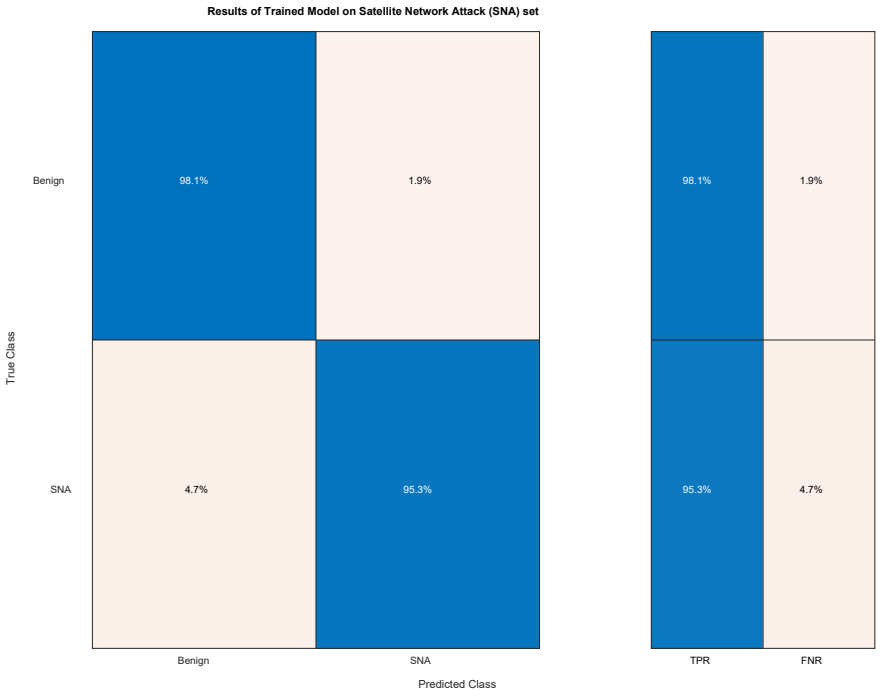

Figure 17. Using GDI: Results from benign-satellite network attack model

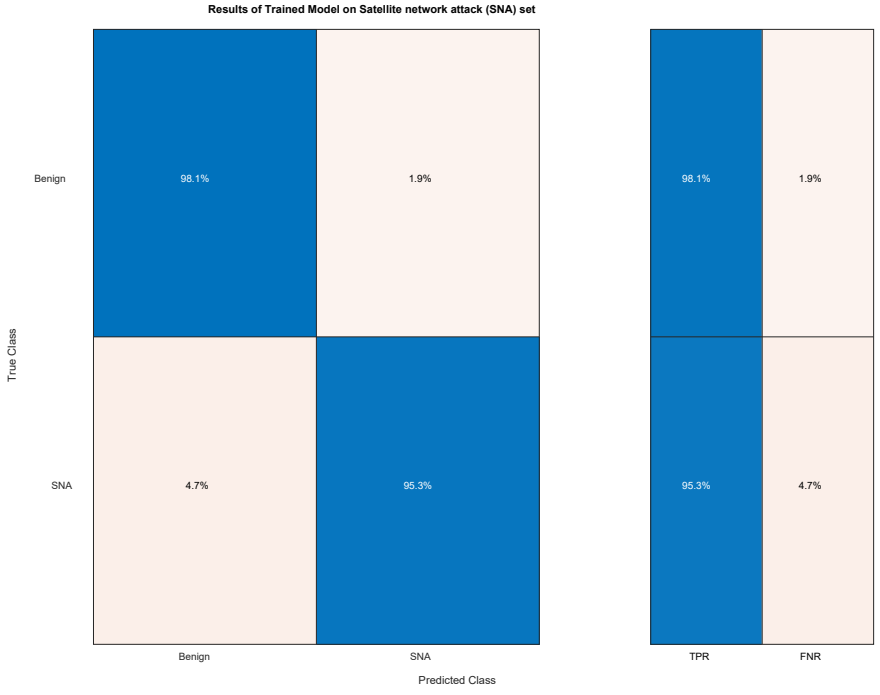

Figure 18. Using TR: Results from benign-satellite network attack model

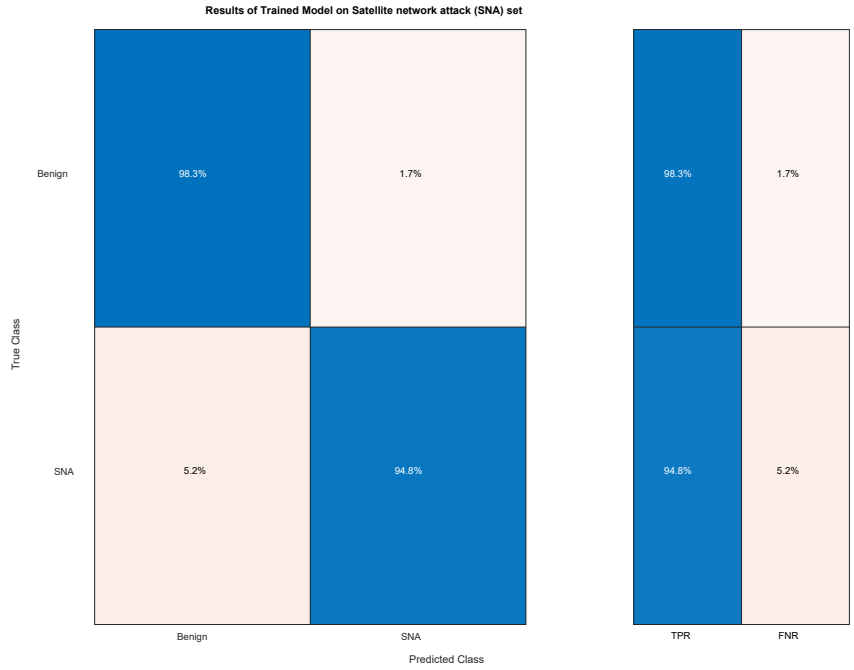

Figure 19. Using MDRS: Results from benign-satellite network attack model

3. Benign-Combined (TNA and SNA)

Figure 20, using GDI, shows that a combined attack can be discerned from Benign data with an overall accuracy of 91.7% probability of detection and a false positive rate of 3.7% for Benign, 11.7% for NetBIOS_DDoS attack and 12.3% for a Botnet attack. The training time took 1.6596 sec. Figure 21 shows the number of observations for the combined-individual attacks (using GDI).

Figure 22, using TR, shows that a combined attack can be discerned from Benign data with an overall accuracy of 91.0% probability of detection and a false positive rate of 3.3% for Benign, 5.7% for NetBIOS_DDoS attack and 45.4% for Botnet attack. The training time took 1.2823 sec. Figure 23 shows the number of observations for combined-individual attack (Using TR).

Figure 24, using MDRS, shows that a combined attack can be discerned from Benign data with an overall accuracy of 91.2% probability of detection and a false positive rate of 3.2% for Benign, 18.0% for NetBIOS_DDoS attack and 44.6% for Botnet attack. The training time took 1.1948 sec. Figure 25 shows the number of observations for combined-individual attacks (using MDRS).

The results show that by using GDI the overall accuracy is the highest at 91.7% probability of detection to discern a combined attack from Benign traffic. However, using the MDRS, the false positive rate for Benign is the lowest at 3.2%. However, for the NetBIOS_DDoS attack, it has the highest false positive rate at 18.0%.

It is also shown that for a Botnet attack, using TR, it has the highest false positive rate at 45.4%. Lastly, using MDRS has the shortest time for training.

