

USE OF HARDWARE FINGERPRINTING FOR INTRUSION DETECTION IN AVIONICS  
SYSTEMS

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF INFORMATICS OF  
THE MIDDLE EAST TECHNICAL UNIVERSITY  
BY

İSA CAN BABİR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE  
IN  
THE DEPARTMENT OF CYBER SECURITY

JUNE 2025



**USE OF HARDWARE FINGERPRINTING FOR INTRUSION DETECTION IN AVIONICS  
SYSTEMS**

submitted by **İSA CAN BABİR** in partial fulfillment of the requirements for the degree of **Master of Science in Cyber Security Department, Middle East Technical University** by,

Prof. Dr. Banu Günel Kılıç  
Dean, **Graduate School of Informatics**

\_\_\_\_\_

Assoc. Prof. Dr. Cihangir Tezcan  
Head of Department, **Cyber Security**

\_\_\_\_\_

Prof. Dr. Nazife Baykal  
Supervisor, **Information Systems, METU**

\_\_\_\_\_

**Examining Committee Members:**

Assoc. Prof. Dr. Cihangir Tezcan  
Cyber Security, METU

\_\_\_\_\_

Prof. Dr. Nazife Baykal  
Information Systems, METU

\_\_\_\_\_

Assist. Prof. Dr. Sinem Sav  
Computer Engineering, Bilkent University

\_\_\_\_\_

**Date: 02.06.2025**



**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

**Name, Surname: İSA CAN BABİR**

**Signature :**

# ABSTRACT

## USE OF HARDWARE FINGERPRINTING FOR INTRUSION DETECTION IN AVIONICS SYSTEMS

BABİR, İSA CAN

M.S., Department of Cyber Security

Supervisor: Prof. Dr. Nazife Baykal

June 2025, 46 pages

With the emergence of next-generation avionic platforms, systems that were previously isolated from external networks are now exposed to attacks. Traditional aviation communication buses and avionics systems, such as the MIL-STD-1553 standard, were developed without considering security requirements, as external attacks were deemed improbable due to their closed architecture. MIL-STD-1553, widely implemented across commercial, military, and aerospace avionic platforms, is a communication bus where security concerns were largely overlooked. However, this notion of invulnerability has since been disproven. As aircraft systems become more interconnected with external networks, they are increasingly vulnerable to attacks that threaten operational effectiveness. Implementing security measures, which require hardware and software modifications, results in costly and complex certification challenges, particularly for legacy systems. Intrusion Detection Systems (IDS) have gained popularity as a solution, as they do not require hardware or software modifications. This study aims to enhance the security of MIL-STD-1553 communication by integrating a Hardware Fingerprinting-Based IDS. The research evaluates the effectiveness of machine and deep learning techniques in detecting unauthorized devices on the bus. Supervised learning methods achieved perfect classification accuracy, excelling in precision, recall, and F1-score, while unsupervised methods showed limited success in anomaly detection. Additionally, a feature reduction process was applied to improve performance, revealing that supervised methods maintained high accuracy with fewer features, while unsupervised methods saw performance decline. Lastly, the study investigates the stability of synchronization signals over time, finding they remain consistent, supporting the reliability of the device's unique characteristics.

Keywords: MIL-STD-1553, intrusion detection, hardware fingerprinting, machine and deep learning

# ÖZ

## AVİYONİK SİSTEMLERDE SALDIRI TESPİTİ İÇİN DONANIM PARMAK İZİ KULLANIMI

BABİR, İSA CAN

Yüksek Lisans, Siber Güvenlik Bölümü

Tez Yöneticisi: Prof. Dr. Nazife Baykal

Haziran 2025, 46 sayfa

Yeni nesil havacılık platformlarının ortaya çıkmasıyla birlikte, daha önce dış ağlardan izole edilmiş olan sistemler artık saldırılara maruz kalmaktadır. Geleneksel havacılık iletişim hatları ve aviyonik sistemler, MIL-STD-1553 standardı gibi, güvenlik gereksinimleri göz önünde bulundurulmadan geliştirilmiştir, çünkü dış saldırılar kapalı mimarileri nedeniyle olasılık dışı olarak değerlendirilmiştir. MIL-STD-1553, ticari, askeri ve uzay havacılık platformlarında yaygın bir şekilde uygulanmış olup, güvenlik endişelerinin büyük ölçüde göz ardı edildiği bir iletişim hattıdır. Ancak, bu dokunulmazlık düşüncesinin yanlış olduğu zamanla kanıtlanmıştır. Uçak sistemleri, dış ağlarla giderek daha fazla bağlantı kurdukça, operasyonel etkinliklerini tehdit eden saldırılara karşı daha savunmasız hale gelmektedir. Donanım ve yazılım değişiklikleri gerektiren güvenlik önlemleri eklemek, özellikle eski sistemler için maliyetli ve karmaşık sertifikasyon sorunlarına yol açmaktadır. Bu değişiklikleri gerektirmeyen Saldırı Tespit Sistemleri, popüler bir çözüm olarak öne çıkmaktadır. Bu çalışma, MIL-STD-1553 iletişiminin güvenliğini, Donanım Parmak İzi Tabanlı bir Saldırı Tespit Sistemi entegrasyonu ile artırmayı amaçlamaktadır. Araştırma, makine ve derin öğrenme tekniklerinin, veri yolundaki yetkisiz cihazları tespit etme etkinliğini değerlendirmektedir. Denetimli öğrenme yöntemleri, mükemmel sınıflandırma doğruluğu elde etmiş ve hassasiyet, duyarlılık ve F1-puanı gibi metriklerde üstün başarı göstermiştir; buna karşın, denetimsiz yöntemler, anomali tespitinde sınırlı başarı göstermiştir. Ayrıca, performansı artırmak için öznitelik azaltma süreci uygulanmış, denetimli yöntemler daha az öznitelik ile yüksek doğruluk sağlarken, denetimsiz yöntemlerin performansında düşüş yaşandığı görülmüştür. Son olarak, çalışma, senkronizasyon sinyallerinin zaman içindeki kararlılığını araştırarak, bu sinyallerin tutarlı kaldığını ve cihazın benzersiz özelliklerinin güvenilirliğini desteklediğini bulmuştur.

Anahtar Kelimeler: MIL-STD-1553, saldırı tespiti, donanım parmak izi, makine öğrenmesi ve derin öğrenme

I dedicate this work to a future where the concept of security becomes obsolete through trust and integrity.

## **ACKNOWLEDGMENTS**

I would like to express my deepest gratitude to my supervisor, Prof. Dr. Nazife Baykal, for her invaluable guidance and encouragement throughout the course of this study.

I would also like to extend my sincere thanks to my team leader and my teammates for their collaboration, advice, and continuous support during this journey. Their contributions have been instrumental in overcoming various challenges along the way.

Lastly, I am profoundly grateful to my family and my girlfriend for their unwavering patience and understanding during this demanding period. Their belief in me has been a constant source of motivation and strength.

# TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ.....	v
DEDICATION.....	vi
ACKNOWLEDGMENTS.....	vii
TABLE OF CONTENTS.....	viii
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xii
LIST OF ABBREVIATIONS.....	xiii
CHAPTERS	
1 INTRODUCTION.....	1
1.1 Problem Statement.....	2
1.2 Research Questions.....	3
1.3 Proposed Approach.....	3
1.4 Contributions of the Study.....	4
1.5 Organization of the Thesis.....	4
2 BACKGROUND.....	5
2.1 MIL-STD-1553.....	5

2.2	Security Aspects of MIL-STD-1553 .....	9
2.3	Intrusion Detection Systems .....	11
2.3.1	Types of IDS .....	12
2.3.2	Detection Methods of IDS .....	13
2.4	Intrusion Detection Strategies for MIL-STD-1553 .....	14
2.4.1	Physical Intrusion Detection .....	14
2.4.2	Hardware Fingerprinting-Based Intrusion Detection .....	15
2.4.3	Context-Based Intrusion Detection .....	15
2.4.4	Integrating Detection Mechanisms for Comprehensive Security .....	16
2.5	Machine Learning and Deep Learning .....	17
2.5.1	Supervised Approaches .....	17
2.5.1.1	Random Forest .....	17
2.5.1.2	K-Nearest Neighbors .....	18
2.5.1.3	Support Vector Machines .....	18
2.5.1.4	Long Short-Term Memory .....	18
2.5.2	Unsupervised Approaches .....	18
2.5.2.1	Isolation Forest .....	18
2.5.2.2	One-Class Support Vector Machines .....	19
2.5.2.3	Local Outlier Factor .....	19
3	METHODOLOGY .....	21
3.1	Data Acquisition Process .....	21
3.2	Threat Model .....	24
3.2.1	Attack Scenario 1 .....	24

3.2.2	Attack Scenario 2 .....	25
3.3	Experiments and Datasets .....	26
3.3.1	Experiment 1 .....	26
3.3.2	Experiment 2 .....	27
3.3.3	Experiment 3 .....	28
3.4	Preprocessing, Model and Training .....	28
3.4.1	Preprocessing .....	28
3.4.2	Model and Training .....	29
3.5	Evaluation .....	29
4	EXPERIMENTAL RESULTS .....	31
4.1	Results of Experiment 1 .....	31
4.2	Results of Experiment 2 .....	36
4.3	Results of Experiment 3 .....	40
4.4	Further Analysis of Signal Distributions .....	41
5	CONCLUSION .....	43
	REFERENCES .....	45

## LIST OF TABLES

Table 1	Example Attack Scenarios on MIL-STD-1553 .....	10
Table 2	Comparison of NIDS and HIDS .....	13
Table 3	Comparison of SIDS and AIDS .....	14
Table 4	Comparison of Machine Learning and Deep Learning Algorithms .....	20
Table 5	CSV Format of Collected Voltage Signals .....	24
Table 6	Parameters of Learning Algorithms .....	29
Table 7	Results of Experiment 1 Attack Scenario 1 .....	32
Table 8	Results of Experiment 1 Attack Scenario 1 .....	32
Table 9	Results of Experiment 1 Attack Scenario 2 .....	33
Table 10	Results of Experiment 1 Attack Scenario 2 .....	33
Table 11	Results of Experiment 2 Attack Scenario 1 .....	37
Table 12	Results of Experiment 2 Attack Scenario 2 .....	38
Table 13	Results of Experiment 3 .....	41

## LIST OF FIGURES

Figure 1	MIL-STD-1553 Bus Architecture . . . . .	6
Figure 2	MIL-STD-1553 Word Formats . . . . .	6
Figure 3	MIL-STD-1553 Encoding . . . . .	7
Figure 4	BC to RT, RT to BC, and RT to RT Communication Types . . . . .	8
Figure 5	Minor and Major Frame Structure . . . . .	8
Figure 6	Tools for Network Traffic Monitoring and Threat Detection . . . . .	12
Figure 7	Data Acquisition Setup . . . . .	22
Figure 8	Obtained Voltage-Time Values . . . . .	22
Figure 9	Synchronization Signal Extraction Process . . . . .	23
Figure 10	Unique Synchronization Signals of Same Model Transmitters . . . . .	23
Figure 11	Unique Synchronization Signals of Different Transmitters . . . . .	23
Figure 12	Evil BC Injecting Messages During Idle Slots . . . . .	25
Figure 13	Replaced RT Transmitting Malicious Messages . . . . .	26
Figure 14	Feature Importance Derived from Random Forest Model . . . . .	35
Figure 15	Heatmap Visualization of the Absolute Values of SVM Coefficients . . . . .	35
Figure 16	Training and Validation Loss Curves of LSTM Model . . . . .	36
Figure 17	F1-Score of Unsupervised Methods vs Voltage Sample Count, Attack Scenario 1 .	37
Figure 18	F1-Score of Unsupervised Methods vs Voltage Sample Count, Attack Scenario 2 .	39
Figure 19	Spearman Correlation Matrix of Voltage Sample Features . . . . .	40
Figure 20	Mean vs Standard Deviation for Sync Signals Across Devices . . . . .	42
Figure 21	Skewness vs Kurtosis for Sync Signals Across Devices . . . . .	42

## LIST OF ABBREVIATIONS

AC	Accuracy
AI	Artificial Intelligence
AFDX	Avionics Full-Duplex Switched Ethernet
AIDS	Anomaly-based Intrusion Detection System
BC	Bus Controller
BM	Bus Monitor
CIA	Confidentiality, Integrity, and Availability
CSV	Comma Separated Values
DC	Direct Current
DL	Deep Learning
DoS	Denial of Service
EGI	Embedded GPS/INS
EMI	Electromagnetic Interference
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GPS/INS	Global Positioning System/Inertial Navigation System
HF	High Frequency
HIDS	Host-based Intrusion Detection System
ICS/OT	Industrial Control Systems/Operational Technology
IDS	Intrusion Detection System
IF	Isolation Forest

IFF	Identification Friend or Foe
KNN	K-Nearest Neighbors
LOF	Local Outlier Factor
LSTM	Long Short-Term Memory
Mbit/s	Megabits per Second
MHz	Megahertz
MITM	Man in the Middle
ML	Machine Learning
NIDS	Network-based Intrusion Detection System
OCSVM	One-Class SVM
PR	Precision
RC	Recall
RF	Random Forest
RNN	Recurrent Neural Network
RT	Remote Terminal
SIDS	Signature-based Intrusion Detection System
SIL	System Integration Laboratory
SVM	Support Vector Machines
TN	True Negative
TP	True Positive
T/R	Transmit/Receive
TSN	Time-Sensitive Networking
VPN	Virtual Private Network
V/UHF	Very/Ultra High Frequency

# CHAPTER 1

## INTRODUCTION

Avionic platforms have undergone rapid evolution, with significant technological advancements being integrated into next-generation systems. Modern platforms are now characterized by continuous connectivity, enabling real-time exchange of data with the outside world. These platforms receive live weather updates, flight information, and other critical data, enhancing their operational capabilities. Moreover, innovations that require high levels of connectivity, such as artificial intelligence, have begun to play a pivotal role, with technologies like virtual pilot assistants and autonomous flight systems becoming increasingly common. However, with this increased connectivity, next-generation platforms are now more exposed to the outside world, expanding their attack surfaces and introducing new vulnerabilities [1], [2]. In 2023, the aviation industry faced a 24% increase in cyberattacks, underscoring the growing importance of robust cybersecurity measures [3].

Traditional avionic platforms were developed with safety requirements in mind, as security was not a major concern at the time, given their isolation from external networks. However, with the advent of new technologies, modern platforms have become highly connected, integrating advanced systems and adopting a 'security by design' approach. One of the areas most impacted by these advancements is the avionics bus structure, which serves as the communication backbone between systems. New-generation communication protocols, such as Avionics Full-Duplex Switched Ethernet (AFDX) and Time-Sensitive Networking (TSN), now include security measures to protect against various types of attacks [4], [5]. In contrast, legacy protocols lack these protections and are increasingly vulnerable, necessitating the addition of modern security measures to mitigate these risks.

Introducing security features into these legacy protocols requires hardware and software modifications, followed by a re-certification process. This cycle is both costly and time-intensive, and with a significant number of platforms still relying on these protocols, implementing such changes in a timely manner is impractical. As an alternative that does not require modifications, intrusion detection systems (IDSs) have emerged as an effective solution.

MIL-STD-1553 is one of the most widely used legacy communication protocols, having been in use for over 40 years [6]. It features a dual-redundant bus with a master-slave architecture and was originally designed with a primary focus on safety rather than security. To address modern security challenges, IDSs have been proposed as a solution. Various IDS approaches have been suggested to enhance the security of MIL-STD-1553, including context-based IDS, hardware fingerprinting techniques, and physical IDS [7], [8], [9], [10], [11].

Research on MIL-STD-1553 security has primarily concentrated on analyzing the content of messages transmitted through the data bus, often employing a context-based approach. In parallel, other systems focus on the signal characteristics of the data bus, utilizing hardware-based detection techniques.

