ANOMALY DETECTION OF MIL-STD 1553 TRAFFIC: MACHINE LEARNING METHODS AND REALISTIC SIMULATION EVALUATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HÜSEYİN SAĞIRKAYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONIC ENGINEERING

September 2023

Aproval of the thesis:

ANOMALY DETECTION OF MIL-STD 1553 TRAFFIC: MACHINE LEARNING METHODS AND REALISTIC SIMULATION EVALUATION

submitted by **HÜSEYİN SAĞIRKAYA** in partial fulfillment of the requirements for the degree of Choose an item. i**n** Choose an item.**, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. İlkay Ulusoy
Head of the Department, **Electrical and Electronics Eng.** _____

Prof. Dr. Ece Güran Schmidt
Supervisor, **Electrical and Electronics Eng, METU** _____

Examining Committee Members:

Prof. Dr. Asım Egemen Yılmaz
Electrical and Electronics Eng, Ankara University _____

Prof. Dr. Ece Güran Schmidt
Electrical and Electronics Eng, Middle East Technical University _____

Assist. Prof. Dr. Serkan Sarıtaş
Electrical and Electronics Eng, Middle East Technical University _____

Date: 06.09.2023

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 Last name : Hüseyin Sağırkaya

Signature :

ABSTRACT

ANOMALY DETECTION OF MIL-STD 1553 TRAFFIC: MACHINE LEARNING METHODS AND REALISTIC SIMULATION EVALUATION

Sağırkaya, Hüseyin
Master of Science, Electrical and Electronic Engineering
Supervisor : Ece Güran Schmidt

September 2023, 84 pages

In this thesis, we evaluate K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Naïve Bayes, Logistic Regression and Decision Tree machine learning (ML) methods for the anomaly detection of MIL-STD 1553 traffic to support cybersecurity. MIL-STD 1553 is a very widely used communication bus for military avionics systems. The fault tolerance features of MIL-STD 1553 target the safety and robustness of the aircraft. However, there is no built-in support against malicious attacks. Such cybersecurity issues are raised because of the increased connectivity of the MIL-STD 1553 to the outside world particularly for maintenance and diagnostics reasons. An imitated remote terminal and bus controller can behave as a member of the bus to change the data or corrupt the data and traffic to prevent messaging or stop communication. Furthermore, cyber security attacks such as denial-of-service can cause bus scheduling failure.

In the scope of this thesis, we identify attack scenarios and MIL-STD 1553 message features that can be used for anomaly detection. We construct a testbed with real avionics hardware and a simulator that can generate attack messages. We inject the messages generated by the simulator into the MIL-STD 1553 bus using a PCIe card that is connected to the PC with the simulator. Furthermore, we employ bus monitoring and analysis tools to collect data. To this end, we modify the driver of the PCIe card and write software to parse and analyze the traffic data.

We perform anomaly detection with the selected ML algorithms and compare their results.

Keywords: MIL-STD 1553, Avionics, Machine Learning, Anomaly Detection, Machine Learning

ÖZ

MIL-STD 1553 TRAFİĞİNDE ANOMALİ TESPİTİ: MAKİNE ÖĞRENMESİ YÖNTEMLERİ VE GERÇEKÇİ SİMÜLASYON İLE DEĞERLENDİRME

Sağırkaya, Hüseyin
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Tez Yöneticisi: Ece Güran Schmidt

Eylül 2023, 84 sayfa

Bu tezde, MIL-STD 1553 haberleşmesinde siber güvenliği desteklemek üzere anomali tespit etmek için en yakın komşu algoritması, destek vektör makinası, naïve bayes, lojistik regresyon ve karar ağacı makine öğrenme metodları değerlendirilmektedir. MIL-STD 1553 askeri aviyonik sistemlerde yaygın olarak kullanılan bir veri yoludur. MIL-STD 1553 standardının hataya dayanıklılık özellikleri uçakların emniyetli ve güvenilir olmasını hedeflemektedir. Ancak kötü niyetli saldırılara karşı yerleşik ve tanımlı bir çözüm mevcut değildir. Uçağın operasyonel ömrü boyunca karşılaşabileceği siber güvenlik problemleri giderek önem kazanmaktadır çünkü MIL-STD 1553 veri yolunun bakım ve teşhis amaçlı çözümlerin gelişmesi ile birlikte dış dünya ile bağlantı ihtiyaçları da artmaktadır. MIL-STD 1553 haberleşmesinde gerçekte veri yolunda bağlı olmayan terminaller veri yolu terminali gibi davranarak haberleşmede yer alan mesajları değiştirebilir ya da haberleşmeyi kesmek amacı ile mesajlaşmayı bozabilir ya da durdurabilir. Ayrıca veri yolunu kullanım dışı bırakma amacı ile tasarlanan siber saldırılar hatalı haberleşmeye neden olabilmektedir.