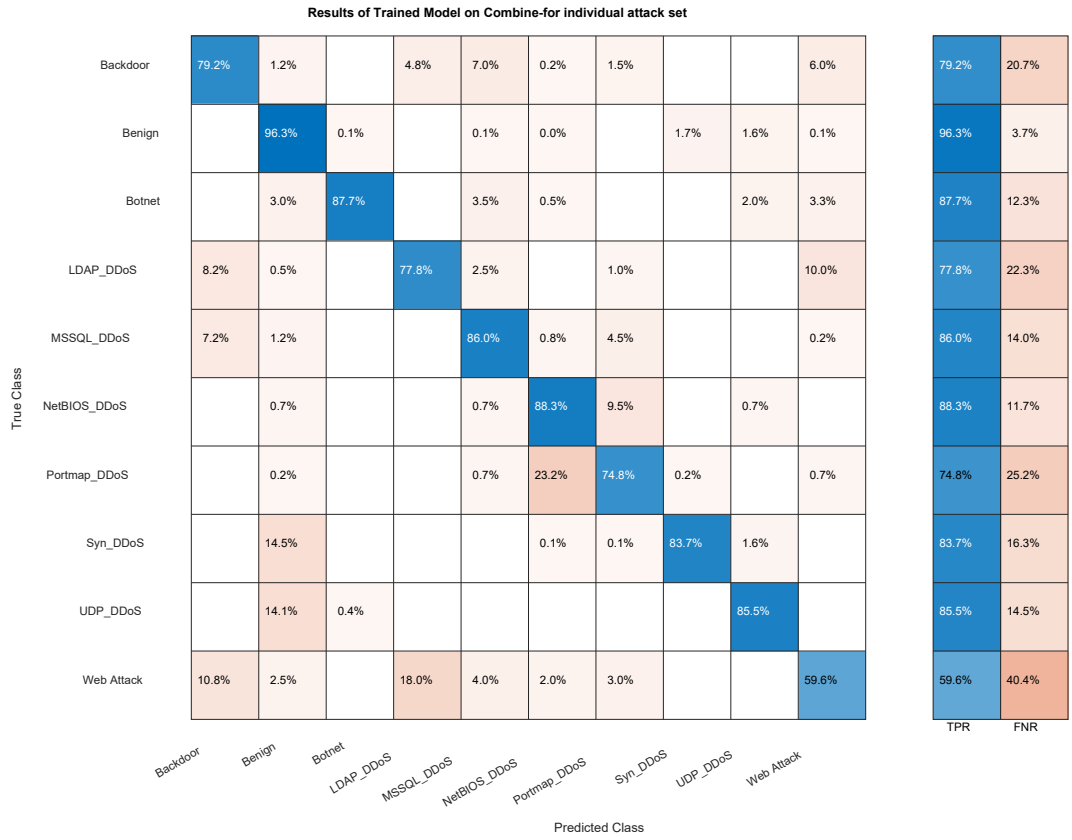

Figure 20. Using GDI: Results from benign-combined attack model

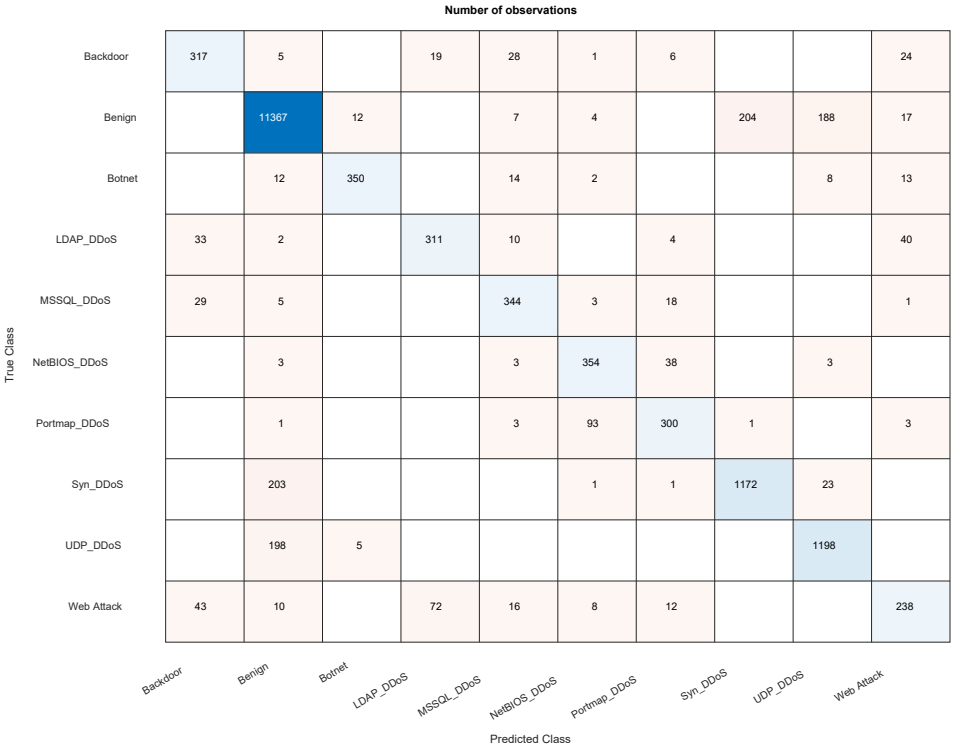

Figure 21. Number of observations from GDI

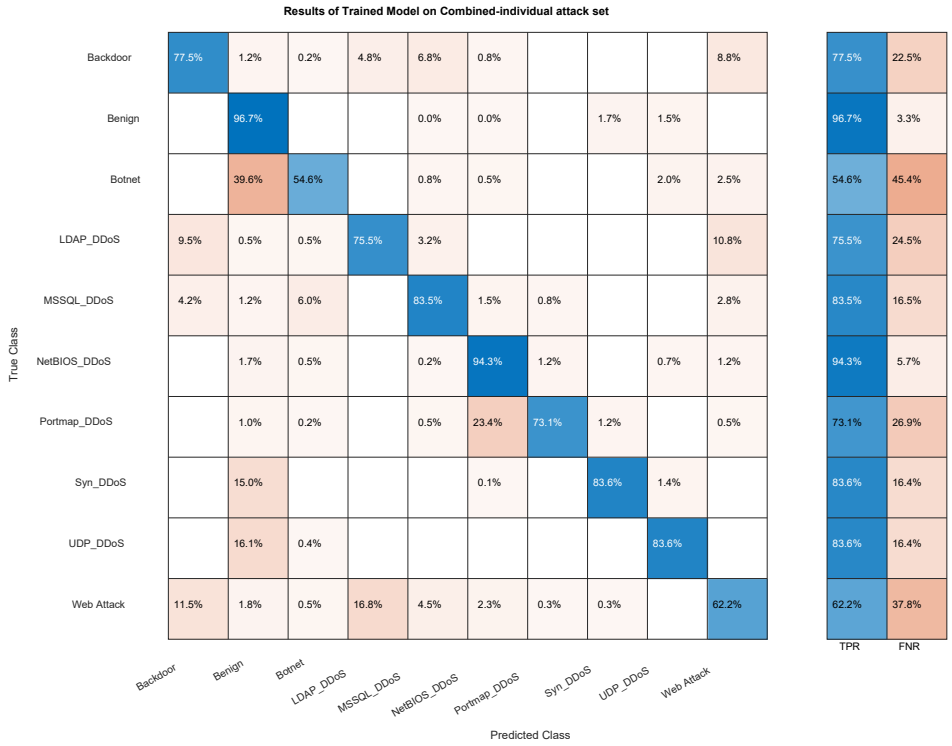

Figure 22. Using TR: Results from benign-combined attack model

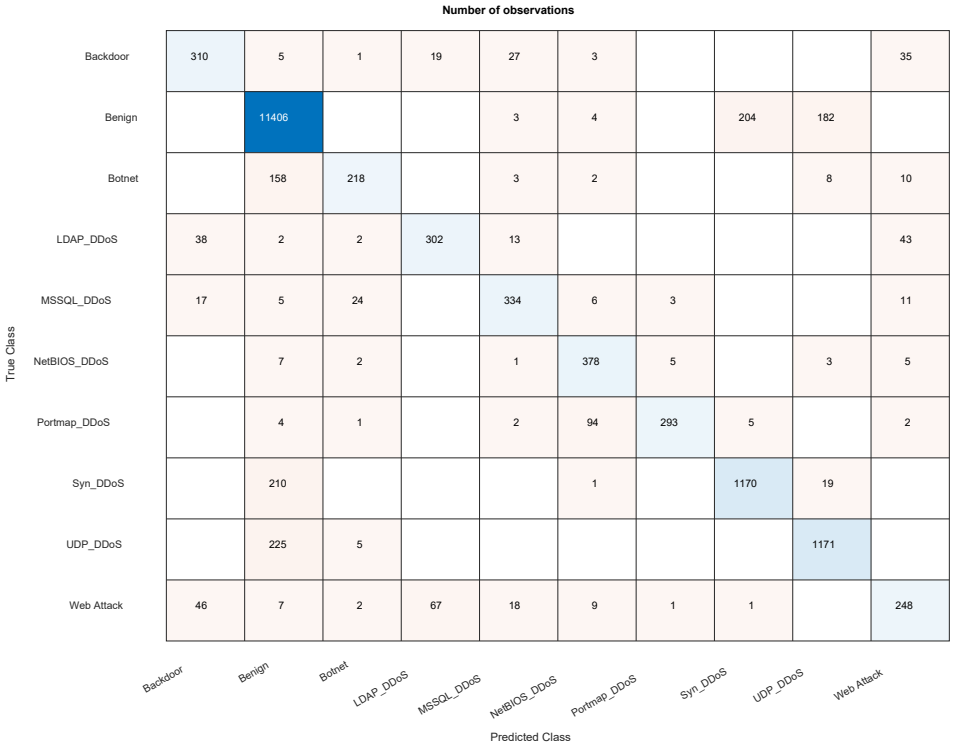

Figure 23. Number of observations from TR

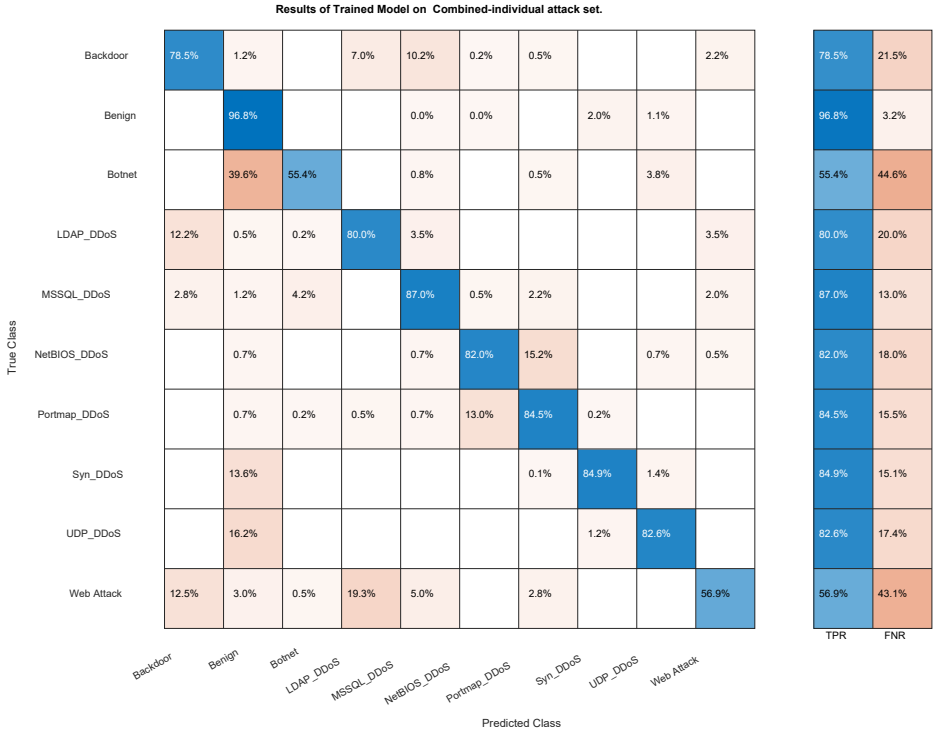

Figure 24. Using MDRS: Results from benign-combined attack model

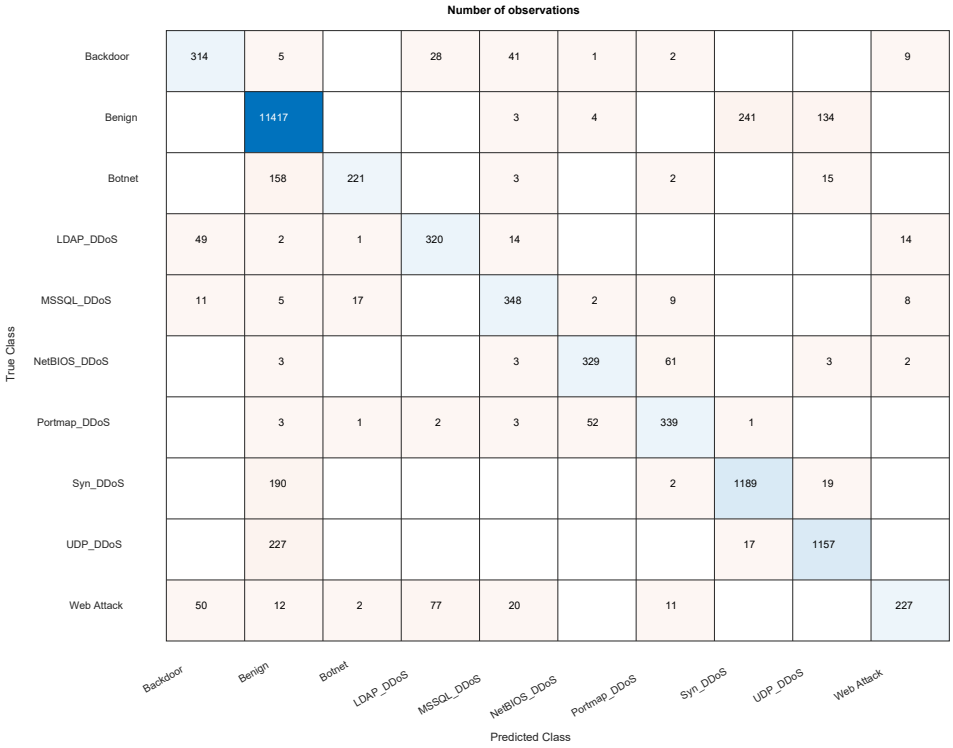

Figure 25. Number of observations for MDRS

C. WITHOUT PCA

The datasets for combined-individual attacks were used to train for three different split criterions, but PCA was disabled in these experiments. The results shown in Figure 26 are for GDI and TR. Both show that