Context-based IDS monitors communication, specifically message traffic on the bus, and model normal communication patterns using machine and deep learning techniques. Messages transmitted over the bus are analyzed, and any deviation from the model is flagged as anomalous. While context-based IDSs are effective at identifying adversaries that display abnormal behavior, they cannot detect adversaries that mimic normal behavior, since MIL-STD-1553 lacks an authentication mechanism. To address this issue, hardware fingerprinting techniques have been proposed as a solution, effectively functioning as an authentication mechanism.

Hardware fingerprinting techniques analyze the signal characteristics of MIL-STD-1553 transmitters and classify devices on the data bus for authentication purposes. Each transmitter, even those of the same model, has distinct signal characteristics due to minor defects during production, which can be used as a fingerprint. This unique characteristic serves as the hardware fingerprint for authentication.

However, hardware fingerprinting techniques are effective only if the illegitimate device is actively transmitting on the bus. Silent rogue devices that merely monitor the bus cannot be detected. To address this limitation, side-channel techniques offer a solution by detecting any changes in the bus's voltage profile when a new connection is made. A model is created by analyzing the behavior of bus voltage using machine and deep learning methods, and during operation, voltage levels are continuously monitored. Any deviation from the expected voltage profile is flagged as an anomaly. When used together, context-based IDS, hardware fingerprinting, and side-channel techniques provide a comprehensive end-to-end security solution.

## **1.1 Problem Statement**

New-generation avionic platforms, which are heavily connected to the outside world, have expanded the potential attack surfaces, making them more vulnerable to threats. Many of these platforms still rely on legacy communication protocols that prioritize safety over security, leaving them exposed to various attacks. MIL-STD-1553, one of these legacy communication protocols, lacks an authentication mechanism, which prevents the detection of unauthorized devices connected to the bus.

Although hardware fingerprinting methods have been proposed in a limited number of studies, these are typically based on single machine learning (ML) or deep learning (DL) approaches, and there is a lack of comprehensive analysis in the literature. Notably, many studies do not provide a detailed comparison of the ML/DL techniques used, nor do they address feature optimization or provide exhaustive result tables.

Another significant challenge in the avionics domain is minimizing false positive rates, which are crucial for maintaining flight safety. High false positive rates can distract pilots and undermine operational reliability, so it is essential to keep these rates as low as possible. This thesis addresses the gap in the existing literature by conducting a comprehensive analysis of the ML/DL methods applicable to hardware fingerprinting and evaluating their performance with respect to false positive rates.

## 1.2 Research Questions

Based on the identified problem, this thesis aims to answer the following research questions:

1. What are the performance outcomes of machine and deep learning methods applied to hardware fingerprinting for the MIL-STD-1553 communication bus?
2. What are the key features that can be used in ML/DL models for hardware fingerprinting, and how can feature optimization improve model performance?
3. Which methods offer an optimal solution for the avionics domain by minimizing false positive rates without compromising security?
4. How does the performance of the model change over time, given that signal characteristics may vary during runtime?

## 1.3 Proposed Approach

The effectiveness and accuracy of an IDS for MIL-STD-1553 depend heavily on the quality and diversity of the data used during its development. This study considers both normal and abnormal bus conditions to create datasets that accurately reflect real-world scenarios. Given that MIL-STD-1553 is predominantly used in military avionics and aerospace systems, acquiring real-world data presents significant challenges, as access to such platforms is often restricted. Therefore, many studies rely on data generated through simulations or testbeds.

Data is collected from the System Integration Laboratory (SIL) of a real platform for this research. The SIL replicates the real-world configuration of the platform and simulates various operational scenarios. This setting ensures the data reflects a wide range of conditions, making it suitable for developing a reliable IDS.

The proposed method focuses on detecting unauthorized devices that attempt to inject illegitimate messages into the bus, without disrupting ongoing communications. The goal is to identify these malicious devices based on their unique hardware fingerprints, which are derived from the voltage signal characteristics of the MIL-STD-1553 bus. To achieve this, a combination of machine learning and deep learning methods is employed to differentiate between legitimate and unauthorized devices effectively.

The data collection process involves capturing voltage signals from the MIL-STD-1553 bus during different operational phases. A digital oscilloscope, coupled with a bus coupler and differential probe, is used to record voltage-time series at a high sampling rate of 100 mega samples per second. This allows for the extraction of synchronization signals, which are critical for identifying unique hardware fingerprints. The extracted signals are then structured into a dataset suitable for machine and deep learning analysis.

The experiments in this study are designed to evaluate the performance of the models under different attack scenarios. The first experiment tests the ability of supervised and unsupervised learning methods to detect malicious devices in both attack scenarios: one where an evil device injects messages and

another where a compromised device mimics a legitimate device. The second experiment explores the optimization of feature sets, specifically the minimum number of voltage samples required to maintain classification accuracy. The third experiment analyzes how fluctuations in signal characteristics over time affect model performance. By collecting data at multiple time intervals, this experiment assesses the models' robustness to environmental factors that may influence signal stability.

#### 1.4 Contributions of the Study

Following contributions are made for this thesis:

1. *Collecting Real Data from MIL-STD-1553*: Since MIL-STD-1553 is primarily used in military platforms, open datasets are not readily available. For this study, we collected MIL-STD-1553 data from an avionic platform operating under real-world conditions to form the datasets.
2. *Development of Attack Injection Tool for Anomaly Generation*: A novel attack injection tool is developed for injecting attacks on MIL-STD-1553. This tool generates realistic attack scenarios and provides an opportunity to assess the true performance of the proposed methods.
3. *Extensive Analysis of Machine and Deep Learning Methods for Hardware Fingerprinting in Anomaly Detection*: Studies on hardware fingerprinting for anomaly detection are scarce and typically propose only one method without detailed performance results. In our study, we provide an in-depth analysis of machine and deep learning methods, presenting detailed performance results that highlight the best approaches for hardware fingerprinting.
4. *Feature Optimization for Hardware Fingerprinting*: This is the first study focused on feature optimization for hardware fingerprinting. Extensive performance analyses are presented in this thesis, demonstrating the optimal feature selection process for improving accuracy.

#### 1.5 Organization of the Thesis

This thesis is organized as follows:

- The Introduction chapter presents the problem, research questions, the proposed approach, and the contributions of the study.
- The Background chapter provides information about the MIL-STD-1553 protocol, its security aspects, intrusion detection systems and strategies related to MIL-STD-1553, as well as machine learning and deep learning techniques.
- The Methodology chapter covers the data acquisition process, the threat model and attack scenarios, the experiments and datasets, as well as the preprocessing steps, model selection, and training process.
- The Results chapter discusses the outcomes of the proposed methods.
- The Conclusion chapter summarizes the findings and outlines directions for future work.

## CHAPTER 2

### BACKGROUND

#### 2.1 MIL-STD-1553

As integrated avionics systems grew more complicated, the number of discrete connections between terminal devices increased. Direct point-to-point wiring subsequently became impractical, introducing added complexity and increased costs. In response to this need, the United States Department of Defense developed MIL-STD-1553 in 1973 as a standardized serial data bus communication protocol [6]. Operating at a speed of 1 Mbit/s as a half-duplex, bi-directional data bus, it uses a command/response structure for transmissions. The protocol features a dual-redundant bus topology and bi-phase Manchester II encoding. Although initially designed for military avionics, MIL-STD-1553 has gained widespread use across both military and commercial aviation, as well as in broader aerospace applications. Its high reliability and low fault rate make it ideal for safety-critical applications globally, including communication networks, navigation systems, weapon systems, and satellite systems.

MIL-STD-1553 utilizes three distinct types of devices to interface with the data bus. These devices include the bus controller (BC), remote terminal (RT), and bus monitor (BM) which are shown in Figure 1.

1. *Bus Controller:* It is responsible for coordinating data transmission on the bus, controlling the entire data flow by sending commands to remote terminals and receiving status replies that contain conditional information about RTs. As the sole controller of the data bus, BC initiates and manages all communication, playing a crucial role in the data bus system. To ensure redundancy and reliability, multiple bus controllers may be connected to the bus as backups, with additional controllers activated in the event of a failure. However, only one bus controller can be active at any given time.
2. *Remote Terminal:* It is a device that connects a system to the bus, providing an interface with other systems. It operates under the control scheme established by the bus controller in accordance with the protocol. Remote terminals respond to commands sent by the bus controller with status information and transmit or receive data corresponding to those commands. A maximum of 31 remote terminals can be connected to the bus.
3. *Bus Monitor:* It is a passive device that monitors bus communication and collects traffic data. It can filter specific activities for real-time analysis or store them for later examination. The bus monitor is primarily used for testing purposes.

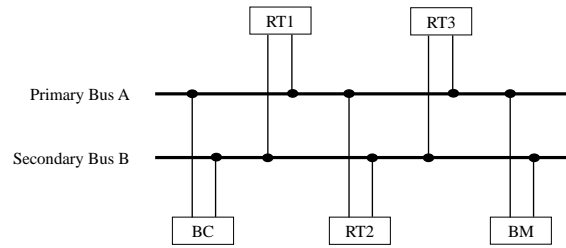


Figure 1: MIL-STD-1553 Bus Architecture

Communication in MIL-STD-1553 is carried out using three different types of words: command words, status words, and data words.

- ❖ *Command words* are transmitted by the bus controller to the remote terminals, instructing them to perform a certain task, which can either be to receive or transmit data. Command words include a terminal address field that specifies the targeted RT, a transmit/receive (T/R) bit to indicate the expected action from the RT, and data transfer details such as the message's subaddress and the count of data words to be delivered.
- ❖ *Status words* are transmitted by remote terminals in response to the command words sent by BC. Status words include a terminal address field to specify which RT is replying, as well as multiple flag bits that convey essential information about the operational state of the RT.
- ❖ *Data words* are transmitted by either an RT or the BC. They hold the actual information exchanged between systems connected to the bus.

In MIL-STD-1553, each transmitted word consists of a 20-bit sequence. The first 3 bits form a synchronization waveform, the next 16 bits specify the content of the word, and the final bit is used for parity to check for bit errors during transmission. Figure 2 shows MIL-STD-1553 word formats.

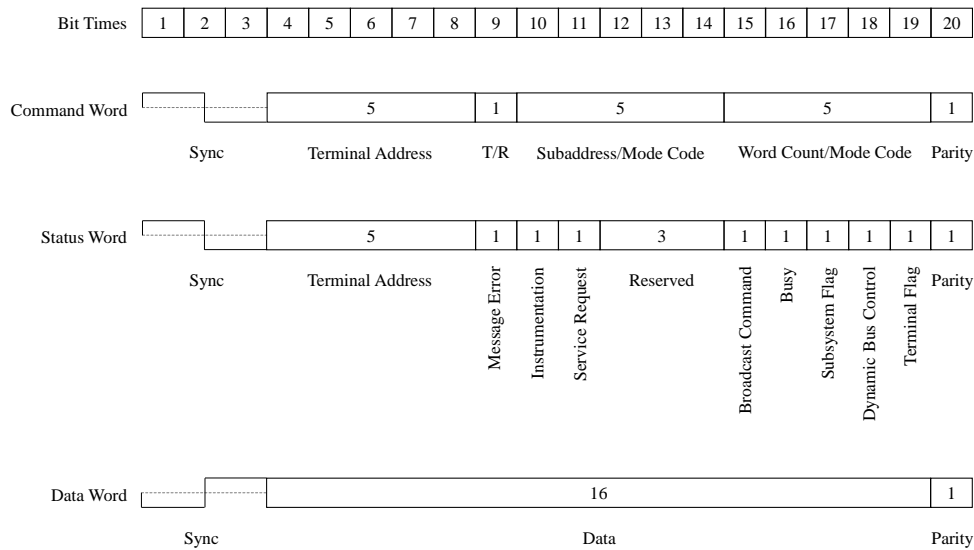


Figure 2: MIL-STD-1553 Word Formats

MIL-STD-1553 employs bi-phase Manchester II encoding to maintain synchronization and data integrity on the data bus. In this approach, each bit is represented by a voltage transition within a designated time interval, generating a self-clocking signal. A transition from high-to-low voltage at the midpoint of the interval represents a binary 1, while a low-to-high transition represents a binary 0. The sync waveform, which spans 3 bit times, is positive for the first one-and-a-half bit times and negative for the remaining one-and-a-half bit times in command and status words. For data words, the sync waveform also spans 3 bit times but differs slightly: it is negative for the first one-and-a-half bit times and positive for the remaining one-and-a-half bit times. Figure 3 exhibits sync waveform patterns and encoding scheme used in MIL-STD-1553.

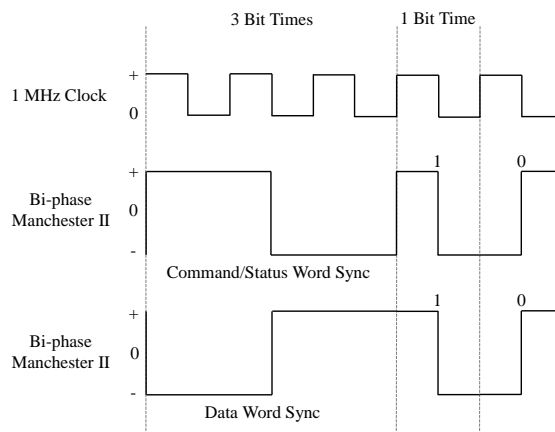


Figure 3: MIL-STD-1553 Encoding

The protocol supports several types of communication between devices, including mode commands, broadcast commands, and the most common transfers: BC to RT, RT to BC, and RT to RT. The bus controller transmits mode command messages to a remote terminal to regulate its operations, check its status, or conduct overall data bus management. RTs recognize commands from the BC as mode commands by checking whether the subaddress/mode code field is '00000' (0) or '11111' (31). If this condition is met, the word count/mode code field defines the specific type of mode command issued by the BC. The bus controller can also send broadcast mode commands to multiple remote terminals simultaneously by setting the terminal address field to '11111' (31).

The communication types BC to RT, RT to BC, and RT to RT are illustrated in Figure 4. Communication from the bus controller to a remote terminal begins with a receive command sent by the BC to the RT. After this, the BC starts transmitting data words. Once the RT has completed receiving the data words, it sends a status word to inform the BC about the status of the data transfer. Communication from a remote terminal to the bus controller begins with a transmit command sent by the BC to the RT. Following this, the RT sends a status word to inform the BC that it is ready to begin transmission and then starts transmitting data words. Communication between two remote terminals is initiated by the bus controller, which sends a receive command to one RT and a transmit command to the other. The transmitting RT first sends a status word to indicate its condition, then begins transmitting data words. After receiving all the data words, the receiving RT sends a status word to report the status of the data transfer.

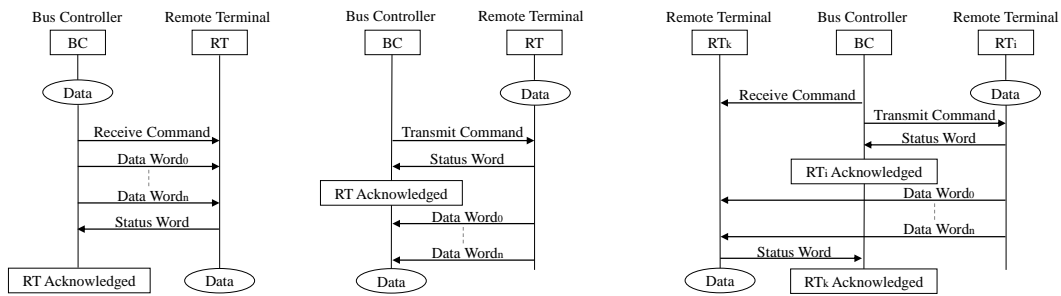


Figure 4: BC to RT, RT to BC, and RT to RT Communication Types

In MIL-STD-1553, the bus controller transmits command words according to a periodic schedule, organizing messages into predefined structures known as major frames. These major frames are composed of multiple minor frames, with each minor frame containing MIL-STD-1553 messages. A sequence of messages within each minor frame is timed precisely to maintain synchronized data flow across the bus, enabling deterministic bus traffic design. Depending on the implementation, minor frames may contain empty slots. Aperiodic messages, which are on-demand messages triggered by events such as operator actions or system interrupts, can be placed in these empty slots or at the end of a minor frame if time permits. Figure 5 shows minor and major frame structure.