Bu tez kapsamında, siber saldırı senaryoları ve anomali tespiti amacı ile MIL-STD 1553 mesaj özellikleri tanımlanmaktadır. Gerçek aviyonik cihazların ve saldırı senaryolarını yaratabilmek amacı ile simülatörün kullanıldığı bir test ortamı kurulmaktadır. PC'de yer alan PCIe kart ve simülatör tarafından oluşturulan mesajlar MIL-STD 1553 veri yoluna yönlendirilmektedir. Ayrıca veri yolu takip ve

analiz yazılımları ile mesajlar kayıt edilmektedir. Bu amaçla, PCIe kart sürücü yazılımı geliştirilerek ve haberleşme verisini ayrıştırabilen analiz edecek yazılım hazırlanmaktadır. Seçilen makine öğrenme algoritmaları ile anomali tespiti sağlanarak sonuçlar değerlendirilmektedir.

Anahtar Kelimeler: MIL-STD 1553, Simülasyon, Siber Güvenlik, Aviyonik Sistemler, Makine Öğrenmesi

To My Beloved Family

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

AFDX : Avionics Full Duplex Switched Ethernet

API : Application Programming Interface

ASCII : American Standard Code for Information Interchange

ATLAS : Advanced Test Lab with Automated Simulators

AUC : Area Under the Curve

BC : Bus Controller

BM : Bus Monitor

CSMA/CD : Carrier Sense Multiple Access With Collision Detection

CVR/FDR : Cockpit Voice Recorder / Flight Data Recorder

DDC : Data Device Corporation

DoD : Department of Defence

DVDR : Digital Video and Data Recorder

EGI : Embedded GPS/INS System

FADEC : Full Authority Digital Engine Control

FCS : Flight Control System

HUD : Head Up Display

IFF : Identification Friend or Foe

LRU : Line Replaceable Unit

K-NN : K-Nearest Neighbor

ML : Machine Learning

MMR : Multi Mode Receiver

Mbit/s : Megabit per second

OS : Operating System

PCIe : Peripheral Component Interconnect Express

RADALT : Radio Altimeter

ROC : Receiver Operation Characteristic

RT : Remote Terminal

TTP : Time-triggered protocol

TDMA : Time-division multiple access

SAE : Society of Automotive Engineers

UFCP : Up Front Control Panel

V/UHF : VHF and UHF

LIST OF SYMBOLS

SYMBOLS

$\sum$: Summation

μ : Micro

σ : Sigma

π : Pi

e : Exponential

log : Logaritmic

CHAPTER 1

INTRODUCTION

Avionics means aviation electronics. The development of avionics has caused various data bus technologies in the aviation industry instead of point-to-point communication which needs more cabling and bigger connectors. Data transmission means data is sent from source to destination using data communication techniques. For the integration of avionics, different communication techniques have been developed and standardized such as MIL-STD-1553 (1553). These standards have been used in different aircraft to transfer data between various avionics. A bus is categorized as serial or parallel considering the physical connection. In serial communication all data bits are sent using the same wire synchronously or asynchronously however data is sent via separate wires simultaneously in parallel communication. Serial communication has different types such as single or multiple sources or sinks. 1553 is a dual redundant time division multiplexing serial bus protocol invented by the Department of Defense of the United States connecting multiple sources to multiple sinks. In 1553 architecture, The Bus Controller (BC) terminal has the task of initiating data transfers on the data bus via command word. The Remote Terminal (RT) is responsible for sending the information with data word as well as status word. The Bus Monitor (BM) terminal is assigned the task of receiving bus traffic and extracting data on the bus to record and analyze. It is an accepted and widely used standard in military aviation, vehicles, ships, and satellites and is expected to be used in the future since it is very powerful to set up reliable and robust communication. It is designed to achieve successful integration and reliable data transmission in hard military environment conditions such as high temperature and vibration, high reliability, high fault tolerance, redundancy, error detection, and good electromagnetic compatibility.