the overall accuracy has risen to 100% compared to PCA, which is 91.7% for GDI and 91.0% for TR. Both have significantly dropped in the false positive rate, which is 0.3% for Botnet, 0.2% for NetBIOs_DDoS and Portmap_DDoS.

Figure 27 shows the results for MDRS without PCA, which is similar to GDI and TR. It also rose to 100% for the overall accuracy probability of detection to discern Combined-Individual Attack from Benign, and the false positive rate for Syn_DDoS is 0.1%. We also noticed that the training times are reduced for all three split criterions: for GDI, 3.87 seconds to 1.13 seconds; for TR, 2.78 seconds to 2.24 seconds; for MDR, 2.17 seconds to 1.28 seconds.

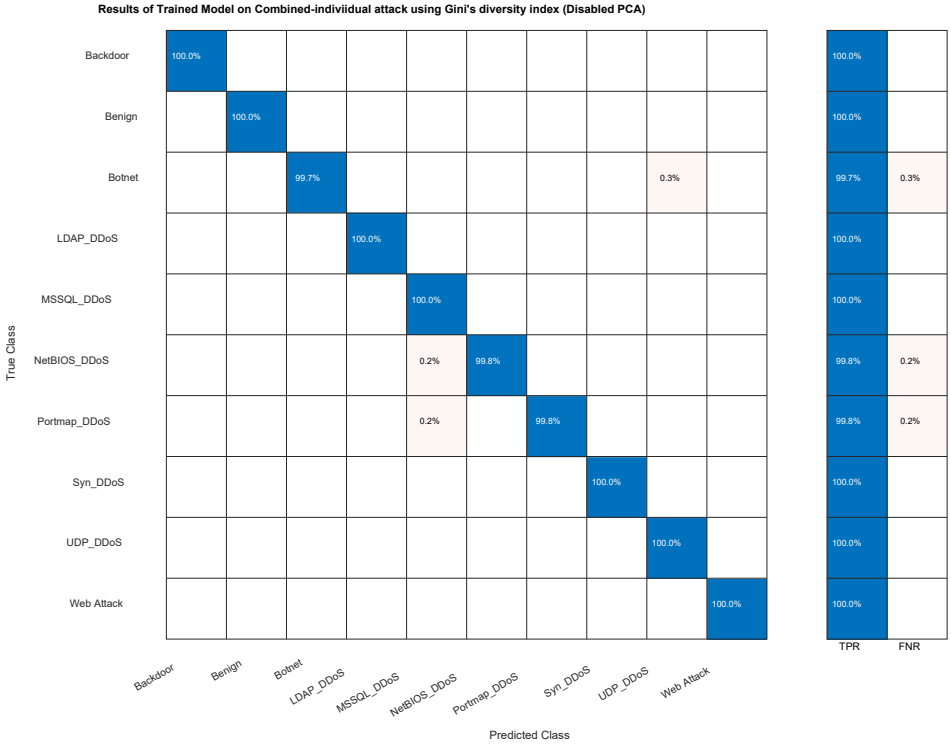

Figure 26. Using GDI and TR (Disabled PCA): Results from benign-combined attack model