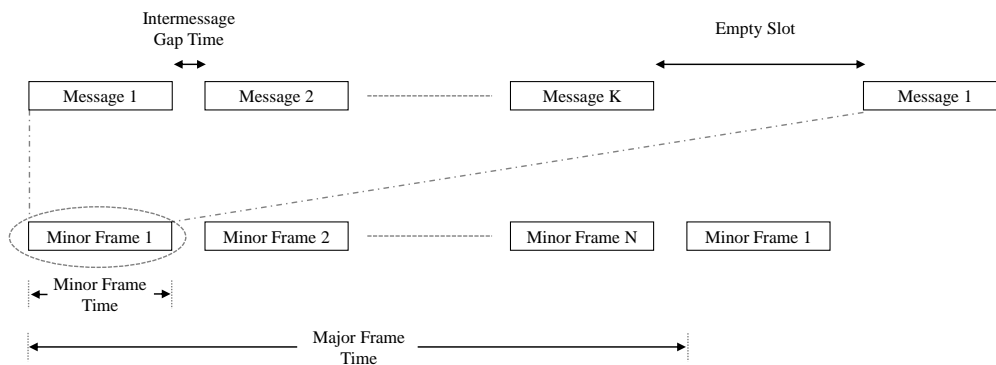


Figure 5: Minor and Major Frame Structure

The 1553 data bus is a twisted-shielded pair transmission line that uses couplers to connect terminal devices to the bus via stubs. Twisted-shielded pair cables mitigate electromagnetic interference (EMI) through both physical shielding and the noise-canceling effect created by the twisting of the wires. MIL-STD-1553's physical layer uses differential signaling and a defined "idle" state to prevent collisions. It features transformer isolation for direct current (DC) isolation, common-mode rejection, and lightning protection, along with series resistors to prevent short circuits. High transmit voltages enable efficient data transmission, while specifications for rise and fall times minimize EMI. The standard also establishes receiver voltage thresholds and sets requirements for noise rejection and input impedance. These features collectively enhance the reliability, safety, and performance of MIL-STD-1553, making it suitable for demanding applications in military and aerospace environments.

## 2.2 Security Aspects of MIL-STD-1553

MIL-STD-1553 was developed long before the introduction of the triad of Confidentiality, Integrity, and Availability (CIA), which serves as a foundational model for guiding policies on information security. At the time of its development, avionics systems were isolated from external networks. This allowed designers to focus primarily on safety and reliability, emphasizing fault tolerance and dual redundancy. Cyber attacks, as we are familiar with today, had not yet surfaced, so security was not a primary design consideration. However, with the evolution of next-generation platforms that are now interconnected with external networks through mediums such as cloud services, Wi-Fi, datalink, and satellite links, new vulnerabilities have emerged, making security a critical concern [12], [13], [14].

The protocol lacks crucial security features such as source authentication and data encryption, which leaves it vulnerable to exploitation through various attack vectors. These attack vectors often require in-depth knowledge specific to the platform, including details about message formats, and time scheduling. While some attacks require physical access to the system, others can be conducted remotely. For example, an attacker could perform an over-the-air update of a device using malicious software to compromise it from a distance. Compromises can also occur earlier in the supply chain, as systems are made up of components sourced from diverse manufacturers. Additionally, firmware or software provided by subcontractors for devices can also be compromised, introducing vulnerabilities that may be exploited later in the development process.

The broad usage of MIL-STD-1553 in military and aerospace systems has led to various studies on its security vulnerabilities and potential attack vectors. Stan et al. (2017) highlight these vulnerabilities in their security analysis of MIL-STD-1553, identifying critical assets, attack vectors, and potential attacker profiles [15]. Physical components, including the transmission medium, transceivers, and couplers, are key assets susceptible to compromise, along with data assets like transmitted messages and subsystem information. These assets are important for the functionality and confidentiality of MIL-STD-1553 systems, and any breach could lead to operational disruption or exposure of sensitive information.

Stan et al. classify potential threats into three primary types: denial of service (DoS) attacks, data leakage, and data integrity violations. *DoS attacks* can occur through both physical damage and logical manipulation, such as modifying command words to create collisions. *Data leakage* may involve unauthorized transmission between components of different security levels, often by exploiting reserved fields in status words or idle bus times. *Data integrity threats* involve the modification of data, which can lead to incorrect operations. Examples include altering navigation information or spoofing system messages [15].

Another study on MIL-STD-1553 security by Lounis et al. (2022) dives into specific attack vectors that exploit the protocol's lack of inherent security features, categorizing vulnerabilities based on Stallings' security classification: fabrication, interception, interruption, and modification attacks [16].

*Fabrication attacks* exploit the absence of authentication on the bus, allowing attackers to impersonate legitimate devices. For instance, techniques such as man-in-the-middle (MITM) attacks and "BC Evil-Twin" scenarios enable unauthorized interception and command injection. In severe cases, an attacker may even introduce a rogue bus controller, disrupting communication by causing collisions with legitimate commands. *Interception attacks* exploit the plaintext nature of MIL-STD-1553 data, allowing compromised components to eavesdrop on or log sensitive information directly from the bus. Attack-

ers may use techniques such as covert channels, embedding hidden data in unused bits within status words, or redirecting data between remote terminals to extract critical information without detection. *Interruption attacks* can disrupt bus operations by targeting timing gaps or physically tampering with components. Techniques like random word generation can inject noise into the bus, leading to communication interference or complete denial of service. Such interruptions are especially concerning when they affect mission-critical functions. *Modification attacks* pose a threat to data integrity on the bus by altering transmitted data or interfering with command sequences. Attackers may corrupt or manipulate messages through techniques like command invalidation, data corruption, and word-count tampering, leading to malfunctioning systems or inaccurate operations [16].

Table 1 provides a small selection of attack vectors on MIL-STD-1553, illustrating how vulnerabilities related to authentication, confidentiality, integrity, and availability can impact system security. In the context of information security:

- ❖ *Authentication* means that the receiver should be able to identify the true identity of the transmitter.
- ❖ *Confidentiality* means that only authorized users can access the information during transmission.
- ❖ *Integrity* means that only authorized users can modify or delete information.
- ❖ *Availability* means that authorized users should be able to access and use information at any time necessary.

While this table provides only a glimpse of the broader range of potential threats, the identified attack vectors highlight the need for robust security measures. Solutions such as intrusion detection systems are essential to protect MIL-STD-1553 communications from the expanding range of risks introduced by interconnected environments.

Table 1: Example Attack Scenarios on MIL-STD-1553

<i>Security Feature</i>	<i>Attack Vector</i>	<i>Attack Description</i>	<i>Potential System Impact</i>
Authentication	Fake Command Injection	Injecting malicious commands during bus idle times	An attacker injecting malicious commands during idle periods might cause unauthorized activation of flight control surfaces. This could lead to unsafe maneuvers or loss of control.
Confidentiality	Eavesdropping on the Bus	Intercepting and recording the bus traffic	An attacker intercepting and recording messages exchanged on the bus might extract sensitive information. This could lead to the leakage of classified mission data.

Table 1 (cont.)

Integrity	Data Word Cor- ruption	Injecting fake data words into the bus	An attacker injecting fake data words into the bus might change GPS data being sent to onboard systems. This could mislead the pilot, causing a misinterpretation of the current location.
Availability	Random Word Generation	Inserting random words into the bus to create col- lisions	An attacker inserting random words into the bus might cause messages to fail to reach their intended destinations due to collisions. This could result in blocked communication and subsequent disruptions to the system.

### 2.3 Intrusion Detection Systems

Intrusion is defined as any unauthorized access, manipulation, or interference with system resources, often leading to security breaches and compromised data integrity. Intrusion detection system help counter intrusions by discovering, determining, and identifying unauthorized use, duplication, alteration, or destruction of information [17]. An IDS functions by continuously monitoring the network in which it operates and searching for signs of intrusions. Detecting intrusions is achieved by whether analyzing known attack signatures and patterns or identifying deviations from normal behaviors. Through early detection, the objective of IDSs is to mitigate potential threats. In response to detected intrusions, IDSs may alert administrators, enabling them to take appropriate action.

The importance of IDS is underscored by recent findings in the SANS ICS/OT Cybersecurity Survey 2023, which identifies IDS as one of the top security priorities for industrial control systems (ICS) [18]. The survey emphasizes the need for anomaly and intrusion detection tools to safeguard ICS networks, especially as cyber threats continue to target critical infrastructure with increasing sophistication. As a proactive security measure, IDS offers essential capabilities for detecting unauthorized access and unusual behaviors in real time. This provides an indispensable layer of protection in environments where system integrity is crucial.

The importance of IDS is further reinforced by findings in the 2024 State of Network Threat Detection Report by Cybersecurity Insiders, which highlights the diverse tools security professionals rely on to address visibility gaps and enhance network threat detection capabilities [19]. Among these, network-based intrusion detection systems are the most commonly utilized, with 67% of respondents reporting their deployment.

Figure 6 shows types of tools used for network traffic monitoring and threat detection, as highlighted in the report.

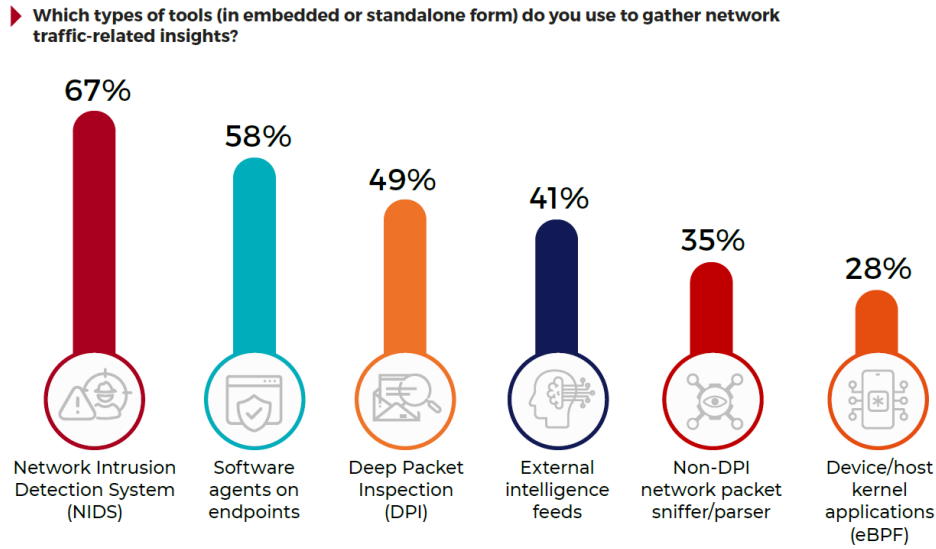


Figure 6: Tools for Network Traffic Monitoring and Threat Detection

While IDS plays a critical, proactive role in detecting intrusions, it differs fundamentally from other network protection mechanisms, such as firewalls. Firewalls are designed to block unwanted inbound and outbound traffic by enforcing a set of predefined rules. They need to be carefully configured and kept up to date to avoid hindering normal network traffic while still blocking malicious activity. The key difference lies in the proactive nature of IDSs in identifying intrusions. Even if the attack is novel, IDSs can detect the suspicious activity. In contrast, firewalls are generally reactive and they focus on blocking known threats.

### 2.3.1 Types of IDS

Intrusion detection systems can be categorized into two main groups based on the scope of their deployment and the perimeter they monitor: Network-based IDS (NIDS) and Host-based IDS (HIDS).

- **Network-based IDS** screens the entire network traffic or a particular segment of it to detect abnormal activity. NIDS can detect unauthorized access attempts, malicious data flows, and abnormal activity by analyzing network packets in transit. It is typically deployed at network boundaries, such as near border firewalls, routers, virtual private network (VPN) servers, remote access servers, and wireless networks.
- **Host-based IDS** monitors an individual device and tracks internal events to detect unusual activity. By observing changes to critical system files, logins, and system calls, HIDS can identify unauthorized modifications, local attacks, and insider threats. It is most often deployed on critical hosts, such as publicly accessible servers and those storing sensitive information [20].

Table 2 lists some of the advantages and disadvantages of NIDS and HIDS [21].

Table 2: Comparison of NIDS and HIDS

<i>IDS Type</i>	<i>Advantages</i>	<i>Disadvantages</i>
NIDS	+ Broad network coverage: Monitors multiple hosts simultaneously by analyzing network packets without requiring installation on each individual device. This allows it to detect a wide range of attacks across various network protocols.	- Limited visibility: Monitors data transmitted through the network, so it may not catch malicious activities that happen on individual hosts, such as insider threats. - High volume of data: Must examine large amounts of network traffic, which can lead to performance degradation during high traffic loads.
HIDS	+ Detailed host analysis: Inspects the full range of host activities without requiring additional hardware. This allows it to detect intrusions even within encrypted communications, by reassembling and analyzing complete data packets.	- Limited visibility: Monitors activity on the host it's installed on, making it difficult to detect network-wide or coordinated attacks. - Host-specific: Requires installation on all hosts to cover the entire attack surface, leading to increased resource usage and deployment complexity.

### 2.3.2 Detection Methods of IDS

Intrusion detection systems can be categorized into two main groups based on the methods they use to identify potential threats: Signature-based IDS (SIDS) and Anomaly-based IDS (AIDS).

- **Signature-based IDS** identifies intrusions by comparing observed incidents to a database of known patterns or signatures of previous attacks. Although simple, SIDS is an effective method for detecting known threats and is widely used in security tools such as Snort [22].
- **Anomaly-based IDS** detects intrusions by forming a baseline of normal behavior over a time period and then flagging any activity which deviates from that behavior as potentially malicious. AIDS operates in two phases: a training phase, where normal activity is profiled, and a testing phase, where current activity is compared against the established profile of normal behavior. Statistical methods or machine and deep learning techniques can be used to model system behavior.

Table 3 lists some of the advantages and disadvantages of SIDS and AIDS [20], [21].

Table 3: Comparison of SIDS and AIDS

<i>Detection Method</i>	<i>Advantages</i>	<i>Disadvantages</i>
Signature-based	+ High precision: Highly effective in detecting known attacks with minimal false alarms, making them reliable due to their straightforward design.	- Limited to known threats: Unable to detect attacks with unknown signatures, including zero-day attacks. - High maintenance costs: Needs frequent updates to the signature database to stay effective, which can demand significant time and resources.
Anomaly-based	+ Capable against unknown threats: Highly effective in detecting previously unseen attacks by identifying deviations from normal behavior, making it suitable for zero-day attack detection.	- Prone to false alarms: May mistake normal behavior for abnormal, resulting in a high rate of false positives. - Challenging profile creation: Difficult to form a reliable baseline for highly dynamic and complex systems, making alert validation harder.

## 2.4 Intrusion Detection Strategies for MIL-STD-1553

MIL-STD-1553 remains widely used across a range of critical platforms, from military aviation to space systems, due to its reliability and well-established protocol. However, updating hardware and software configurations to integrate modern security measures often proves both costly and limited by strict certification requirements. These limitations present a challenge for integrating security enhancements, such as encryption, directly into MIL-STD-1553 networks. Modifying the protocol or equipment could require re-certification and compromise system stability.