Avionics systems tests are necessary to verify the robustness of the design since avionics are necessarily safety critical equipment of an aircraft [17]. Avionics integration considers the architecture and also functional verification of the avionics. It is used during verification to utilize certification studies. To this end, it incorporates validation of the requirements and verification of the design. In order to verify the design, simulation is the necessary solution since functionality of the avionics equipment can be achieved. When the system is integrated in the laboratory, it is not possible to activate all the related messages of the equipment since they are not fully operational. Also aircraft behaviour in flight is needed to be created to achieve realistic integration. The functional behavior models of avionics in the test environment leverage the success of the integration process. As a result, simulators are commonly used in integration environment to provide all the test and integration scenarios.

Anomaly or in other words, anomaly in the bus means a message which is different from the messages generated by bus terminals and remote terminals. Anomalous traffic can be created via the test equipment during the maintenance activities of the aircraft or equipment which is not a part of the aircraft bus architecture or modified equipment. Systems have lots of parts that are developed in various production facilities centers, and supply chains [23]. Consequently, validating all the equipment parts becomes complex and difficult causing these parts to be installed in the aircraft causing the open point for the planned access to the aircraft during operation in a nonsecure ground facility. Supply chain platforms with wireless interfaces and data links for maintenance activities can generate vulnerabilities in defense systems. Monitoring the bus for illegal activity can be an effective solution to success cyber security in the defense industry.

The contributions of the thesis are as follows:

1) Detailed simulation evaluation of 1553 with a commercial simulation tool.

We use commercial software that simulates avionics and creates bus communication messages as in real aircraft. ATLAS, Advanced Test Lab with Automated Simulators [26], is a modeling and simulation program providing simulator and test automation tool structure for avionic systems developed by Simsoft Information Technologies.

The simulator generates message traffic that contains data defined as anomalies from a known or an unknown RT. We define related data as anomalous or malicious data since it is not an authorized message that is defined in bus controller of the 1553 bus. The communication protocol parameters can be adjusted with the model interface structure of the simulator.

The simulator runs on a PC and sends data to 1553 network using a PCIe 1553 card that is installed on that PC. The card can be used as an RT or BC to achieve message transmission. We use 1553 PCIe cards from Altadata [26] configured as BC. Also in order to record data and simulate malicious RT Altadata and Ballard bustools are used. Ballard [28] bustool is used as BM and Altadata bustool is used for malicious RT.

For a card to function and to have an interface with the PC's software, the driver specific to the card is required to be installed on the PC using API functions. The API functions provide the mechanisms of the card via code blocks written in C#. In the scope of this thesis work, we develop a driver interfacing with the API of the card. Furthermore, we enabled the 1553 PCIe card software to read and write a standard table format such as .csv or .xls file.

Other hardware requirements for the testbed are integration with twin axial cable and 1553 coupler providing the connection between BC, RTs, and BM. The generated data is monitored using Ballard [27] and Altadata [26] bus tools and USB to 1553 converters.

2) Providing realistic datasets and 1553 configurations that are reflecting real-life use cases.

We create a collection of realistic data sets using simulators and real integration environment. These datasets are published with this thesis.

2) Evaluating the machine learning models for anomaly detection for MIL-STD-1553 protocol.

We compare different machine learning models under different data sets in terms of correctness and speed for a two minutes recorded dataset. We evaluate K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Naïve Bayes, Logistic Regression and Decision Tree machine learning (ML) models to detect anomalous traffic using the improved simulator environment for realizing the attack scenarios.

The models are created using Google Colab, Jupyter and Visual Studio code software tools. The models work offline to detect if any malicious software and hardware are placed in the aircraft. We use various dataset properties will be used in different machine-learning algorithms to detect anomalous traffic.