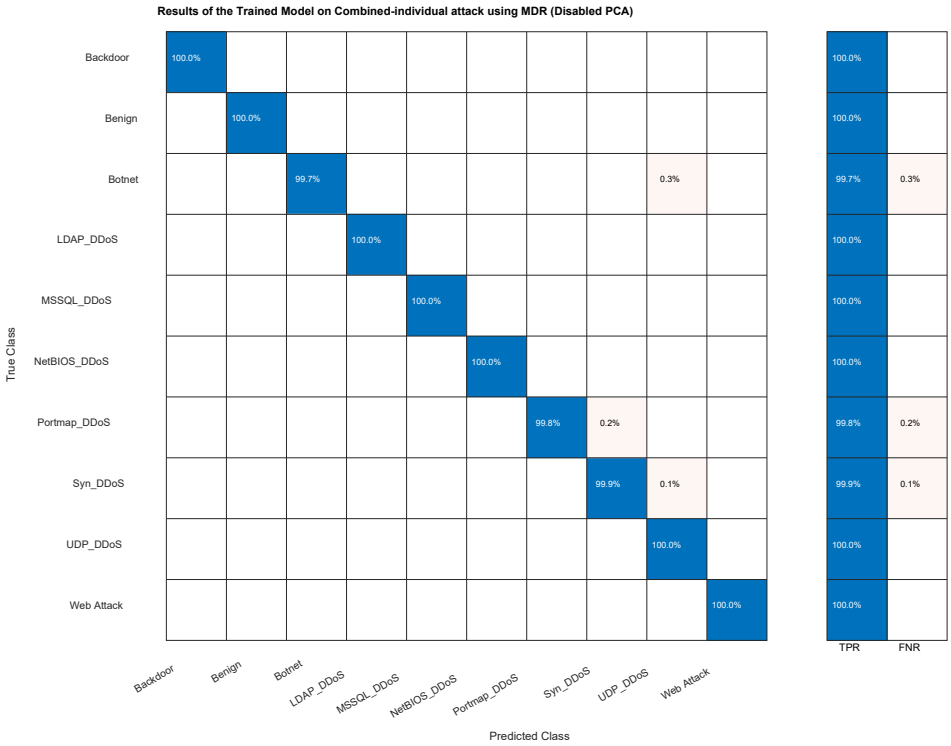

Figure 27. Using MDRS (Disabled PCA): Results from benign-combined attack model

D. INCREASE NUMBER OF SPLITS

In our next set of experiments, we increased the number of splits from 100 to 200. Figure 28, using GDI, shows improvement in the overall accuracy from 91.7% to 93.3% as well as the false positive rate through the different attacks over Figure 20. Figures 29 and 30, using TR, and MDRS, respectively also show improvement in the overall accuracy from 91.0% to 93.2% and 91.2% to 93.0%, respectively.

Therefore, the overall results show that by increasing the number of splits for all three splits criterion, we are able get higher accuracy on the training sets.

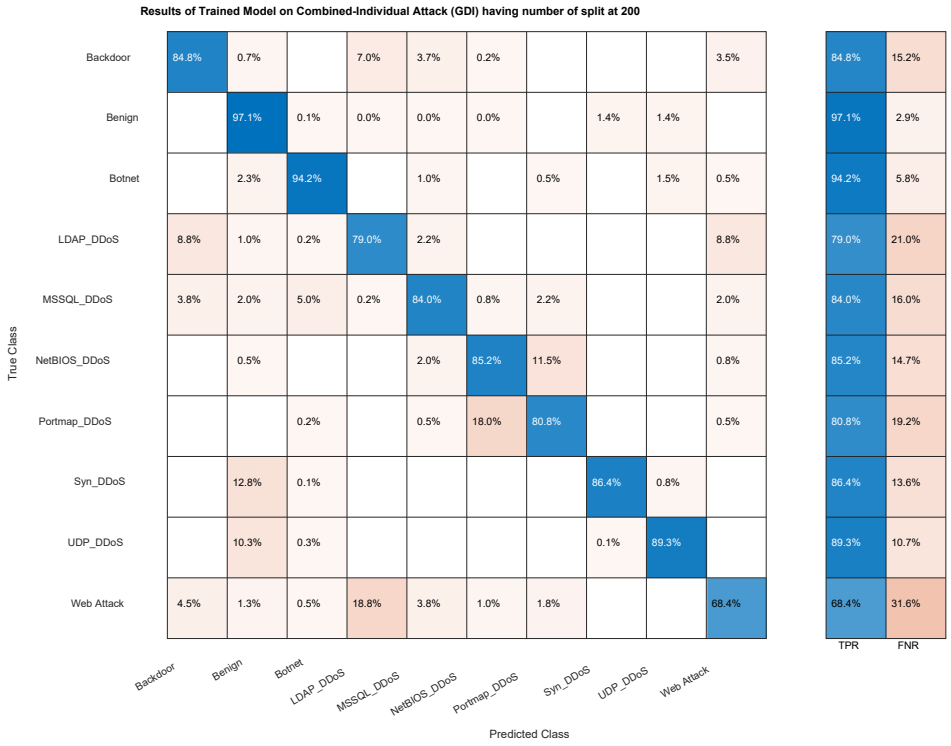

Figure 28. Using GDI (Number of splits at 200): Results from benign-combined attack model

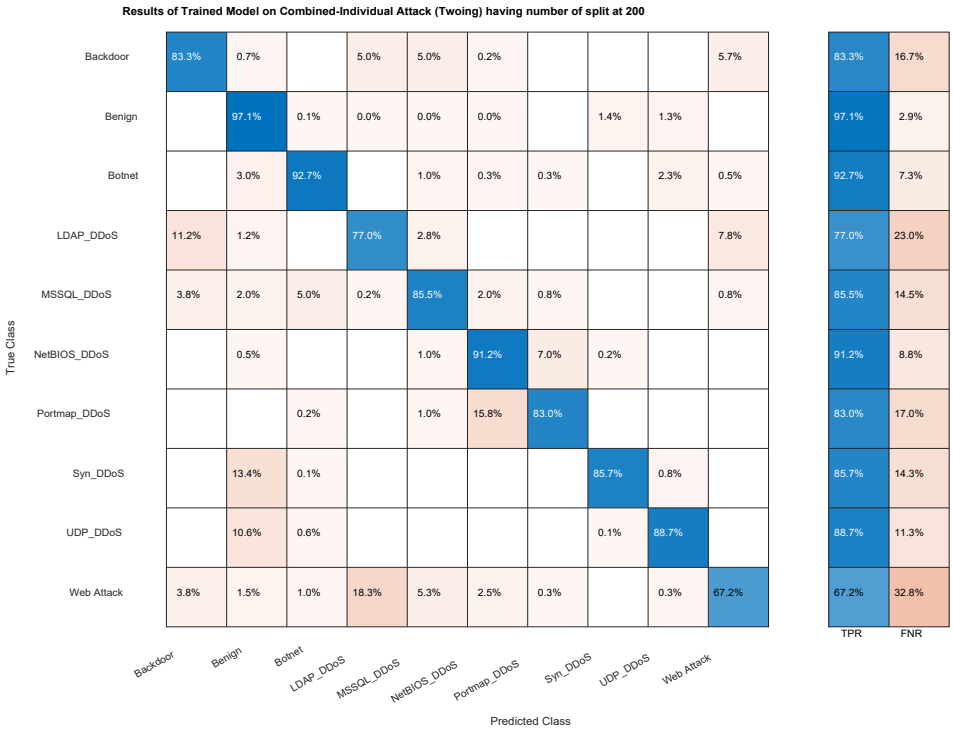

Figure 29. Using TR (number of splits at 200): Results from benign-combined attack model

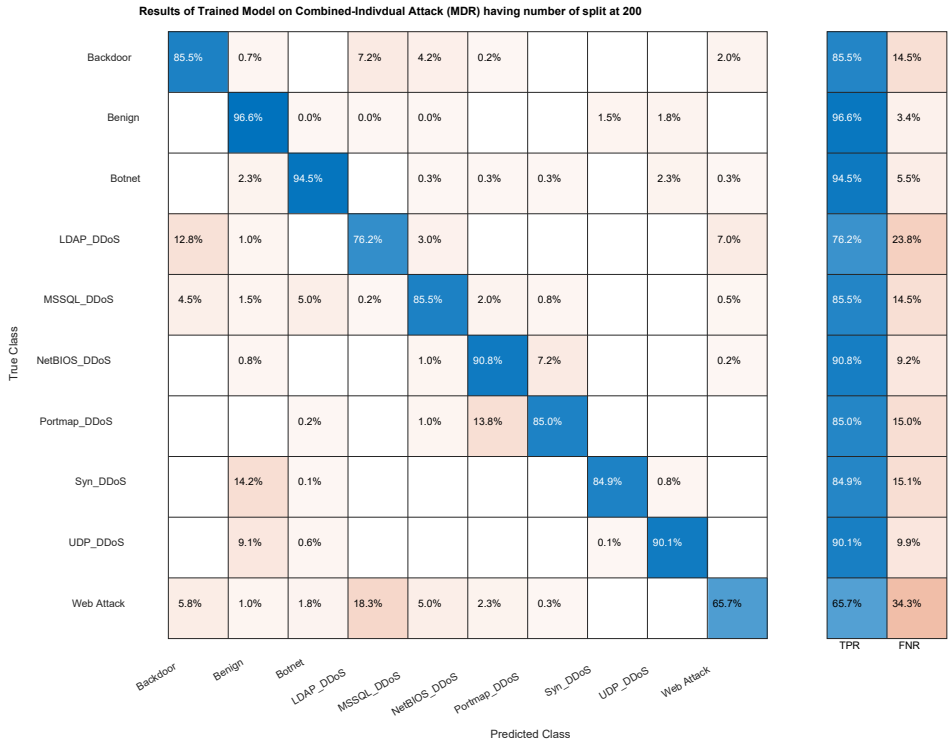

Figure 30. Using MDRS (number of splits at 200): Results from benign-combined attack model

E. SUMMARY OF ANALYSIS

In Table 4, we show the overall accuracy for the simulation runs for DT (Fine) machine learning algorithm using three different split criterions. The results show that MDRS performs the best when it comes to detecting either benign-terrestrial or benign-satellite attacks with an overall accuracy of 99.6% and 98.3% respectively. However, when it comes to detecting benign-combined attacks, GDI performs slightly better than MDRS with an overall accuracy of 91.7%.