As a solution, intrusion detection systems are increasingly proposed as an effective means of security in MIL-STD-1553. IDS offer the ability to monitor and analyze bus traffic as well as device hardware characteristics without requiring substantial changes to existing hardware or software protocols. By continuously observing communication patterns and flagging anomalous behaviors, IDS can detect signs of unauthorized access, malicious commands, or unusual device responses. This approach enables a proactive security layer that can adapt to MIL-STD-1553's constraints while providing real-time alerts.

Intrusion detection within the protocol focuses on three primary mechanisms: physical intrusion detection, hardware fingerprinting-based intrusion detection, and context-based intrusion detection.

### 2.4.1 Physical Intrusion Detection

Physical intrusion detection primarily identifies unauthorized devices connected to the bus. As shown previously in Figure 3, data transmitted on the bus consists of a series of positive and negative voltages.

When a new device is introduced, it changes the overall impedance characteristics of the bus, resulting in a distinct voltage pattern. This mechanism is particularly significant as it can detect silent devices on the bus, which may be used for extracting confidential information.

Building on this concept, and as part of a comprehensive protection system, Levy et al. propose a physical intrusion detection mechanism specifically designed for the MIL-STD-1553 bus [7]. The system uses voltage signal analysis and targets vulnerabilities introduced by unauthorized devices, including passive ones. By detecting changes in voltage patterns caused by additional devices, the system effectively identifies intrusions. To implement this approach, the authors use an autoencoder-based machine learning model trained to recognize the normal voltage signal patterns associated with legitimate devices. Observed signals are analyzed for reconstruction errors, with deviations beyond a defined threshold signaling the presence of an unauthorized device.

#### **2.4.2 Hardware Fingerprinting-Based Intrusion Detection**

Hardware fingerprinting-based intrusion detection is employed to authenticate the source of transmissions on the data bus. By analyzing voltage signal characteristics, particularly the synchronization signals fixed at the start of each word, this method ensures that only legitimate devices are transmitting on the bus. Each transmitter, even those of the same model, exhibits unique signal characteristics due to subtle differences from production processes. These unique traits serve as hardware fingerprints, enabling reliable device authentication.

Building on the concept of hardware fingerprinting-based intrusion detection, the RT authentication module is proposed by Stan et.al [23]. The RT authentication module utilizes voltage signal analysis to detect unauthorized connections and transmissions. Each RT generates a distinct signal profile due to minor variations in its hardware. These profiles act as fingerprints, enabling the module to classify and authenticate signals as legitimate or illegitimate. The authentication process consists of two steps. In the first step, the module classifies each captured signal as legitimate or illegitimate using machine learning-based clustering techniques. If a signal's fingerprint does not align with known legitimate patterns, it is flagged as illegitimate. In the second step, for signals classified as legitimate, the module verifies whether the signal's source RT address matches the expected address. Any mismatch results in the signal being marked as spoofed. Signal features, such as statistical metrics, time-frequency transformations, and regressions are extracted to enhance the fingerprinting process. These features are calculated from the synchronization bits at the beginning of each word, as these bits are consistent across words and allow for accurate signal characterization.

#### **2.4.3 Context-Based Intrusion Detection**

Context-based intrusion detection is designed to identify anomalous messages transmitted on the bus. Message transmission on the bus is managed by the bus controller through a predefined command/response pattern. This pattern follows a strict set of rules, where each message is placed within minor frames, which are prearranged structures. Given this cyclic architecture, context-based IDS achieves its functionality by analyzing the timing, sequence, and content of messages to detect out-of-order or unexpected transmissions.

The deterministic nature of the protocol makes context-based IDS an effective solution for improving security. The delivery of messages in periodic frames creates a predictable environment, allowing the system to anticipate subsequent messages after a given transmission. This predictability paves the way for the development of accurate real-time intrusion detection.

Building on the capabilities of context-based intrusion detection, Elsayed et al. introduce AdaptIDS, an advanced adaptive intrusion detection system designed for MIL-STD-1553 communication [8]. It uses data science principles, with an emphasis on sequence-to-sequence learning through Long Short-Term Memory (LSTM) neural networks. Their framework elevates context-based intrusion detection for MIL-STD-1553 by using a three-component architecture: Data Encoder, Ensemble Learner, and Intrusion Detector. It processes bus traffic into fixed-length time windows, using LSTM-based classifiers and a meta-learner to identify temporal and spatial anomalies in bus sequences. Operating in preparation and real-time production phases, AdaptIDS continuously adapts to new traffic patterns.

#### **2.4.4 Integrating Detection Mechanisms for Comprehensive Security**

Context-based IDS, physical IDS, and hardware fingerprinting-based IDS form a trio of complementary mechanisms designed to enhance the security of a bus communication protocol.

Physical IDSs are essential for detecting unauthorized silent devices added to the bus topology to monitor communications. By analyzing slight fluctuations in bus voltage characteristics, physical IDSs can identify passive devices that neither transmit nor actively interact with the bus. In contrast, hardware fingerprinting-based IDSs and context-based IDSs are ineffective against such passive devices, as they rely on the transmission of messages to function.

On the other hand, hardware fingerprinting-based IDSs are critical for authenticating transmitters by using their unique signal characteristics. This method is effective in detecting replaced transmitters, where physical IDSs might fail, as replacing a transmitter is less likely to change the overall voltage characteristics of the bus.

Context-based IDSs, however, play an important role in identifying compromised transmitters engaged in unauthorized data transmissions. A compromised device may retain the same hardware fingerprint as a legitimate device and avoid altering the bus's voltage characteristics, evading detection by both physical IDSs and hardware fingerprinting-based IDSs. In such scenarios, context-based IDSs play a vital role in detecting unauthorized data activities on the bus.

Each intrusion detection mechanism —physical IDS, hardware fingerprinting-based IDS, and context-based IDS— addresses distinct vulnerabilities in the MIL-STD-1553 communication protocol. Physical IDSs detect passive, unauthorized devices; hardware fingerprinting-based IDSs authenticate transmitters and identify replacements; and context-based IDSs uncover unauthorized data transmissions by compromised devices. Individually, these methods provide critical layers of defense, addressing specific attack vectors. However, no single mechanism is sufficient to protect the bus system comprehensively. Combining these measures ensures robust, versatile security, capable of mitigating a wide range of potential threats. By leveraging the strengths of each detection mechanism, this integrated approach raises the overall resilience of the bus system.

## **2.5 Machine Learning and Deep Learning**

In the present day, technological advancements are steering towards a field where computers not only perform specific tasks but also have the capability to learn and adapt. In the realm of artificial intelligence (AI), machine learning emerges as a foundational tool for systems to navigate and respond to changing environments. Unlike explicit programming, ML relies on data-driven approaches to refine algorithms and uncover patterns. Machine learning focuses on developing algorithms and statistical models that enable computers to execute tasks without explicit instructions, making it a powerful instrument for deriving insights and identifying patterns [24].

Building upon the principles of ML, deep learning extends these capabilities by leveraging artificial neural networks to model complex, hierarchical representations of data. DL excels in tasks requiring the analysis of large datasets with intricate relationships, such as image recognition, natural language processing, and time-series analysis. In the context of anomaly detection, an essential application of ML and DL, algorithms aim to identify instances that deviate significantly from expected behavior. By searching through data, these algorithms uncover outliers, which may signify critical events, potential threats, or system errors.

Algorithms can be broadly categorized based on the learning approach they adopt. Supervised learning, a fundamental aspect, involves training algorithms to map inputs to outputs using labeled examples provided by a supervisor. This approach is particularly effective when clear labels are available.

Conversely, unsupervised learning operates without labeled data, focusing instead on uncovering hidden patterns, relationships, or anomalies within the data. This method is especially valuable for tasks like anomaly detection, where the aim is to identify underlying structures or deviations from expected behavior.

To address the challenges of anomaly detection in complex systems, such as the MIL-STD-1553 communication bus, various machine and deep learning algorithms are employed. Each algorithm offers unique strengths in processing and analyzing the distinct characteristics of data captured from this critical legacy protocol.

### **2.5.1 Supervised Approaches**

#### **2.5.1.1 Random Forest**

Random Forest (RF) constructs multiple decision trees during the training phase. For classification tasks, it determines the final output based on the majority class chosen by its constituent trees. In the context of anomaly detection, Random Forest leverages the collective decision boundaries of these trees to effectively identify irregularities within the data.

Random Forest is particularly well-suited for anomaly detection due to its capability to handle large datasets and its adaptability to high-dimensional feature spaces. Its ensemble approach ensures robust performance even when dealing with complex data distributions.

### **2.5.1.2 K-Nearest Neighbors**

K-Nearest Neighbors (KNN) classifies data points by examining the majority class among their closest neighbors based on feature similarity. Anomalies are effectively identified by their distance from normal instances, as they typically do not align with established clusters.

KNN is particularly well-suited for scenarios involving moderate-dimensional data and balanced datasets. Its straightforward nature and adaptability make it a valuable tool for detecting irregularities.

### **2.5.1.3 Support Vector Machines**

Support Vector Machines (SVM) classify data by constructing a hyperplane that separates input data in a multidimensional feature space. Each dimension represents a feature from the input dataset, and the hyperplane serves as the decision boundary that maximizes the margin between classes.

SVM is particularly effective in scenarios with clear class separations, as the maximized margin enhances its ability to differentiate between normal and anomalous instances. Its versatility and robustness make SVM a valuable tool for detecting anomalies.

### **2.5.1.4 Long Short-Term Memory**

Long Short-Term Memory (LSTM) networks are a specialized form of Recurrent Neural Network (RNN) designed to capture long-term dependencies in sequential data. They are particularly well-suited for anomaly detection tasks involving temporal datasets, as they can uncover patterns and relationships within time series data to identify deviations from typical behavior.

LSTM incorporates memory cells that store information from previous inputs, allowing the network to establish connections between past and current data. This capability enables LSTM to effectively predict future inputs by leveraging stored memories, making it an invaluable tool for detecting anomalies in time-series datasets.

## **2.5.2 Unsupervised Approaches**

### **2.5.2.1 Isolation Forest**

Isolation Forest (IF) isolates anomalies by constructing a collection of isolation trees. Each isolation tree is built through iterative branching, where the data is split based on randomly selected features until individual data points are isolated. Anomalies are identified by their shorter average path lengths from the root node to the leaf node in the isolation trees, as they are more easily separated from the rest of the data.

IF is particularly effective for handling large datasets due to its linear time complexity, making it well-suited for industrial applications with high demands for efficiency.

### **2.5.2.2 One-Class Support Vector Machines**

One-Class Support Vector Machines (OCSVM) is designed to isolate a single class, typically representing normal data. OCSVM constructs a boundary, often shaped as a sphere, around the majority of data points in a high-dimensional feature space. The algorithm aims to find the smallest possible sphere that encapsulates most of the normal instances while tolerating a few outliers.

During prediction, a new data point is projected into the learned feature space, and its distance from the sphere's center is calculated. If this distance exceeds a predefined threshold, indicating significant deviation from the normal pattern, the point is classified as an anomaly. OCSVM is particularly well-suited for scenarios where the dataset is dominated by normal instances, and the goal is to identify rare or abnormal events.

### **2.5.2.3 Local Outlier Factor**

Local Outlier Factor (LOF) identifies anomalies by analyzing the local density of data points in relation to their neighbors. For each point in the dataset, LOF computes its local density by measuring the distance to its nearest neighbors. Points with shorter distances have higher local density, indicating they are surrounded by closely packed neighbors.

LOF assigns a score to each data point by comparing its local density to that of its neighbors. A high LOF score indicates that the point resides in a region with significantly lower density compared to its surroundings, suggesting it is an outlier. This method is particularly effective for detecting anomalies in datasets with varying densities.

Table 4 provides a comprehensive comparative summary of the supervised and unsupervised learning algorithms utilized in this study. It highlights their key features, such as strengths in handling specific types of data and tasks, and their computational complexity.

Table 4: Comparison of Machine Learning and Deep Learning Algorithms

<i>Algorithm</i>	<i>Approach</i>	<i>Key Features</i>	<i>Computational Complexity</i>
RF	Supervised	Handles large datasets; robust against noise	Moderate
KNN	Supervised	Simple; effective for moderate-dimensional data	Low
SVM	Supervised	Maximizes margin for clear class separation	Moderate to High
LSTM	Supervised	Captures long-term dependencies in sequential data	High
IF	Unsupervised	Isolates anomalies based on separation from data bulk	Low
OCSVM	Unsupervised	Defines boundary for normal data; effective for rare anomalies	Moderate
LOF	Unsupervised	Detects outliers in datasets with varying densities	Moderate

## CHAPTER 3

### METHODOLOGY

#### 3.1 Data Acquisition Process

Building an effective, reliable, and accurate intrusion detection system depends significantly on the quality and diversity of data used during the development phase. For MIL-STD-1553 networks, this requires careful attention to the bus's operational characteristics and potential attack scenarios. The data acquisition process involves capturing bus traffic under both normal and abnormal conditions to ensure the resulting dataset accurately reflects real-world behaviors.

Acquiring real-world MIL-STD-1553 data is particularly challenging, as the protocol is predominantly utilized in military avionics and aerospace systems. Researchers often face significant barriers when trying to access these platforms for data collection. Consequently, most studies on MIL-STD-1553 rely on data generated from simulation environments or dedicated testbeds.

In this study, data is collected from the System Integration Laboratory of a helicopter. The SIL plays a critical role in the development process, serving as a testing and verification environment for both hardware and software before integration with the actual platform. It replicates the configuration and devices of the real platform and can simulate its full range of operational capabilities. This makes the SIL ideal for emulating distinct operational phases, such as take-off, cruising, and landing, as well as mission-specific scenarios like engaging an enemy or evading an incoming missile. For this study, utilizing the SIL enabled a reliable data acquisition process and helped avoid inaccuracies that could arise from less representative simulation environments.

The SIL includes two mission computers, one of which is active and functions as the bus controller, while the other is passive and serves as a backup. It also contains several remote terminals, such as high-frequency (HF) and very/ultra-high frequency (V/UHF) radios, embedded GPS/INS (EGI), and identification friend or foe (IFF) devices.

With the SIL providing a realistic and controlled environment, the next step in this study involves capturing voltage signals directly from the MIL-STD-1553 bus in various operational scenarios. This is accomplished using a digital oscilloscope with bus decoding capabilities, which records the electrical characteristics of bus communication, specifically the voltage signals. Figure 7 illustrates the data acquisition setup, highlighting the use of a bus coupler, a differential probe and a digital oscilloscope to capture and decode voltage signals from the MIL-STD-1553 bus in the SIL environment. A bus coupler is utilized to connect the Keysight MSOS204A digital oscilloscope to the bus via the Keysight N2818A differential probe.

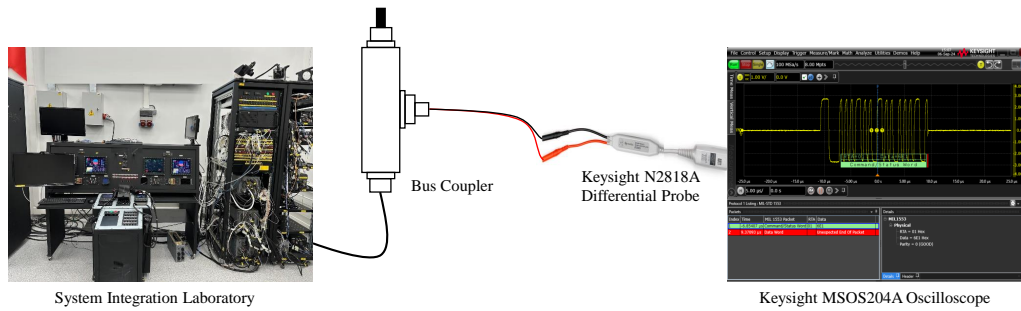


Figure 7: Data Acquisition Setup

Sampling the signals at a rate of 100 mega samples per second produces a digital representation with 100 data points per microsecond of bus communication, which operates at a frequency of 1 megahertz. The signals acquired from the oscilloscope are essentially voltage-time series, where voltage values are recorded at discrete time intervals. Figure 8 shows a sample voltage-time series captured from the oscilloscope, depicting the recorded voltage values over discrete time intervals.