The remainder of this thesis is organized as follows. Chapter II discusses previous work on cyber security of 1553 bus. Chapter III defines background information such as 1553 standard description, bus topology, failure management, machine learning models, attack scenarios and dataset properties. Chapter IV defines simulation environment such as test set-up, driver integration and dataset and parser information. Chapter V defines the feature selection process and dataset properties. Chapter VI defines the results of the machine learning models and performance evaluation. Chapter VII is the conclusion giving a brief of evalution and future work.

CHAPTER 2

PREVIOUS WORK

There are various research works on detecting anomalies in the MIL-STD-1553 bus messages. Losier et al.'s research defines the anomaly using a histogram method to detect it in terms of time [1]. In their simulation setup, a PC with a PCI slot integrated into a DDC MIL-STD-1553 simulation board is used. Then the recorded data is processed via C programming language using the Bus Tool library. Their dataset is a reference to offline processes instead of real-time processing. Each feature in the dataset is evaluated in terms of frequencies plotted as a histogram. They reference a baseline for each feature and compared it with the processed cases. If the value is over the baseline value as a percent target value, they labeled the data as anomalous. This method's disadvantage is histogram is compared offline.

Genereux at al's also used a histogram comparison approach which defines the anomalies in terms of time [2]. In their work, time-based features are referenced similar to Losier's work. Their work contains a test bench and simulation software so that training and prediction are decided in real time using histograms with features and data. They propose additional algorithms to increase the accuracy of the model such as bin width optimization algorithm, sliding window parser, and minimum window size algorithms. Anomalies in each bin are identified if there is a difference from the benign histogram exceeding a threshold.

Onodueze uses machine learning models to determine normal or malicious traffic using the benign dataset for training [3]. However, the testing dataset contains anomalies. The ML models are trained using publicly available datasets.

Stan et al. research testbed includes a simulator that is SimPy1 (Python Package. https://pypi.org/project/simpy/) to simulate MIL-STD-1553 messages connecting to the bus via a 1553 interface card [4]. Their work contains two experiments – one on an operational testbed and the other on records. Their anomaly detection is based on command data features and time features. Command word features are used for aperiodic messages, and command and timing features are profiled for periodic messages. These profiles are trained Markov chain models using benign datasets. The state transition probabilities of Markov models are computed. Transition time probabilities between messages are compared with a reference and anomalies are detected. Their work accuracy lowers when a realistic MIL-STD-1553 integration is analyzed with more than 3 RTs that are connected to the bus.

Levy et al. use deep learning-based classifiers to analyze the unique characteristics of the voltage signals [5]. Their work detects maliciously connected devices to the 1553 bus. They used an unsupervised deep learning algorithm to identify anomalous messages.

Krunal Sachdev treats the data on the bus as time series to analyze the sequential data when a property of the time series changes [22].

Yahalom created datasets for research but there is not enough data to train the models also attack scenarios are limited. Genereux and Losier created their dataset digitally by inserting attack data manually based on a physical simulation dataset.

We identify the main missing part of the previous works as the absence of real data. A realistic data generation can not be provided by these studies. Furthermore, the data sets are required to cover different phases of flight such as take-off or landing.

Table 2.1 Previous work on anomaly detection on MIL-STD 1553 data

Author	Setup	Detection Reference	Algorithms	Method
Losier et al	Hardware-Software	Simulated data	Histogram	Frequency difference
Genereux et al	Software	Bus recording data	Histogram	Density difference between times
Stan et al.	Hardware-Software	Simulated data	Markov Chain	IDS based on timing and command words
Onodueze et al/	Software	Publicly data	Deep Learning	ML algorithms on command and timing features

Table 2.2 Features used by the selected previous work on MIL-STD 1553 data.