Table 4. Overall accuracy

Experiments	Split criterion		
Using Decision Tree (Fine)	Gini's diversity index (GDI)	Twoing rule (TR)	Maximum deviance reduction selection (MDRS)
Benign-Terrestrial Network Attack	98.6%	98.6%	**99.6%**
Benign-Satellite Network Attack	98.1%	98.1%	**98.3%**
Benign-Combined	**91.7%**	91.0%	91.2%

In Table 5, we show the overall accuracy for the simulation runs for DT (Fine) machine learning algorithm using three different split criterions with an increasing number of splits from 100 to 200. The results show that GDI performs the best when it comes to detecting benign-combined attacks with an overall accuracy of 93.3% compared to TR and MDRS, which are slightly lower.

Table 5. Overall accuracy for benign-combined (with an increase of number of splits from 100 to 200)

Experiments	Split criterion		
Using Decision Tree (Fine)	Gini's diversity index (GDI)	Twoing rule (TR)	Maximum deviance reduction selection (MDRS)
Benign-Combined	**93.3%**	93.2%	93.0%

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION AND FUTURE WORKS

Existing ISTNs with various unmanned system are prone to security and privacy issues. Once an unsecure ISTN infrastructure is place in a network, the ISTN can be easily exposed to cyberattacks. Therefore, establishing a robust and reliable cybersecurity defense is critical within the ISTN architecture.

A. CONCLUSION

Our objective was to build a simple ISTN with various unmanned system by modifying the unique case architecture network presented in [3], followed by using the generated datasets for ISTN from [2]. Our goal was also to build a NIDS using the Decision Tree machine learning algorithm while identifying the critical location of NIDS in the ISTN architecture.

Therefore, in this thesis, we have completed the following work:

1. We have proposed a security system for an ISTN with various unmanned systems in which a NIDS is incorporated with an extra router, whereby it is added to the network. The router acts as a gateway, connecting to a Network Control Center (NCC), and a Network Management Center (NMC). The NIDS is within the NCC that is comprised of a data collector, feature extractor, classifier, anomaly detector, and response administrator to allow the network architecture to react according to the cyberattack threat.
2. We have identified the NIDS be placed at two critical locations within the ISTN. This implementation allows the ISTN security system to react accordingly to different cyber threat vectors.
3. We implemented the attack classifier using the DT machine learning algorithm. We trained the classifier using an open-source ISTN dataset.
4. We showed that a classifier based on the DT algorithm, using different split criteria, can separate attack traffic from benign traffic using MATLAB machine learning toolbox.

B. FUTURE WORK

1. Implement Other Supervised Machine Learning Algorithms

In this thesis, only one supervised machine learning algorithm was used for training and to classify different cyberattack types for an ISTN. One might not able to prove that DT machine learning algorithm is the best classifier to use for ISTN. Therefore, one suggestion is that by using multiple supervised Machine Learning algorithms and comparing with one another, could result in identifying the best classifier to use. Unsupervised learning approaches should also be investigated.

2. Generate and Obtain More Datasets for ISTN

Due to the low number of datasets for ISTNs that were publicly available, we were only able to use one of the datasets available for analysis and to run our simulations. In the future, it will be beneficial to generate our own ISTN dataset with unmanned system or to get different datasets that can be used to train with the DT machine learning algorithms for implementation of NIDS.

APPENDIX A. USING THE MATLAB CLASSIFICATION LEARNER APP

The appendix objective is to provide a guideline how to use the MATLAB classification learner app. The figures are provided using version 2020a.

Step 1. Once MATLAB program is loaded, click on the "APPS" tab and then click on the "show more" to locate "Classification Learner"

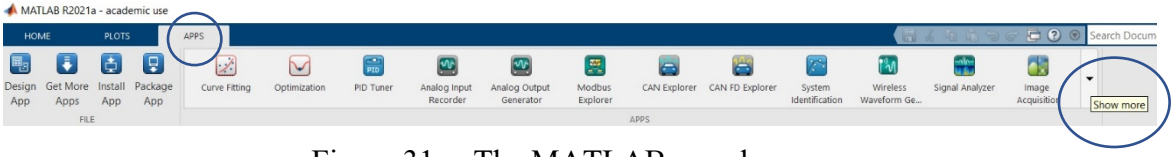

Figure 31. The MATLAB apps bar

Step 2. Click on the "Classification Learner" under ML and deep learning bar

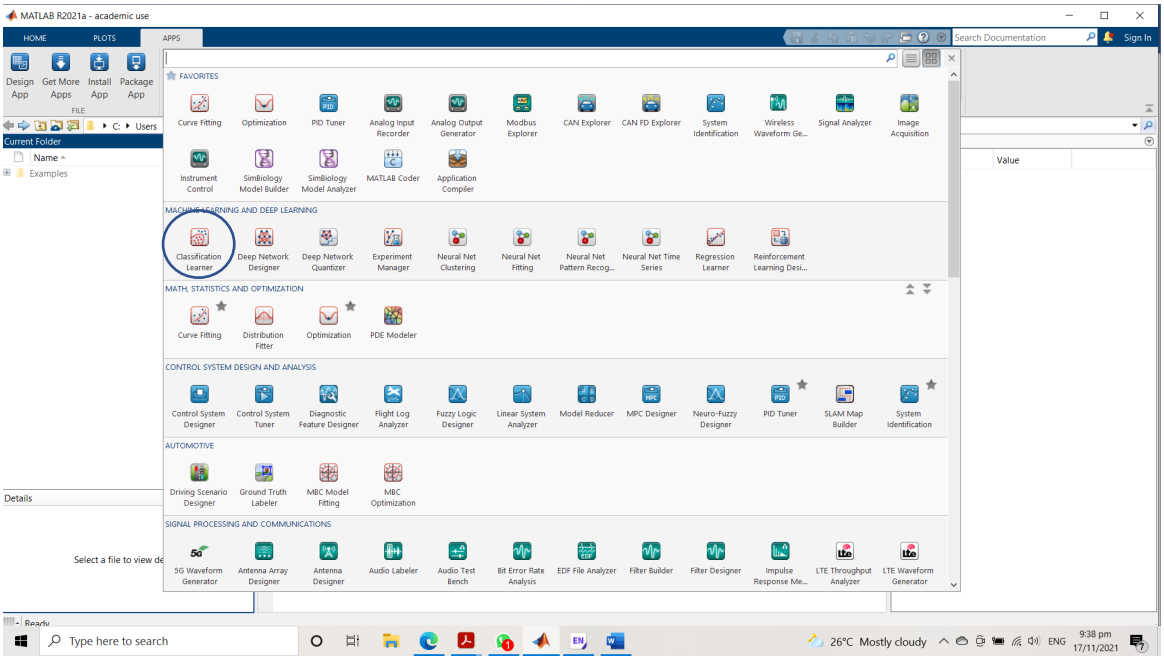

Figure 32. Locating classification learner app

Step 3. Click on the "New Session," select "From File" and upload the CSV file for training.

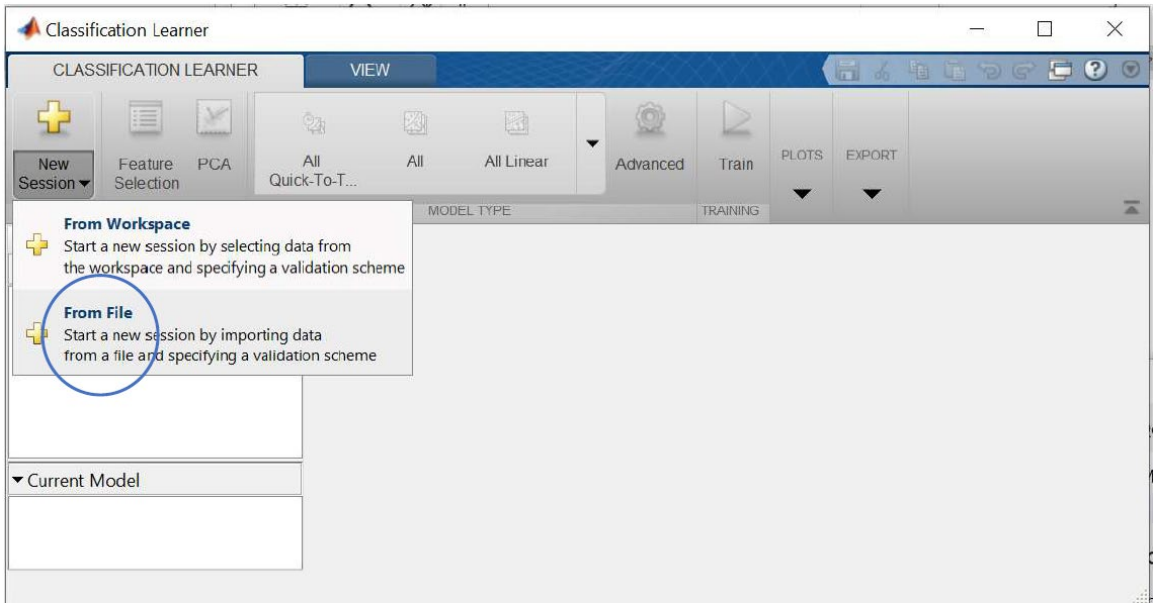

Figure 33. Starting a new session

Step 4. Select “Import Selection”