Time	Voltage
0.02172134774	0.0131803
0.02172135774	-0.0115987
0.02172136774	-0.0170026
0.02172137774	-0.0131803
0.02172138774	0.0155527
0.02172139774	0.0089626
0.02172140774	0.0080398
0.02172141774	-0.0081718
0.02172142774	0.0289966
0.02172143774	0.0631336

Figure 8: Obtained Voltage-Time Values

To facilitate hardware fingerprinting, synchronization signals are extracted from the recorded voltage-time series. Specifically, the command sync signals transmitted by the bus controller and the status sync signals sent by the remote terminals are extracted, as these are critical for identifying unique hardware characteristics. Data sync signals are excluded, as they are not relevant to the fingerprinting process.

The Stumpy library is employed for extracting synchronization signals from the recorded data [25]. Stumpy is a Python library designed for time series data mining and analysis, with a focus on algorithms for pattern discovery, motif identification, and similarity search. In this study, a sample synchronization signal is used as a reference pattern, and matching sync signals are extracted from the recorded messages. Each sync signal spans 3 bit times, and with a sampling rate of 100 mega samples per second, a total of 300 voltage-time values are obtained for each sync signal. The extraction process of sync signals is shown in Figure 9.

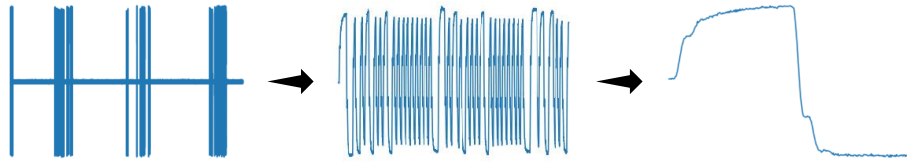


Figure 9: Synchronization Signal Extraction Process

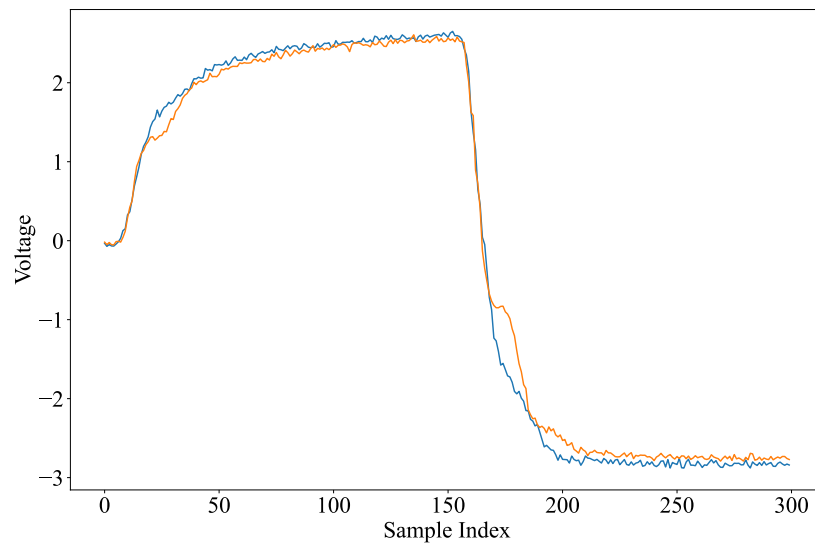


Figure 10: Unique Synchronization Signals of Same Model Transmitters

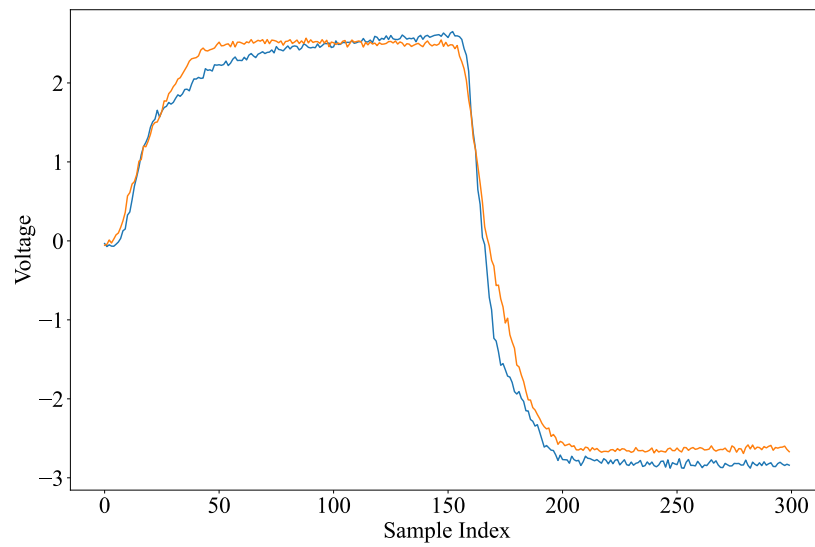


Figure 11: Unique Synchronization Signals of Different Transmitters

Once the relevant patterns are identified, each signal containing these patterns is extracted and saved to a Comma Separated Values (CSV) file. In this CSV file, each voltage sample from the signal corresponds to a separate column, while each row represents a distinct signal. Each voltage sample within the signal serves as a feature for the learning algorithms. This structured representation allows for easy analysis of the data, facilitating the application of learning algorithms for classification. Table 5 illustrates the format of the CSV file, showcasing how voltage samples are organized as features and signals as rows.

Table 5: CSV Format of Collected Voltage Signals

voltage1	voltage2	...	voltage300	classLabel
-0.0338049	-0.0720301	...	-2.8336065	cduPilot
-0.0185501	-0.0504390	...	-2.7585094	cduCopilot

The synchronization signals extracted from the bus provide a unique fingerprint for each transmitter, even among devices of the same model. Figure 10 illustrates the voltage patterns of sync signals generated by two transmitters of the same model. While the overall structure is consistent due to shared hardware and software configurations, subtle variations, arising from manufacturing tolerances and hardware defects, make each transmitter’s signal unique.

In contrast, Figure 11 demonstrates the sync signals of two transmitters from different models. These signals exhibit more pronounced differences, reflecting the varying hardware designs and configurations. These distinctions highlight the reliability of synchronization signals as hardware fingerprints, enabling precise identification of transmitters on the bus, even in challenging scenarios.

### 3.2 Threat Model

Since this study focuses on an IDS based on hardware fingerprinting, the attack scenarios and threat model are carefully aligned with its capabilities. The primary objective is to detect unauthorized transmissions on the MIL-STD-1553 bus. While other threats, such as eavesdropping, are relevant to MIL-STD-1553 networks, this study focuses on unauthorized transmissions, as they pose a direct and immediate threat to the integrity of the system. These unauthorized sources could originate from newly attached devices attempting to infiltrate the bus or from replaced devices that have been compromised. These two scenarios form the core of the threat model and will be explored in detail in this section. This threat model assumes that attackers have physical access to the bus, allowing them to add or replace devices. It also presumes that compromised devices are pre-configured to execute malicious operations without relying on external command inputs.

#### 3.2.1 Attack Scenario 1

An unauthorized "evil" bus controller is connected to the bus to send illegitimate messages during bus idle times. These messages are crafted to mislead targeted remote terminals into processing them as valid commands. Additionally, random message injection can disrupt bus communication by inducing collisions, causing delays and compromising system reliability. To avoid collisions and maintain the

appearance of normal bus operation, the attack leverages idle slots for message injection, making sure that ongoing communication is not directly disrupted.

Using idle slots is critical for this type of attack, as it avoids triggering immediate errors due to bus traffic interference. The attack is executed using an algorithm that detects intermessage gap times, identifying periods long enough to accommodate a message of the maximum allowable size. For example, a message simulating RT to RT communication with a maximum data word count of 32 requires approximately 720 microseconds. By targeting these idle slots, it is ensured that the messages can be injected stealthily while minimizing the risk of detection through collision-induced disruptions.

This attack scenario exemplifies a fake command injection targeting the authentication of MIL-STD-1553 communication, as detailed in Table 1. By injecting illegitimate commands, the attacker exploits idle slots to bypass normal bus activity, potentially leading to unauthorized access to sensitive subsystems.

Figure 12 illustrates an unauthorized "evil" bus controller connected to the MIL-STD-1553 bus, injecting illegitimate messages during idle times to avoid collisions and mislead remote terminals.

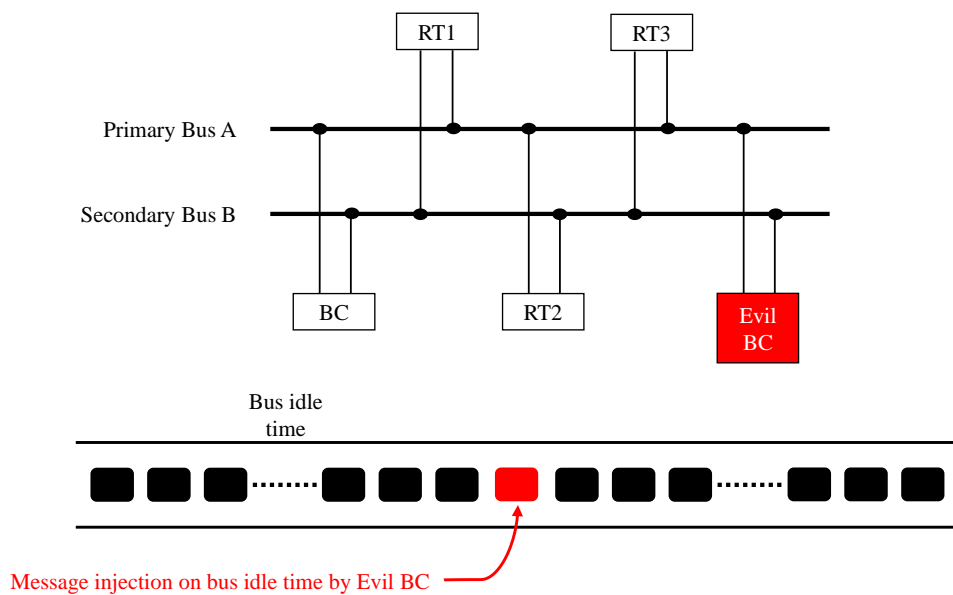


Figure 12: Evil BC Injecting Messages During Idle Slots

### 3.2.2 Attack Scenario 2

In this scenario, an existing remote terminal, specifically a V/UHF radio unit, is physically replaced with a compromised one. The compromised RT is designed to mimic the functionality of the removed device, featuring the same model transmitter. Although both devices share identical hardware and software configurations, minor differences exist due to slight variations in manufacturing tolerances, which result in unique hardware fingerprints. The objective of this scenario is to analyze the perfor-

mance of the proposed methods using the same model transmitters to assess their effectiveness when the sync signals of the compromised and legitimate RTs are closely matched.

This attack scenario corresponds to a compromise of integrity in the security framework outlined in Table 1. A compromised RT could send fake data words to alter the content of messages being shared between radios, leading to misinformation or operational disruptions.

Figure 13 illustrates a compromised remote terminal replaced with a device of the same functionality, transmitting illegitimate messages.

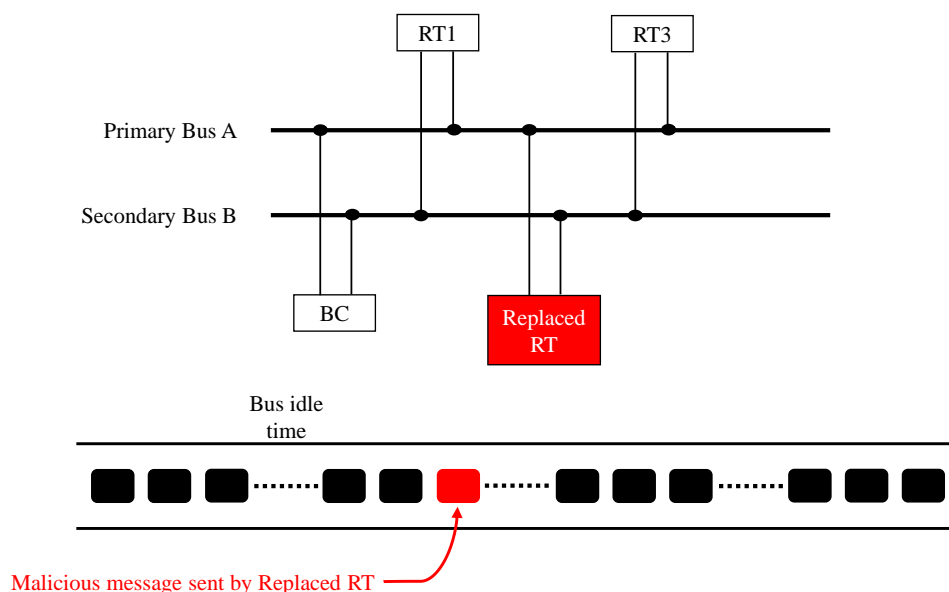


Figure 13: Replaced RT Transmitting Malicious Messages

### 3.3 Experiments and Datasets

Three distinct experiments are conducted to analyze the effectiveness of hardware fingerprinting in detecting unauthorized transmissions. These experiments focus on the two attack scenarios previously described, ensuring a comprehensive evaluation of the proposed intrusion detection system. They aim to evaluate the system’s ability to distinguish legitimate devices from unauthorized ones under controlled scenarios. Each experiment is designed to address a specific question about the system’s performance.

#### 3.3.1 Experiment 1

Experiment 1 evaluates the performance of supervised and unsupervised machine learning and deep learning methods in detecting attacks described in scenarios 1 and 2.

For supervised learning, datasets are labeled in a multiclass format to enable identification of all devices connected to the bus. In multiclass classification problems, ensuring balanced class distributions is crucial, as imbalanced datasets can lead to biased models that underperform for minority classes. Models trained on imbalanced data often overpredict majority classes, reducing accuracy for under-represented devices. To address this, datasets are prepared with equal sample sizes from each device, ensuring robust training for supervised methods.

For unsupervised learning, datasets are labeled in a binaryclass format to identify anomalous devices connected to the bus. In binaryclass classification, the focus shifts to distinguishing between normal and anomalous signals, rather than identifying specific devices. The datasets are structured such that signals from legitimate devices are labeled as 'normal', while those from unauthorized or compromised devices, such as the Evil BC, are labeled as 'anomalous'. This approach is particularly effective for detecting new or previously unseen threats, as it does not rely on predefined device identities.

For Experiment 1, two datasets are prepared, corresponding to attack scenarios 1 and 2. Dataset 1 is created for attack scenario 1 by collecting 70,000 signals from the bus, including 1 BC, 1 Evil BC, and 5 RTs, with 10,000 signals captured from each device. It is used for training supervised methods, with a multiclass labeling format to identify all devices connected to the bus. For unsupervised learning, which focuses on identifying patterns in the data without predefined labels, signals from the Evil BC are excluded, resulting in a training dataset of 60,000 signals from legitimate devices. A test dataset is then created for attack scenario 1, consisting of approximately 2,000 signals from each legitimate device and 1% malicious signals from the Evil BC. This test dataset is used to evaluate both supervised and unsupervised methods.

For attack scenario 2, dataset 2 is prepared with a total of 60,000 signals, including 1 BC, 4 RTs, and 1 unauthorized RT, with 10,000 signals collected from each device. It is used for training supervised methods, with a multiclass labeling format to identify all devices connected to the bus. Similar to dataset 1, signals from the unauthorized RT are excluded from the training dataset for unsupervised learning, resulting in a dataset of 50,000 signals from legitimate devices. A test dataset is created for Attack Scenario 2, consisting of approximately 2,000 signals from each legitimate device and 1% malicious signals from the unauthorized RT. This test dataset is used to evaluate both supervised and unsupervised methods. For this dataset, Device 1 and Device 7 are identical, representing the same model of transmitters. This scenario evaluates the system's ability to distinguish between these closely matched devices based on their unique hardware fingerprints.