Author	Features
Losier et al.	All data are identical, timing related features such as inter-message gap, periodicity, response time, data throughput
Genereux et al.	Time-based histogram comparison
Stan et al.	Time interval analysis of messages by inspecting deviations from normal time cycles
Onodueze et al.	Command word features; source and destination terminal address, subaddress, channel, word count, mode code, timing features time cycle

Table 2.3 Accuracy values of previous work

Author	Algorithm	Accuracy
Losier et al.	Histogram	Presence of anomaly
Genereux at al	Histogram	Start time (0.38-1.14) End Time (0.30-47.11)
Stan et al.	Supervised	98 %-99 %
Onodueze et al.	Supervised, unsupervised	15.9 %-95.9 %

Previous work uses various statistical or machine learning models to detect anomalies. Various solutions are proposed for MIL-STD 1553 anomaly detection. However, datasets used in the previous work are obtained from a simulation environment defining the attack messages manually or using the simulators. They are not realistic and do not present aircraft in different phases of flight. For training, they use a benign dataset recorded from a simulated 1553 setup. Furthermore, simulation times are limited. Attack scenarios are defined using command words. However, it is possible in reality that an unauthorized RT can inject malicious messages without receiving any command word.

CHAPTER 3

BACKGROUND

3.1 MIL-STD 1553 Protocol

MIL-STD 1553 is a Dual Redundant Time Division Multiplexing Serial Bus Protocol invented by the Department of Defence of the United States. The US DoD gave-up oversight of the standard (and most MIL-STD documents) in the early 1990s and the standard is now overseen by the Society of Automotive Engineers (SAE) as a commercial document "AS15531." It is the most widely used standard in military aviation, ground vehicles, military ships, and satellite systems. and is expected to be used in the future.

It is designed to achieve high integrity and reliable data transmission in an adverse military environment such as high temperature and vibration, high reliability, fault tolerance, redundancy, error detection, and good electromagnetic compatibility.

There are three types of terminals specified for use on the bus. The Bus Controller (BC) is the only terminal that can initiate data transfer on the bus. Remote Terminals (RT) are the sources and sinks of data. Bus Monitor (BM) Terminals listen to bus traffic for recording and analysis purposes but do not transmit or receive data on the bus. These three types of words are used together to form messages which allow communication between the bus controller and addressed RTs.

The multiplexed data bus system in its most elemental configuration is shown in Figure 3-1. There are three terminals which are Bus Controller, Remote Terminal, and Bus Monitor.

The Bus Controller (BC) terminal has the task of initiating information transfers on the data bus. The Bus Monitor (BM) terminal has the task of receiving bus traffic. Remote terminal (RT).

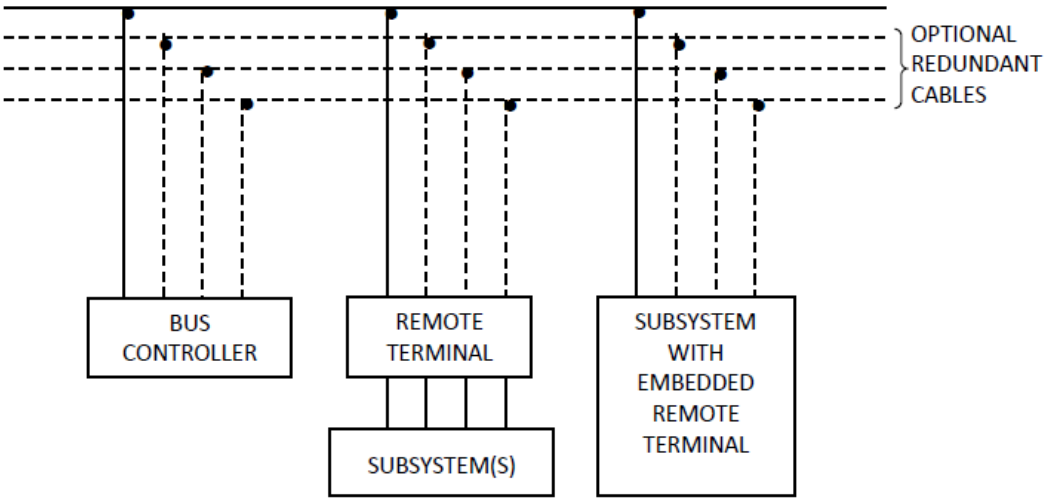


Figure 3-1 Data Bus Architecture [24]

The multiplexed data bus system functions asynchronously in a command/response mode and transmission occurs in a half-duplex manner. The information flow on the data bus is comprised of messages formed by three types of words (command, data, and status) as defined in Figure 3-2. The word size is 16 bits plus the sync waveform and the parity bit for a total of 20 bits times as shown in Figure 3-2. Each word includes 20 bits that include a 3-bit synchronization waveform, 16 information bits, and 1 parity bit. Command words are transmitted by the bus controller and contain 5 bit RT address field, a transmit/receive bit, a 5-bit subaddress/mode field, and a 5-bit word count/mode code field. Data words contain the information in 16 bits. Status words are only transmitted by RTs in response to valid commands and contain a 5-bit RT address and an 8-bit flag used for indicating RT status.