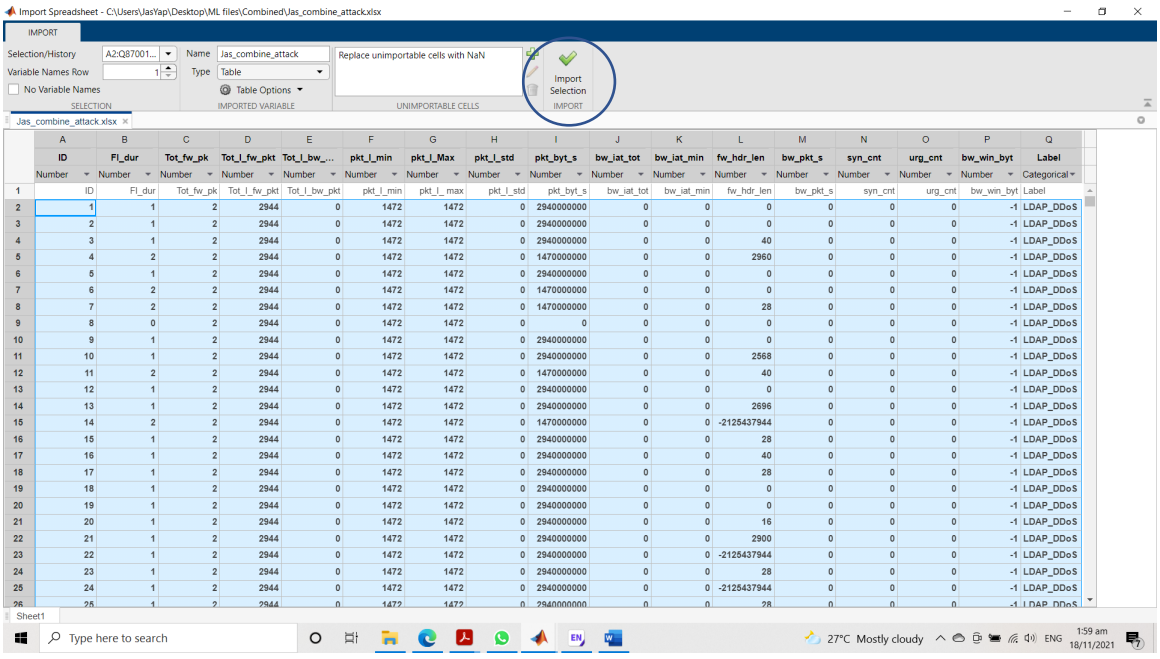

Figure 34. Importing data in MATLAB

Step 5. Set your validation settings then click “Start Session.”

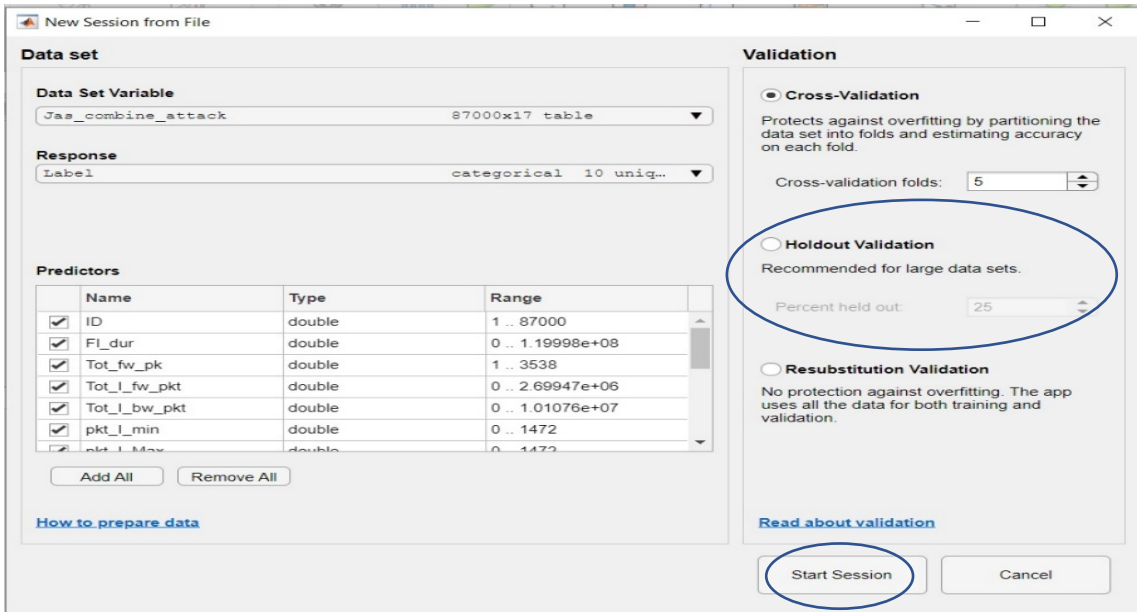

Figure 35. Validation method selection

Step 6. In drop-down window. Select your ML technique. (Fine Tree)

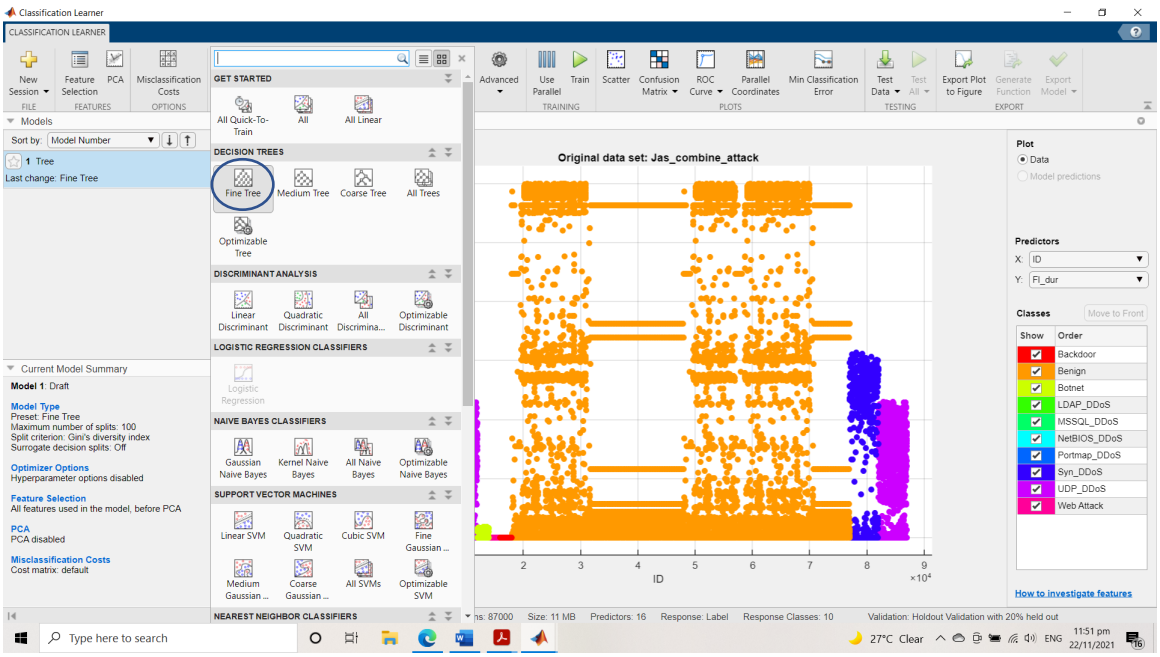

Figure 36. Selecting the machine learning technique

Step 7. Click “Advanced” to input the desired maximum number of splits and three different types of split criterion accordingly with Surrogate decision splits as “Off.”