The structure of the datasets are designed to ensure balanced and representative data for effective training and testing. Equal sample sizes from each device are chosen to address class imbalance, which could otherwise lead to biased models favoring majority classes. The inclusion of 1% malicious signals in the test dataset simulates realistic scenarios in avionics systems, where anomalies are rare yet crucial to detect.

### **3.3.2 Experiment 2**

Experiment 2 aims to determine the minimum number of voltage samples per sync bit required to maintain the same success rate in classification. The focus is on evaluating the performance of learning methods under varying feature configurations to identify the optimal setup. This optimization seeks to

enhance runtime efficiency for learning algorithms, reduce computational resource requirements, and improve the scalability of the model.

Instead of generating a new dataset, the data from attack scenarios 1 and 2 are rearranged to match the selected feature configurations. Voltage samples are progressively reduced to 300, 150, 100, 50, and 25, and the impact on classification performance is evaluated.

A feature reduction process is applied to determine the optimal number of features needed for accurate classification. Successful classification with fewer voltage samples minimizes model complexity and computational overhead, making the system more efficient while preserving detection accuracy.

### **3.3.3 Experiment 3**

Experiment 3 assesses the effectiveness of the selected methods in handling voltage changes, recognizing that bus voltage signals can fluctuate over time due to environmental factors. To evaluate this, dataset 3 is prepared specifically for experiment 3, focusing on the system's ability to maintain accuracy despite these variations.

The experiment is conducted in the system integration laboratory, isolating the analysis from the effects of external conditions such as heat and vibration on signal changes. Voltage signals are recorded at four distinct time intervals:  $T_0$ ,  $T_0+2.5h$ ,  $T_0+5h$ , and  $T_0+7.5h$ , to capture the natural progression of signal variation over time. This setup provides a controlled environment to evaluate how well the methods adapt to temporal changes in voltage signals.

## **3.4 Preprocessing, Model and Training**

Building on the experiments conducted and the datasets prepared, this section focuses on the machine and deep learning models employed to detect unauthorized transmissions on the MIL-STD-1553 bus.

### **3.4.1 Preprocessing**

Raw voltage signals captured directly from the MIL-STD-1553 bus are utilized for analysis. The captured signals are used as-is, without applying normalization, standardization, or any other preprocessing techniques. This decision is based on the fact that the signals are inherently clean and free from noise due to the controlled environment of the system integration laboratory.

Each signal is represented by 300 features, corresponding to the voltage samples that comprise the synchronization signal. No feature selection mechanisms are applied in experiments 1 and 3, as the entire signal is necessary to preserve the structural integrity of the synchronization pattern. The only exception is in experiment 2, where feature reduction is applied to determine the minimum number of voltage samples required for accurate identification.

This straightforward preprocessing approach ensures that the raw characteristics of the voltage signals are preserved.

### 3.4.2 Model and Training

Supervised and unsupervised learning methods are employed to evaluate the performance of the proposed intrusion detection system. Supervised models, including Random Forest, K-Nearest Neighbors, Support Vector Machines, and Long Short-Term Memory, are implemented using well-known Python libraries such as scikit-learn [26]. Similarly, unsupervised methods, such as Isolation Forest, One-Class SVM, and Local Outlier Factor, are also employed. The parameters used for the learning methods are detailed in Table 6. Unless otherwise specified, all other parameters are set to their default configurations.

For supervised learning, the datasets are partitioned into 60% for training, 20% for validation, and 20% for testing. Anomalies are uniformly distributed within the datasets, ensuring balanced representation in the training, validation, and testing splits. Conversely, unsupervised methods are trained exclusively on benign data to identify anomalies by distinguishing normal patterns from deviations.

Table 6: Parameters of Learning Algorithms

<i>Method</i>	<i>Used Parameters</i>
RF	n_estimators = 100, criterion = 'gini'
KNN	n_neighbors = 1, weights = 'uniform', algorithm = 'brute', metric = 'minkowski'
SVM	C = 1.0, kernel = 'rbf', gamma = 0.05
LSTM	epoch = 20, units = 16, activation = 'tanh', recurrent_activation = 'sigmoid'
OCSVM	kernel = 'rbf', nu = 0.5
IF	n_estimators = 100, contamination = 0.01
LOF	n_neighbors = 5, contamination = 0.01, novelty = true

### 3.5 Evaluation

To assess the performance of the proposed intrusion detection system, standard evaluation metrics are derived from the confusion matrix, including True Negative, True Positive, False Negative, and False Positive.

- **True Negative (TN)** represents the number of negative instances correctly identified by the model in binary classification.
- **True Positive (TP)** represents the number of positive instances correctly identified by the model in binary classification.
- **False Negative (FN)** represents the number of positive instances incorrectly identified as negative by the model in binary classification.
- **False Positive (FP)** represents the number of negative instances incorrectly identified as positive by the model in binary classification.

The following metrics are computed to evaluate model performance:

- **Accuracy (AC)** is calculated as the ratio of the total number of correct predictions to the total number of predictions.

$$AC = \frac{TN + TP}{TN + TP + FN + FP} \quad (1)$$

- **Precision (PR)** is calculated as the ratio of true positives to the total number of predicted positive instances.

$$PR = \frac{TP}{TP + FP} \quad (2)$$

- **Recall (RC)** is calculated as the ratio of true positives to the total number of actual positive instances.

$$RC = \frac{TP}{TP + FN} \quad (3)$$

- **F1-Score** is calculated as the harmonic mean of precision and recall.

$$F1 = 2 * \frac{PR * RC}{PR + RC} \quad (4)$$

- **False Positive Rate (FPR)** is calculated as the ratio of false positives to the total number of actual negative instances.

$$FPR = \frac{FP}{FP + TN} \quad (5)$$

## CHAPTER 4

### EXPERIMENTAL RESULTS

This chapter presents the experimental results obtained from evaluating the proposed IDS for MIL-STD-1553. The experiments are designed to assess the system's performance across various scenarios, including distinguishing legitimate devices from unauthorized ones, handling reduced feature sets, and adapting to temporal variations in bus voltage signals.

The primary objective of these experiments is to validate the effectiveness of supervised and unsupervised machine learning and deep learning methods in detecting anomalies within the bus system. Metrics such as Accuracy, Precision, Recall, F1-Score, and False Positive Rate are employed to provide a comprehensive evaluation of the models.

The experiments are structured to address key challenges in anomaly detection for MIL-STD-1553, including the rarity of malicious signals, the need for real-time detection, and the protocol's deterministic nature. The findings presented in this chapter aim to demonstrate the practicality and reliability of the proposed IDS while highlighting areas for further improvement.

#### 4.1 Results of Experiment 1

Experiment 1 evaluates the performance of various supervised and unsupervised learning methods in detecting unauthorized transmissions on the MIL-STD-1553 bus under two distinct attack scenarios. These scenarios include an evil bus controller injecting fake commands during idle periods and a compromised remote terminal impersonating a legitimate one. Building on the dataset structure and attack design detailed in the methodology chapter, this experiment focuses solely on reporting and interpreting the detection results. The goal is to identify which models most effectively classify legitimate and illegitimate devices based on their voltage signal characteristics, and to assess their precision, recall, and overall robustness in a multiclass or anomaly detection setting, depending on the approach used.

Table 7 presents the classification performance of supervised learning methods applied in Experiment 1 Attack Scenario 1.

Table 7: Results of Experiment 1 Attack Scenario 1

<i>Supervised Methods</i>	<i>Device 1</i>	<i>Device 2</i>	<i>Device 3</i>	<i>Device 4</i>	<i>Device 5</i>	<i>Device 6</i>	<i>Evil BC</i>	<i>FPR</i>	<i>AC</i>	<i>RC</i>	<i>PR</i>	<i>F1</i>
Validation RF, KNN, SVM, LSTM	<i>Device 1</i>	2021	0	0	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 2</i>	0	2036	0	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 3</i>	0	0	1988	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 4</i>	0	0	0	1966	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 5</i>	0	0	0	0	1998	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 6</i>	0	0	0	0	0	2002	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Evil BC</i>	0	0	0	0	0	0	118	0.0000	1.0000	1.0000	1.0000
Test RF, KNN, SVM, LSTM	<i>Device 1</i>	2064	0	0	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 2</i>	0	2007	0	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 3</i>	0	0	2001	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 4</i>	0	0	0	1957	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 5</i>	0	0	0	0	1983	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 6</i>	0	0	0	0	0	1987	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Evil BC</i>	0	0	0	0	0	0	121	0.0000	1.0000	1.0000	1.0000

Table 8 presents the classification performance of unsupervised learning methods applied in Experiment 1 Attack Scenario 1.

Table 8: Results of Experiment 1 Attack Scenario 1

<i>Unsupervised Methods</i>	<i>TN</i>	<i>FP</i>	<i>TP</i>	<i>FN</i>	<i>FPR</i>	<i>AC</i>	<i>RC</i>	<i>PR</i>	<i>F1</i>
OCSVM	6036	5964	120	0	0.4970	0.5079	1.0000	0.0197	0.0387
IF	11635	365	120	0	0.0304	0.9699	1.0000	0.2474	0.3967
LOF	11993	7	120	0	0.0006	0.9994	1.0000	0.9449	0.9717

Table 9 presents the classification performance of supervised learning methods applied in Experiment 1 Attack Scenario 2.

Table 9: Results of Experiment 1 Attack Scenario 2

<i>Supervised Methods</i>		<i>Device 1</i>	<i>Device 2</i>	<i>Device 3</i>	<i>Device 4</i>	<i>Device 5</i>	<i>Device 7</i>	<i>FPR</i>	<i>AC</i>	<i>RC</i>	<i>PR</i>	<i>F1</i>
<i>Validation</i> RE, KNN, SVM, LSTM	<i>Device 1</i>	2018	0	0	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 2</i>	0	2017	0	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 3</i>	0	0	1986	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 4</i>	0	0	0	1993	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 5</i>	0	0	0	0	2001	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 7</i>	0	0	0	0	0	96	0.0000	1.0000	1.0000	1.0000	1.0000
<i>Test</i> RE, KNN, SVM, LSTM	<i>Device 1</i>	1973	0	0	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 2</i>	0	2042	0	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 3</i>	0	0	2013	0	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 4</i>	0	0	0	1968	0	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 5</i>	0	0	0	0	2017	0	0.0000	1.0000	1.0000	1.0000	1.0000
	<i>Device 7</i>	0	0	0	0	0	98	0.0000	1.0000	1.0000	1.0000	1.0000

Table 10 presents the classification performance of unsupervised learning methods applied in Experiment 1 Attack Scenario 2.

Table 10: Results of Experiment 1 Attack Scenario 2

<i>Unsupervised Methods</i>	<i>TN</i>	<i>FP</i>	<i>TP</i>	<i>FN</i>	<i>FPR</i>	<i>AC</i>	<i>RC</i>	<i>PR</i>	<i>F1</i>
OCSVM	6019	3981	100	0	0.3981	0.6058	1.0000	0.0245	0.0478
IF	9769	231	100	0	0.0231	0.9771	1.0000	0.3021	0.4640
LOF	9993	7	100	0	0.0007	0.9993	1.0000	0.9346	0.9662

The classification results presented in Tables 7 and 8 highlight the comparative performance of supervised and unsupervised learning methods for detecting an unauthorized bus controller in Experiment 1 under Attack Scenario 1.

As illustrated in Table 7, all supervised models (RF, KNN, SVM, and LSTM) achieve perfect classification performance on both validation and test sets. Each legitimate device and the unauthorized bus controller are correctly identified, with no misclassifications or false positives in any class. All evaluation metrics, including Accuracy, Precision, Recall, F1-Score, and False Positive Rate, remain at their optimal values, confirming reliable detection. The consistency between validation and test results demonstrates that the models generalize well and are not overfitting to the training data. These findings highlight the effectiveness of supervised learning for fingerprint-based detection when trained on a balanced and representative dataset of sync signals.

In contrast, Table 8 provides the outcomes of unsupervised methods. Among them, LOF clearly outperforms the others, achieving a very low false positive rate (0.0006) and high scores across all metrics, including an F1-Score of 0.9717. IF also performs well, though with a slightly lower precision (0.2474) and F1-Score (0.3967), still indicating a reasonable level of anomaly detection. On the other hand, OCSVM struggles significantly, with a false positive rate close to 0.5 and poor precision and F1-Score values, indicating a high level of misclassification of legitimate signals as anomalies.

Table 9 and Table 10 demonstrate the classification performance of supervised and unsupervised methods for Experiment 1 under Attack Scenario 2, in which a legitimate remote terminal is replaced by a compromised device of the same model.

As shown in Table 9, all supervised learning models (RF, KNN, SVM, and LSTM) achieve perfect classification on both the validation and test sets. Every device is correctly identified, with no misclassifications or false positives, resulting in F1-scores of 1.0000 across the board. The consistent performance on unseen validation data confirms that the models generalize well and are not overfitting. These results demonstrate the models' ability to accurately distinguish between transmitters, even when two devices—such as Device 1 and the compromised Device 7—have identical hardware and software. This highlights the strength of hardware fingerprinting in capturing subtle electrical differences caused by minor manufacturing variations.

Table 10 highlights the results from unsupervised methods. Among them, LOF achieves near-perfect performance, detecting the unauthorized transmitter with high precision (0.9346) and an F1-Score of 0.9662, while maintaining a very low false positive rate (0.0007). IF also performs well, though with moderately lower precision and F1-Score. In contrast, OCSVM continues to exhibit poor performance, with high false positive rates and limited detection capability.

To further reinforce the effectiveness and reliability of the supervised learning models in Experiment 1, additional performance analyses are introduced. These visualizations not only illustrate how the models distinguish between devices using voltage signal data, but also serve to validate the training process. In particular, they help demonstrate that the models are not overfitting and maintain generalization capability across unseen data.

Figure 14 displays the feature importance rankings obtained from the Random Forest model, highlighting which voltage sample indices within the 300-point sync signal contribute most to classification accuracy. The plot reveals clear peaks between indices 0–50 and 150–200, indicating that these regions contain the most discriminative information for identifying individual transmitters. This aligns

with prior visual comparisons of voltage signals from same-model devices (Figure 10), where notable physical variations were also concentrated in these segments. The alignment between learned feature importance and empirical signal characteristics confirms that classification decisions are based on meaningful, device-specific electrical features rather than random noise. Additionally, this focused importance distribution supports the claim that the model is not overfitting, as it avoids relying on scattered or arbitrary features and instead identifies consistent patterns across the dataset.

Figure 15 presents a heatmap of the absolute coefficient values from the SVM model trained on the full 300 sample input. The plot reveals two high-impact regions: around indices 20–50 and 150–180. These segments are responsible for most of the model’s decision boundary formation, while the remaining regions exhibit low coefficients, implying minimal contribution. This confirms that the SVM model, even though it uses a different learning mechanism than Random Forest, focuses on similar temporal regions of the sync signal.