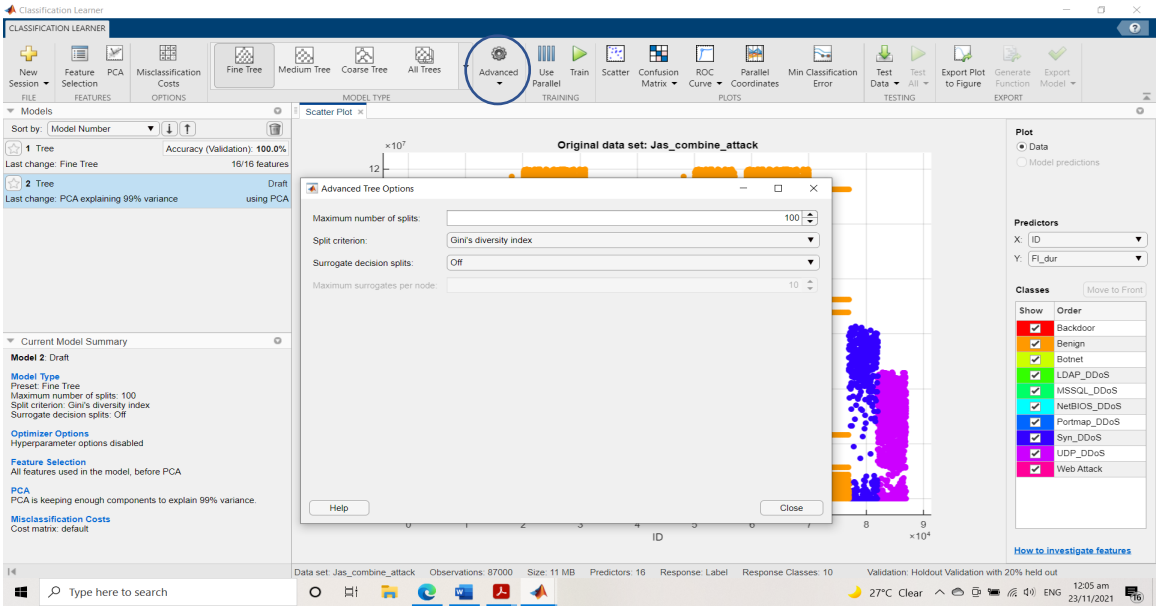

Figure 37. Advanced tree options

Step 8. Click “PCA” tick to enable PCA. Select component reduction criterion as “Specify explained variance” and “Explained variance” as 99 percent. Then click “Train.”

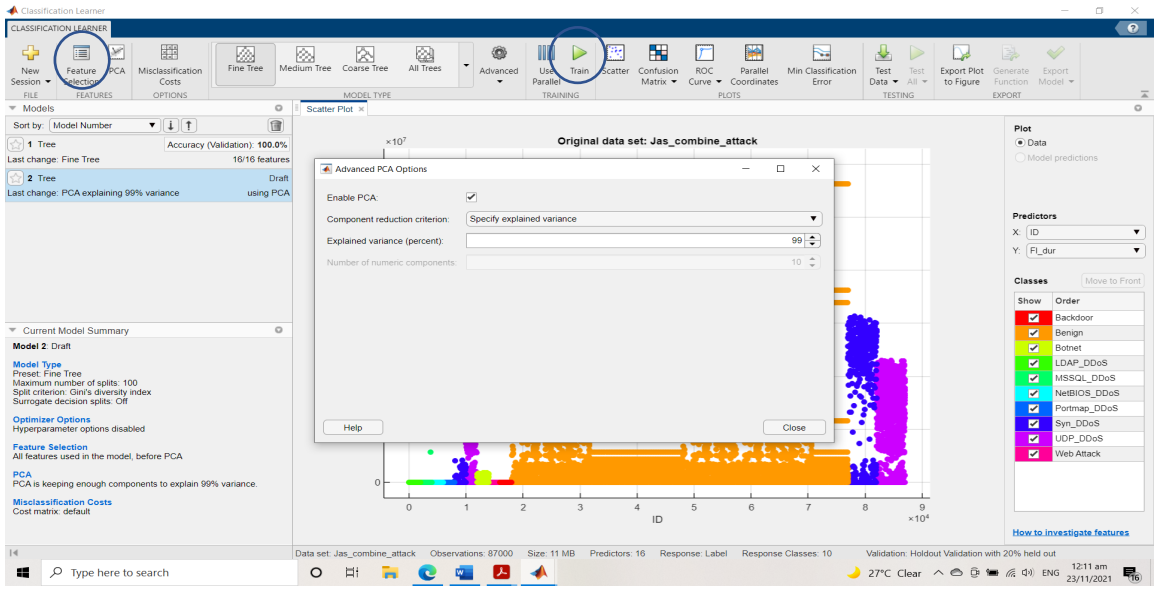

Figure 38. Advanced PCA setting

Step 9. Click “Confusion Matrix” then select “Validation Data” and followed by “TPR and FNR.”

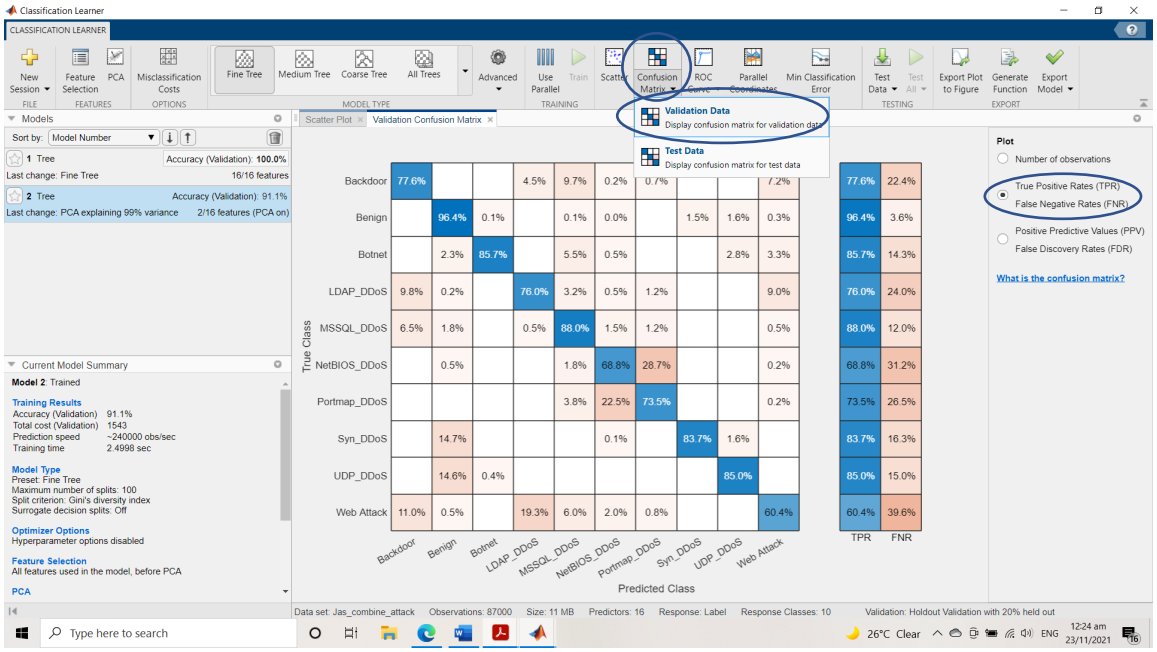

Figure 39. Confusion matrix

Step 10. Click “Export Plot to figure” then edit the model accordingly to the setting.

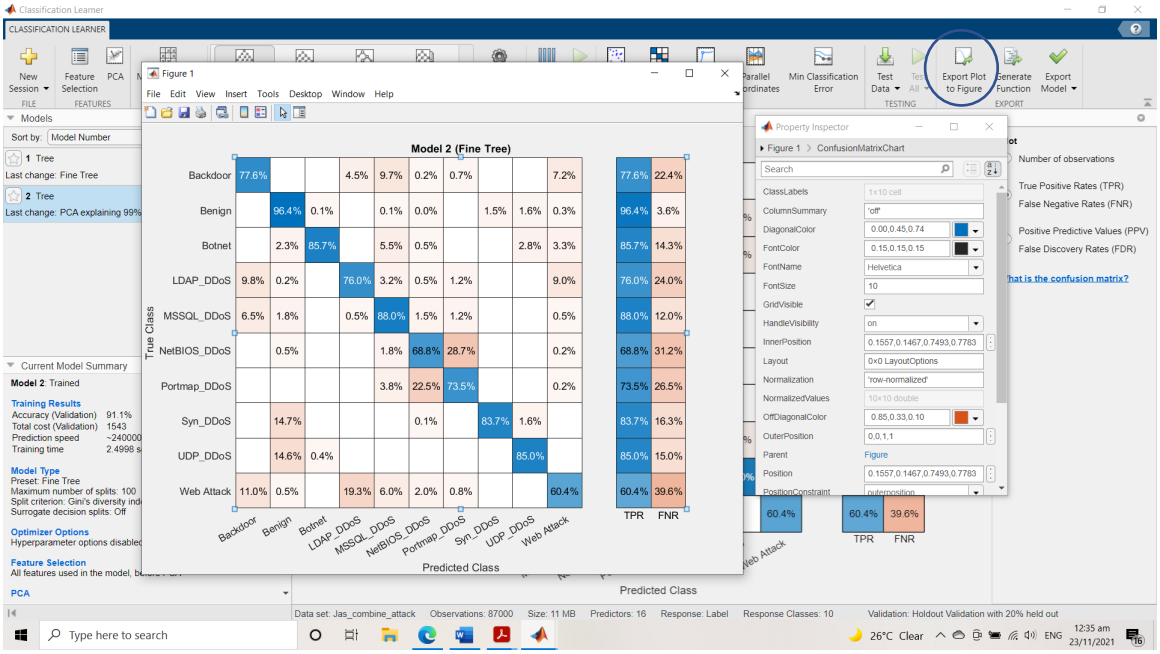

Figure 40. Export plot and customize labelling

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] B. E. Alessandro Guidotti1, Marco Di Renzo. "Integrated satellite-terrestrial networks in future wireless systems." Wiley, November 5, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/sat.1292

[2] H. Z. kun li , Zhe tu , Weilin wang, and A. B. Zhang, "Distributed Network Intrusion Detection System in Satellite-Terrestrial Integrated Networks Using Federated Learning," IEEE Access Novemeber 26 2020. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9274426

[3] Y. K. D. Cheng, ""Performance analysis of an unmanned systems communications network using data distribution serivce in a lossy environment" " Master Naval Postgraduate School, Monterey, CA; 2021.

[4] E scholarly community Encyclopedia, "Unmanned System," 17 Mar 2021 [Online]. Available: https://encyclopedia.pub/8882

[5] J. W. Yawen Tan, Jiajia Liu, and Yanning Zhang, "Unmanned Systems Security: Models, Challenges, and Future Directions," IEEE Network July/August 2020 [Online]. Available: https://ieeexplore.ieee.org/document/9023464

[6] P. W. Webb, *Wireless Communications:The Future*. , The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England: John Wiley & Sons Ltd, 2007 [Online]. Available: https://download.e-bookshelf.de/download/0000/5684/68/L-G-0000568468-0015276220.pdf

[7] M. B. Bharat B Madan, and Doina Bein, "Securing unmanned autonomous systems from cyber threats," Vol. 16, no. 2, pp 119–136 Feb 2019. [Online]. Available: https://journals.sagepub.com/doi/abs/10.1177/1548512916628335?journalCode=dmsa

[8] R. J. S. Brandon bailey, Prashant A. Doshi,nicholas C. Cohen,wayne A. Wheeler "Defending spacecraft in the cyber domain," Nov 2019 [Online]. Availble: https://csps.aerospace.org/sites/default/files/2021-08/Bailey_DefendingSpacecraft_11052019.pdf

[9] "National Air and Space Intelligence Center; Competing in Space," p. page 18, December 2018. [Online]. Available: https://media.defense.gov/2019/Jan/16/2002080386/-1/-1/1/190115-F-NV711-0002.PDF.

[10] J. Z. a. C. Wang, "Satellite Networking Intrusion DetectionSystem Design Based on Deep Learning Method," June 2018. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-10-6571-2_280.

[11] "Cyber Security Attack Types – Active and Passive attacks." A. Bhattacharya. Mar 2021 [Online]. Available: https://www.encryptionconsulting.com/active-and-passive-attacks/

[12] Elprocus. “Basic Intrusion Detection System.” 2013–2022 [Online] Avaliable: https://www.elprocus.com/basic-intrusion-detection-system/

[13] Barracuda. “What is a Intrusion Detection System?” [Online]. Available: https://www.barracuda.com/glossary/intrusion-detection-system#section_1

[14] IBM Cloud Education “Unsupervised Learning..” September 21, 2020 [Online]. Available: https://www.ibm.com/cloud/learn/unsupervised-learning

[15] Analytics India Magazine pvt ltd, “Top 6 Regression Algorithms Used In Data Mining And Their Applications In Industry.” September 19, 2017 [Online]. Available: https://analyticsindiamag.com/top-6-regression-algorithms-used-data-mining-applications-industry/

[16] Mathworks. “Machine Learning.” Accessed Jul. 14, 2020. [Online]. Available: https://www.mathworks.com/discovery/machine-learning.html

[17] S. C. L. Shilpashree. S, Nayana G Bhat, Sunil Kumar G, “Decision Tree: A Machine Learning for Intrusion Detection,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* April 2019 [Online]. Avaliable: https://www.ijitee.org/wp-content/uploads/papers/v8i6s4/F12340486S419.pdf

[18] M.-S. A. Fares Fourati, Fellow. ,”Artificial Intelligence for Satellite Communication: A Review,” *IEEE,* Jan 25, 2021 [Online]. Available: https://arxiv.org/pdf/2101.10899.pdf

[19] N. S. Chauhan. “Decision Tree Algorithm.” KDnuggets. February 9, 2022 [Online]. Avaliable: https://www.kdnuggets.com/2020/01/decision-tree-algorithm explained.html

[20] A. Navlani. “Decision Tree Classification in Python.” datacamp. December 29, 2018 [Online]. Avaliable: https://www.datacamp.com/community/tutorials/decision-tree-classification-python

[21] D. S. Sayad. “Decision Tree - Overfitting.” 2010 - 2022 [Online]. Avaliable: https://www.saedsayad.com/decision_tree_overfitting.htm

[22] T. Plapinger, “What is a Descision Tree? ,” *towards data science* Jul 30, 2017. [Online]. Available: https://towardsdatascience.com/what-is-a-decision-tree-22975f00f3e1.

[23] A. Sharma. “4 Simple Ways to Split a Decision Tree in Machine Learning.” Analytics Vidhya. June 30, 2020 [Online]. Avaliable: https://www.analyticsvidhya.com/blog/2020/06/4-ways-split-decision-tree

[24] N. Tyagi. “Understanding the Gini’s Index and Information Gain in Decision Trees.” Medium, Mar 24, 2020 [Online]. Avaliable: https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees ab4720518ba8

[25] L. Breiman, J. Friedman, R. Olshen, and C. Stone., “*Classification and Regression Trees.”* FL: CRC Press, 1984. pp. 11–21

[26] Y. Sitta, “Time efficiency and accuracy improvement using pca,” in *algoritma technical blog*, ed, April 13, 2020. [Online]. Avaliable: https://algotech.netlify.app/blog/time-and-accuracy-improvement-using-pca/

[27] H. Goonewardana, “PCA: Application in Machine Learning,” *Medium,* Feb 28, 2019. [Online]. Available: https://medium.com/apprentice-journal/pca-application-in-machine-learning-4827c07a61db.

[28] Mathworks. “Pca.” Accessed Jul. 14, 2020. [Online]. Available: https://www.mathworks.com/help/stats/pca.html

[29] Python Learn Python Programming. “Advantages of a Decision Tree for Classification.” 2021 [Online]. Available: https://pythonprogramminglanguage.com/what-are-the-advantages-of-using-a-decision-tree-for-classification/

[30] A. O. Rakesh Shrestha, Sajjad Ahmadi Roudi, Robert Abbas and Shiho Kim, “Machine-Learning-Enabled Intrusion Detection System for Cellular Connected UAV Networks,” May 28, 2021. [Online]. Available: https://www.researchgate.net/publication/352667669_Machine_Learning_Enabled_Intrusion_Detection_System_for_Cellular_Connected_UAV_Networks

[31] R. I. a. F. Dovis, “Distinguishing Ionospheric Scintillation from Multipath in GNSS Signals Using Bagged Decision Trees Algorithm,” presented at the IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE) Vicenza, Italy November 23, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9262699

[32] M. S. B Madhevan, “Tracking Algorithm Using Leader Follower Approach for Multi Robots,” Indian Institute of Information Technology Design and Manufacturing (IIITD&M) Kancheepuram, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705813017384

[33] J. L. L. a. T. M. Khoshgoftaar, “A survey and analysis of intrusion detectionmodels based on CSE-CIC-IDS2018 Big Data,” Journal of big data. November 23, 2020. [Online].Available:https://www.researchgate.net/publication/346536711_A_survey_and_analysis_of_intrusion_detection_models_based_on_CSE-CIC-IDS2018_Big_Data

[34] N. M. a. J. Slay, “UNSW-NB15: A comprehensive data set fornetwork intrusion detection systems (UNSW-NB15 network data set),” *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), Canberra, ACT, Australia,,* pp. 1–6, Nov. 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7348942

[35] Canadian Institute for Cybersecurity “NSLKDD dataset” UNB, 2009. [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html

[36] The Tcpdump Group, “Tcpdump” June 9, 2021. [Online]. Available: https://www.tcpdump.org

[37] Qosient, LLC. “Argus.” [Online]. Available: https://www.openargus.org/

[38] Canadian Institute for Cybersecurity “CICFlowMeter” UNB, 2009. [Online]. Available: https://www.unb.ca/cic/research/applications.html

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California