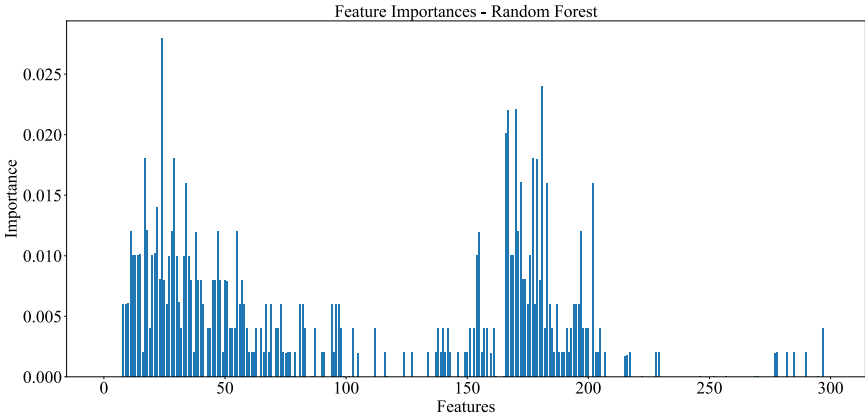


Figure 14: Feature Importance Derived from Random Forest Model

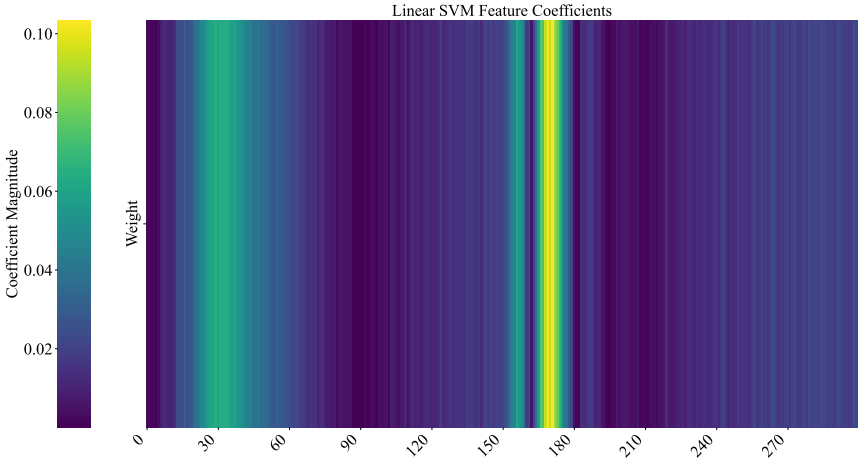


Figure 15: Heatmap Visualization of the Absolute Values of SVM Coefficients

In parallel, the training process of the LSTM model is illustrated in Figure 16, which shows both training and validation loss curves over epochs. The two curves decrease consistently and remain closely aligned throughout the training. The absence of a significant gap between them indicates strong generalization and minimal overfitting. This alignment reinforces the model’s reliability and supports the high classification accuracy observed across both attack scenarios.

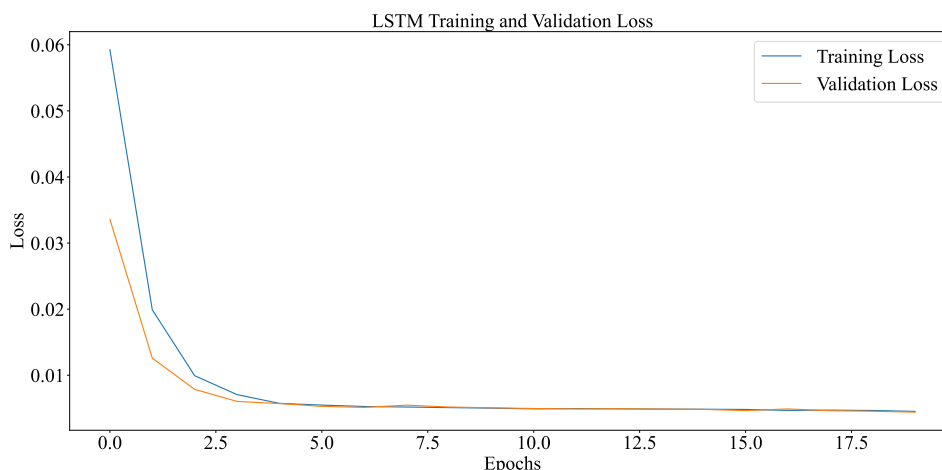


Figure 16: Training and Validation Loss Curves of LSTM Model

## 4.2 Results of Experiment 2

To evaluate the impact of feature reduction on detection performance, Experiment 2 assessed the system’s ability to distinguish legitimate devices from unauthorized ones across both Attack Scenarios 1 and 2. The objective was to determine the minimum number of voltage samples per sync signal required to maintain high classification accuracy, thereby reducing computational overhead without compromising detection capability.

The dataset, originally consisting of 300 voltage samples per sync signal, was progressively reduced to 150, 100, 50, and 25 samples. Feature reduction was applied uniformly across all sync signals to preserve the structure of the waveform and ensure minimal distortion of signal characteristics. It was observed that the results remain consistent across all supervised learning methods, indicating that beyond a certain point, the model’s performance becomes insensitive to additional voltage samples. This suggests that the same level of accuracy can be maintained with fewer features, and further increasing the feature count does not yield performance gains. Given this stability, the results for supervised models are excluded from detailed presentation for simplicity, and the focus is placed on unsupervised methods where feature reduction has a more notable impact.

Table 11 summarizes the performance metrics of each unsupervised method across varying voltage sample counts for detecting an Evil BC in Attack Scenario 1.

Table 11: Results of Experiment 2 Attack Scenario 1

Method	Sample Count	TN	FP	TP	FN	FPR	AC	RC	PR	F1
OCSVM	300	6036	5964	120	0	0.4970	0.5079	1.0000	0.0197	0.0387
	150	6028	5972	120	0	0.4977	0.5073	1.0000	0.0197	0.0386
	100	5983	6017	120	0	0.5014	0.5035	1.0000	0.0196	0.0384
	50	5971	6029	120	0	0.5024	0.5026	1.0000	0.0195	0.0383
	25	5951	6049	120	0	0.5041	0.5009	1.0000	0.0195	0.0382
IF	300	11635	365	120	0	0.0304	0.9699	1.0000	0.2474	0.3967
	150	11793	207	120	0	0.0173	0.9829	1.0000	0.3670	0.5369
	100	11469	531	120	0	0.0442	0.9562	1.0000	0.1843	0.3113
	50	11177	823	120	0	0.0686	0.9321	1.0000	0.1273	0.2258
	25	11028	972	120	0	0.0810	0.9198	1.0000	0.1099	0.1980
LOF	300	11993	7	120	0	0.0006	0.9994	1.0000	0.9449	0.9717
	150	11990	10	120	0	0.0008	0.9992	1.0000	0.9231	0.9600
	100	11987	13	120	0	0.0011	0.9989	1.0000	0.9023	0.9486
	50	11981	19	120	0	0.0016	0.9984	1.0000	0.8633	0.9266
	25	11977	23	120	0	0.0019	0.9981	1.0000	0.8392	0.9125

Among the methods, OCSVM consistently underperformed, with false positive rates approaching 50% at all sampling levels, including full 300 samples. The method failed to discriminate between legitimate and malicious signals, resulting in extremely low precision and F1-scores below 0.04. Isolation Forest demonstrated moderate performance. At 150 samples, it achieved an accuracy of 0.9829 and an F1-score of 0.5369, marking its best result. However, performance declined steadily with fewer samples, with the F1-score dropping to 0.1980 and FPR rising to 8.1% at 25 samples. LOF showed the most robust performance, consistently delivering high accuracy ( $\geq 0.9980$ ) and low FPR ( $\leq 0.2\%$ ) across all sample counts. Even at 25 samples, it achieved an F1-score of 0.9125 and precision of 0.8392, demonstrating strong resilience to feature reduction.

Figure 17 shows how feature reduction influences anomaly detection accuracy by comparing the F1-Score performance of OCSVM, IF, and LOF at different voltage sample counts.

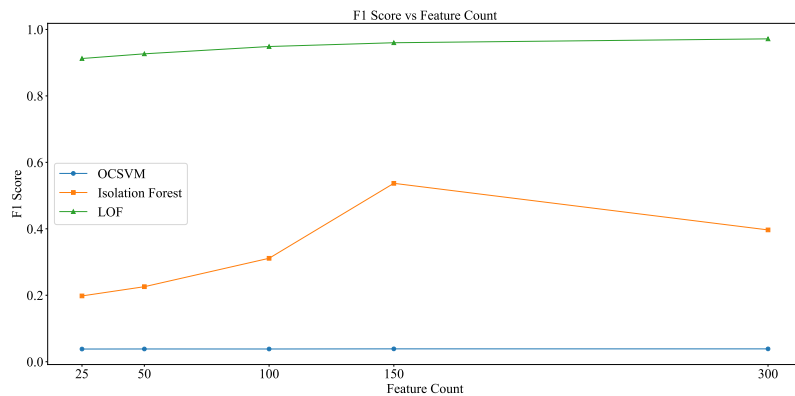


Figure 17: F1-Score of Unsupervised Methods vs Voltage Sample Count, Attack Scenario 1

Table 12 presents the performance unsupervised learning methods evaluated under reduced feature conditions for detecting a compromised RT in Attack Scenario 2.

Table 12: Results of Experiment 2 Attack Scenario 2

<i>Method</i>	<i>Sample Count</i>	<i>TN</i>	<i>FP</i>	<i>TP</i>	<i>FN</i>	<i>FPR</i>	<i>AC</i>	<i>RC</i>	<i>PR</i>	<i>F1</i>
OCSVM	300	6019	3981	100	0	0.3981	0.6058	1.0000	0.0245	0.0478
	150	6009	3991	100	0	0.3991	0.6049	1.0000	0.0244	0.0477
	100	5951	4049	100	0	0.4049	0.5991	1.0000	0.0241	0.0471
	50	5948	4052	100	0	0.4052	0.5988	1.0000	0.0241	0.0470
	25	5272	4728	99	1	0.4728	0.5318	0.9900	0.0205	0.0402
IF	300	9769	231	100	0	0.0231	0.9771	1.0000	0.3021	0.4640
	150	9753	247	100	0	0.0247	0.9755	1.0000	0.2882	0.4474
	100	9701	299	100	0	0.0299	0.9704	1.0000	0.2506	0.4008
	50	9566	434	100	0	0.0434	0.9570	1.0000	0.1873	0.3155
	25	8992	1008	100	0	0.1008	0.9002	1.0000	0.0903	0.1656
LOF	300	9993	7	100	0	0.0007	0.9993	1.0000	0.9346	0.9662
	150	9990	10	100	0	0.0010	0.9990	1.0000	0.9091	0.9524
	100	9987	13	100	0	0.0013	0.9987	1.0000	0.8850	0.9390
	50	9981	19	100	0	0.0019	0.9981	1.0000	0.8403	0.9132
	25	9977	23	100	0	0.0023	0.9977	1.0000	0.8130	0.8969

OCSVM demonstrated consistently poor performance, even at full feature resolution. Its FPR remained high (39.81% at 300 samples) and increased further with feature reduction, reaching 47.28% at 25 samples. Precision and F1-Score were notably low across all sample sizes, indicating that OCSVM struggled to distinguish normal signals from anomalies, leading to excessive false positives. In contrast, Isolation Forest provided moderate performance, with an FPR of 2.31% at 300 samples and a steady decline in accuracy as the sample size was reduced. While it maintained a perfect True Positive Rate throughout, its precision dropped significantly at lower resolutions. For instance, at 25 samples, the FPR increased to 10.08%, and the F1-Score fell to 0.1656. LOF consistently outperformed the other methods, maintaining high accuracy and low false positives across all sample sizes. At 300 samples, LOF achieved near-perfect detection with an FPR of only 0.07% and an F1-Score of 0.9662. Even with only 25 samples, LOF maintained an FPR of 0.23% and an F1-Score of 0.8969, demonstrating strong resilience to feature reduction.

Figure 18 illustrates the comparative performance of OCSVM, IF, and LOF in terms of F1-Score across varying numbers of voltage samples, highlighting the impact of feature reduction on anomaly detection accuracy.

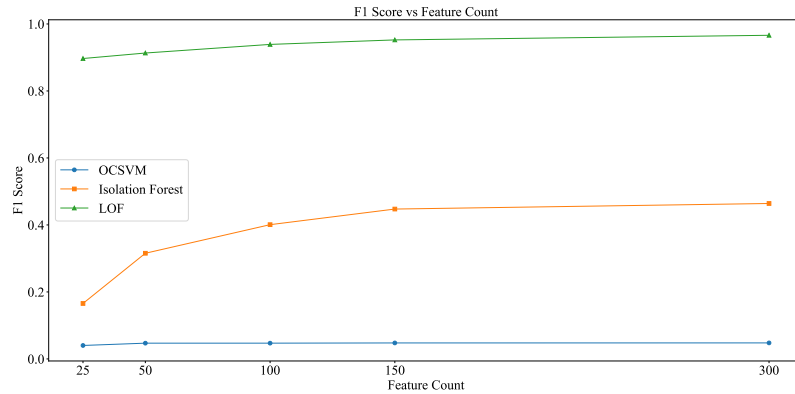


Figure 18: F1-Score of Unsupervised Methods vs Voltage Sample Count, Attack Scenario 2

The results obtained from both Attack Scenario 1 and Attack Scenario 2 clearly demonstrate the varying robustness of unsupervised learning methods under different levels of feature reduction. Across both scenarios, LOF consistently outperformed other methods, maintaining high detection accuracy and precision even with a reduced number of voltage samples. This resilience makes LOF especially suitable for practical applications where computational efficiency is critical. Isolation Forest showed moderate performance, performing well with higher sample counts but degrading at lower resolutions. OCSVM, on the other hand, consistently underperformed, failing to distinguish malicious activity effectively due to high false positive rates and poor precision. Overall, the findings confirm that significant reductions in feature count—down to 100 or even 50 samples—are feasible without sacrificing detection accuracy, especially when using robust algorithms like LOF.

To better understand why performance remained high even with reduced feature counts and to justify the structured feature reduction applied in Experiment 2, a Spearman correlation analysis was conducted on the 300-sample waveform features. The heatmap in Figure 19 reveals strong correlations among adjacent voltage sample features, forming several pronounced blocks along the diagonal. These blocks indicate temporally local segments within the sync signal where values evolve gradually, resulting in high redundancy. This block-wise structure confirms that much of the waveform’s information is repetitive across short intervals. In contrast, correlations diminish as the distance between samples increases, highlighting that more widely spaced features capture distinct signal behavior. These observations validate the downsampling strategy adopted in Experiment 2, where selected samples were retained while their immediate neighbors were excluded. The preserved classification performance, even at reduced resolutions, demonstrates that the models did not overfit to noise but instead leveraged stable and discriminative features inherent in the waveform. This supports the robustness of the approach and reinforces the argument for its efficiency and generalization capability.

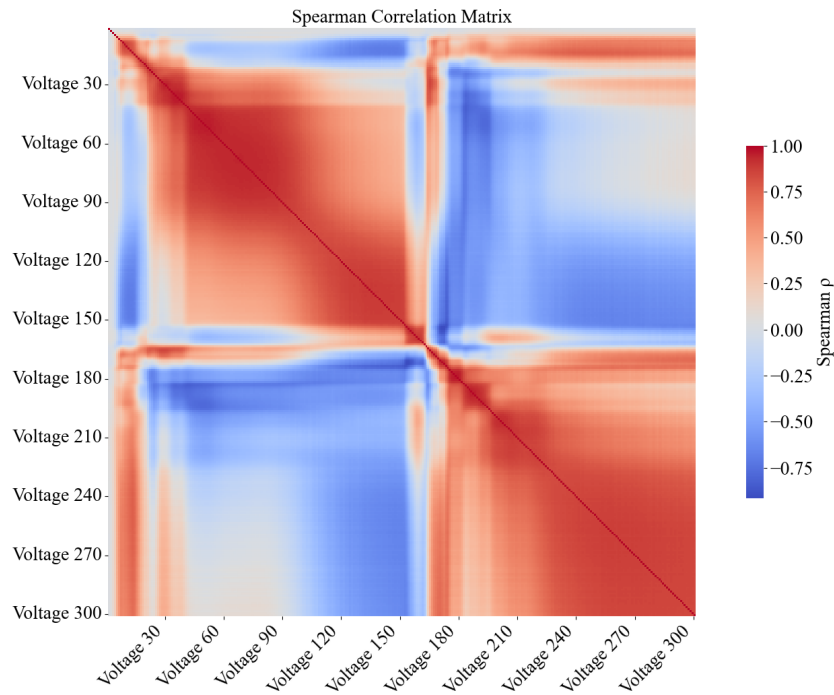


Figure 19: Spearman Correlation Matrix of Voltage Sample Features

### 4.3 Results of Experiment 3

The third experiment focuses on evaluating whether the proposed detection methods remain reliable as signal characteristics on the MIL-STD-1553 bus shift over time. Rather than accounting for external influences such as heat or vibration, which are intentionally held constant within the system integration laboratory, this experiment explores subtle internal variations that may emerge as devices operate for extended periods.

Training is carried out using data collected at the initial time point T0. The performance of the trained models is then assessed using data gathered at three subsequent intervals: 2.5 hours, 5 hours, and 7.5 hours after the initial capture. If the voltage signals change meaningfully during this period, the models may fail to classify them accurately, since they were not exposed to those shifts during training. This setup helps reveal whether the detection system can tolerate temporal variations without retraining or recalibration.

In this experiment, only supervised learning methods are considered. These models demonstrated strong performance in the first experiment. The goal is to determine whether their high classification success carries over when exposed to signals collected at later stages. Since any shift in voltage patterns might affect the structure of the sync signals used for identification, a noticeable decline in accuracy would suggest that time-dependent changes are significant enough to hinder detection reliability.

Table 13 displays the classification results of the supervised learning models when trained on data from T0 and tested on signals collected at later time intervals.

Table 13: Results of Experiment 3

<i>Dataset</i>	<i>Method</i>	<i>TN</i>	<i>FP</i>	<i>TP</i>	<i>FN</i>	<i>FPR</i>	<i>AC</i>	<i>RC</i>	<i>PR</i>	<i>FI</i>
T0 + 2.5h	RF	16190	0	166	0	0.0000	1.0000	1.0000	1.0000	1.0000
	KNN	16190	0	166	0	0.0000	1.0000	1.0000	1.0000	1.0000
	SVM	16190	0	166	0	0.0000	1.0000	1.0000	1.0000	1.0000
	LSTM	16190	0	166	0	0.0000	1.0000	1.0000	1.0000	1.0000
T0 + 5h	RF	17750	0	181	0	0.0000	1.0000	1.0000	1.0000	1.0000
	KNN	17750	0	181	0	0.0000	1.0000	1.0000	1.0000	1.0000
	SVM	17750	0	181	0	0.0000	1.0000	1.0000	1.0000	1.0000
	LSTM	17750	0	181	0	0.0000	1.0000	1.0000	1.0000	1.0000
T0 + 7.5h	RF	17780	0	165	0	0.0000	1.0000	1.0000	1.0000	1.0000
	KNN	17780	0	165	0	0.0000	1.0000	1.0000	1.0000	1.0000
	SVM	17780	0	165	0	0.0000	1.0000	1.0000	1.0000	1.0000
	LSTM	17780	0	159	6	0.0000	0.9937	0.9636	1.000	0.9815

During the initial stages of testing at 2.5h and 5h, all supervised models including Random Forest, K-Nearest Neighbors, Support Vector Machine, and Long Short-Term Memory achieved consistent and flawless results. Metrics such as Accuracy, Precision, Recall, and F1-Score remained at maximum values, suggesting that no noticeable variation had occurred in the voltage signals to affect classification reliability.

By the 7.5-hour mark, the overall performance of the models remained strong. RF, KNN, and SVM continued to detect all anomalous cases without any false predictions. In contrast, the LSTM model experienced a small drop in performance, with six positive cases going undetected. This led to a minor reduction in recall and a slightly lower F1-Score. However, the model produced no false positives.

These findings indicate that the supervised approaches used in this study are largely stable over several hours of operation within controlled lab conditions. While LSTM shows slight sensitivity to time-based signal changes, the impact remains limited. For longer operational durations or more dynamic environments, additional strategies may be needed to maintain consistent detection performance with time-dependent models.

#### 4.4 Further Analysis of Signal Distributions

While the results of all three experiments demonstrate near-perfect classification performance, it is essential to understand the underlying reasons for this high accuracy. To provide further insight, the statistical properties of the sync signal samples from different devices were analyzed. By examining metrics such as mean, standard deviation, skewness, and kurtosis, highlighting the inherent separability of the voltage signals used for classification was aimed. These statistical measures help characterize the signal distributions. They reveal how distinctive the patterns are across devices and explain why learning models were able to perform so well.

Figure 20 shows a scatter plot of the mean and standard deviation calculated from the sync signals. As shown, each device forms a tightly clustered and well-separated group in the feature space. This

separation implies that even using only two basic statistics, signals from different transmitters can be easily distinguished.

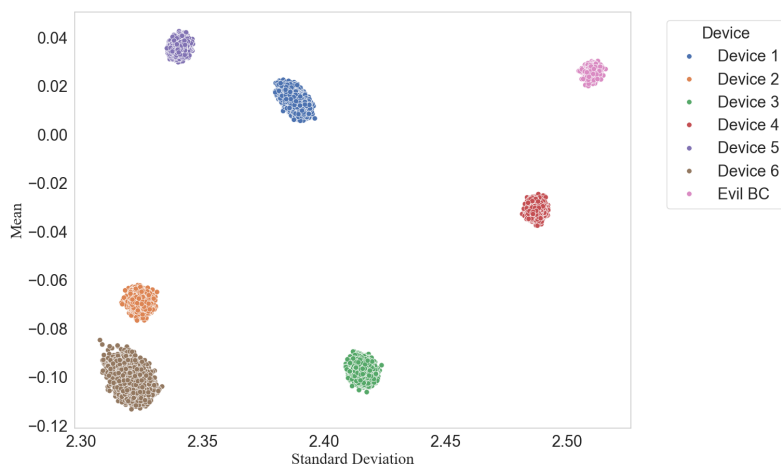


Figure 20: Mean vs Standard Deviation for Sync Signals Across Devices

To further validate the discriminative power of sync signal characteristics, Figure 21 presents another scatter plot, this time using skewness and kurtosis. Again, distinct and non-overlapping clusters are evident. It suggests that sync signals exhibit stable higher-order statistical properties unique to each device.

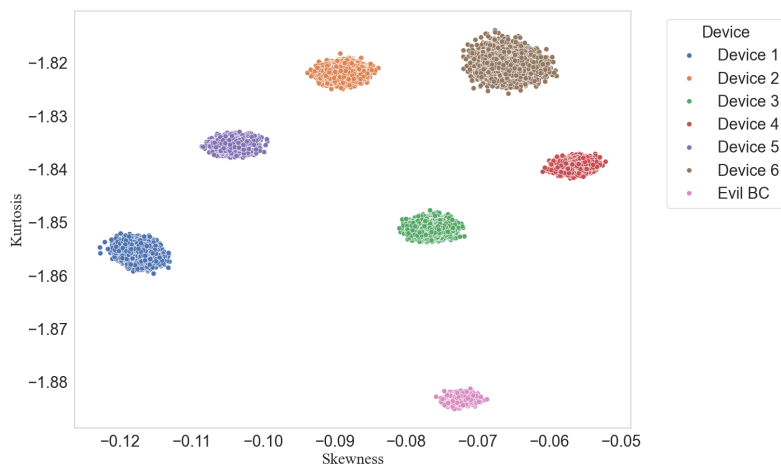


Figure 21: Skewness vs Kurtosis for Sync Signals Across Devices

These plots collectively support the claim that the classification task is statistically well-posed. The signal-level distinctions between devices are significant even under minimal feature transformations. This explains why machine and deep learning models can perform so well without requiring much complexity. The distributions are already well-separated because of device-specific hardware-level characteristics.

## CHAPTER 5

### CONCLUSION

In this study, an intrusion detection approach based on hardware fingerprinting was developed to enhance the security of MIL-STD-1553 communication networks. Recognizing that legacy systems like MIL-STD-1553 lack integrated security features, especially in today's increasingly connected avionic platforms, the proposed method focused on leveraging the subtle voltage pattern differences in sync signals to authenticate transmission sources without modifying the existing certified architecture.

The experimental evaluations provided strong evidence supporting the effectiveness of the proposed system. In Experiment 1, machine and deep learning models achieved perfect classification performance in distinguishing between legitimate and unauthorized devices. Experiment 2 demonstrated that reducing the number of voltage samples per sync signal, while maintaining high detection performance, was feasible. This finding shows that model complexity and computational requirements can be significantly reduced without sacrificing classification accuracy. Experiment 3 investigated the robustness of the trained models over time, revealing that model performance remained stable across several hours of operation, with only a minor recall decrease observed for the LSTM model.

To understand the reasons behind these results, statistical analyses were conducted using measures such as standard deviation, mean, skewness, and kurtosis. These analyses confirmed that signals from different transmitters exhibit naturally separable distributions due to hardware-specific characteristics. This inherent separability simplifies the classification task, allowing relatively simple models to achieve exceptional performance.

The research questions posed at the beginning of this study are addressed as follows:

- What are the performance outcomes of machine and deep learning methods applied to hardware fingerprinting for the MIL-STD-1553 communication bus?

Both supervised and unsupervised learning methods achieved near-perfect results, validating the effectiveness of hardware fingerprinting in the context of MIL-STD-1553.

- What are the key features that can be used in ML/DL models for hardware fingerprinting, and how can feature optimization improve model performance?

The voltage samples forming the sync signal were identified as key features. Feature importance analysis revealed that specific regions of the sync waveform contributed most to classification success. Feature reduction experiments confirmed that performance could be maintained with fewer samples, optimizing both runtime efficiency and model scalability..

- Which methods offer an optimal solution for the avionics domain by minimizing false positive rates without compromising security?

Supervised methods such as Random Forest, K-Nearest Neighbors, Support Vector Machine, and Long Short-Term Memory demonstrated outstanding performance, achieving perfect detection with zero false positives under experimental conditions. Among unsupervised methods, Local Outlier Factor consistently delivered the best results..

- How does the performance of the model change over time, given that signal characteristics may vary during runtime?

Over a monitored period of 7.5 hours, no significant degradation was observed in most models. Only a slight decrease in recall was noted in the LSTM model at the final time interval, suggesting high robustness to temporal variations in bus signals under controlled laboratory conditions.

While the results achieved in this study are promising, future work could focus on extending these findings by exploring more deterministic approaches for device authentication. Techniques such as statistical analysis, transform methods like Fast Fourier Transform, or hybrid statistical-ML models could be investigated to further enhance reliability and interpretability. Additionally, testing under more diverse environmental conditions and real flight scenarios can be essential to validate the system's resilience against environmental factors such as temperature shifts, vibration, and electromagnetic interference.

Overall, this work lays a solid foundation for the development of lightweight and accurate intrusion detection mechanisms tailored for securing legacy avionics communication systems in an increasingly connected world.

## REFERENCES

- [1] Y. Kim, J.-Y. Jo, and S. Lee, “ADS-B vulnerabilities and a security solution with a timestamp,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, no. 11, pp. 52–61, 2017.
- [2] A. Hagl, “Threat modeling and security improvements for ethernet-based avionic networks,” in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, pp. 1–7, 2024.
- [3] S. Al Ghafri, “Navigating the skies of cybersecurity: Unveiling real-world attacks and risk management in aviation,” tech. rep., Civil Aviation Authority, November 2023. Accessed: 2024-12-03.
- [4] Aeronautical Radio INC, “ARINC specification 664 p7-1: Aircraft data network, part 7 - avionics full duplex switched ethernet network,” 2009.
- [5] “IEEE 802.1 time-sensitive networking (tsn) working group.” <https://1.ieee802.org/>.
- [6] United States Department of Defense, “MIL-STD-1553 B aircraft internal time division command/response multiplex data bus,” 1978.
- [7] E. Levy, N. Maman, A. Shabtai, and Y. Elovici, “Anomili: Spoofing hardening and explainable anomaly detection for the 1553 military avionic bus,” *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–17, 2024.
- [8] M. A. Elsayed, M. Wrana, Z. Mansour, K. Lounis, S. H. H. Ding, and M. Zulkernine, “Adaptids: Adaptive intrusion detection for mission-critical aerospace vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23459–23473, 2022.
- [9] F. Onodueze and D. Josyula, “Anomaly detection on MIL-STD-1553 dataset using machine learning algorithms,” in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 592–598, 2020.
- [10] S. J. J. Génereux, A. K. H. Lai, C. O. Fowles, V. R. Roberge, G. P. M. Vigeant, and J. R. Paquet, “MAIDENS: MIL-STD-1553 anomaly-based intrusion detection system using time-based histogram comparison,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 1, pp. 276–284, 2020.
- [11] M. Evcil, Z. Yuce Tok, I. C. Babir, M. A. Gökyer, B. Bozkurt, and S. Akleylek, “Hardware fingerprinting using machine and deep learning methods on MIL-STD-1553,” in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, pp. 1–9, 2024.
- [12] H. Sathaye, G. Noubir, and A. Ranganathan, “On the implications of spoofing and jamming aviation datalink applications,” in *Proceedings of the 38th Annual Computer Security Applications Conference, ACSAC '22*, (New York, NY, USA), p. 548–560, Association for Computing Machinery, 2022.

- [13] Z. Wu, T. Shang, and A. Guo, "Security issues in automatic dependent surveillance - broadcast (ads-b): A survey," *IEEE Access*, vol. 8, pp. 122147–122167, 2020.
- [14] M. L. Salgado and M. S. de Sousa, "Cybersecurity in aviation: the stpa-sec method applied to the tcas security," in *2021 10th Latin-American Symposium on Dependable Computing (LADC)*, pp. 1–10, 2021.
- [15] O. Stan, Y. Elovici, A. Shabtai, G. Shugol, R. Tikochinski, and S. Kur, "Protecting military avionics platforms from attacks on MIL-STD-1553 communication bus," *CoRR*, vol. abs/1707.05032, 2017.
- [16] K. Lounis, Z. Mansour, M. Wrana, M. A. Elsayed, S. H. H. Ding, and M. Zulkernine, "A review and analysis of attack vectors on MIL-STD-1553 communication bus," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 6, pp. 5586–5606, 2022.
- [17] S. Mukkamala, A. Sung, and A. Abraham, "Cyber security challenges: Designing efficient intrusion detection systems and antivirus tools," in *Enhancing Computer Security with Smart Technology* (V. R. Vemuri, ed.), pp. 125–161, Boca Raton, New York, USA: Auerbach Publications, 2006.
- [18] D. Parsons, "SANS 2023 ICS/OT cybersecurity survey," tech. rep., SANS Institute, September 2023. Accessed: 2024-11-09.
- [19] Cybersecurity Insiders, "State of network threat detection report," 2024. Accessed: 2024-11-16.
- [20] K. Scarfone and P. Mell, "Guide to intrusion detection and prevention systems," tech. rep., National Institute of Standards and Technology, Gaithersburg, MD, February 2007. Accessed: 2024-11-09.
- [21] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Springer Open Cybersecurity*, vol. 2, no. 20, 2019.
- [22] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX Conference on System Administration, LISA '99*, (USA), p. 229–238, USENIX Association, 1999.
- [23] O. Stan, A. Cohen, Y. Elovici, and A. Shabtai, "Intrusion detection system for the MIL-STD-1553 communication bus," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 3010–3027, 2020.
- [24] E. Alpaydin, *Machine Learning*. The MIT Press, 08 2021.
- [25] "Stumpy." <https://stumpy.readthedocs.io/en/latest/>.
- [26] "scikit-learn." <https://scikit-learn.org/>.