Status word is comprised of a sync waveform, RT address, message error bit, instrumentation bit, service request bit, three reserved bits, broadcast command received bit, busy bit, subsystem flag bit, dynamic bus control acceptance bit, terminal flag bit, and a parity bit.

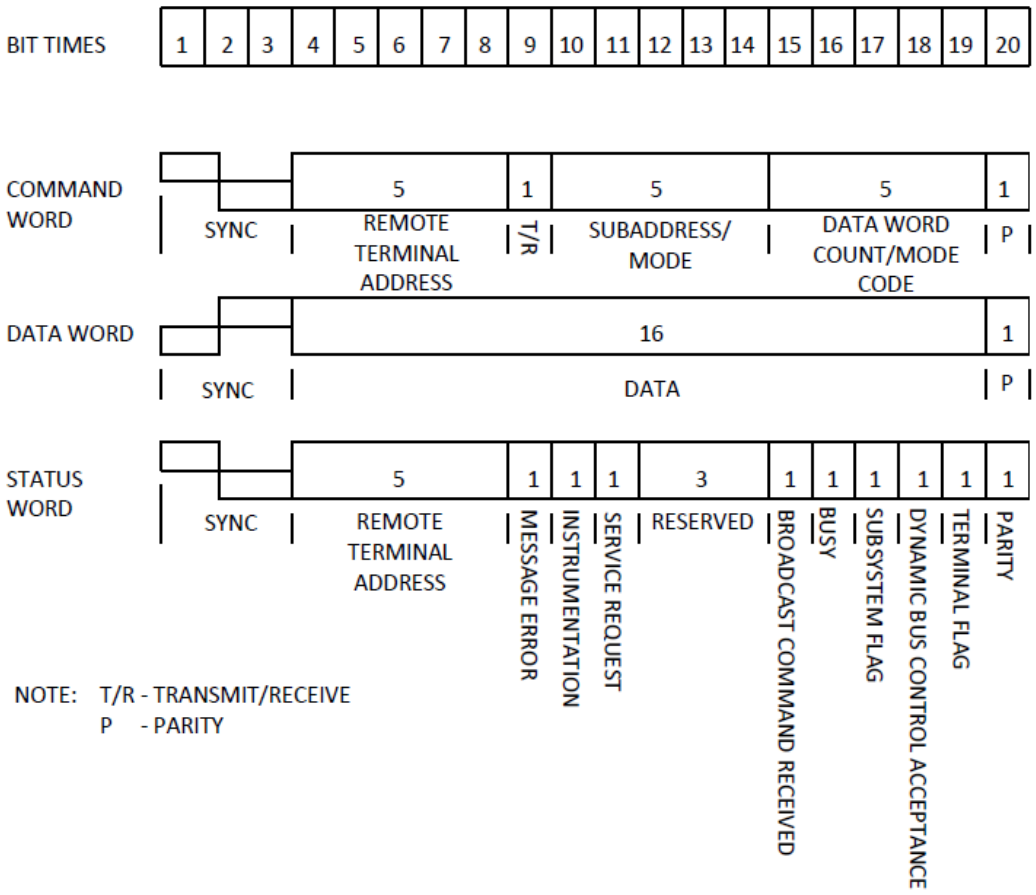


Figure 3-2 Word Formats [24]

There are four basic message formats: (BC) to (RT), (RT) to (BC) transmission, (RT) to (RT) transmission, mode code transmission, and broadcast transmission. In BC to RT messaging BC initiates RT to receive data via command word, in RT to BC messaging BC initiates RT to transmit data, in RT to RT messaging BC initiates data exchange between RTs. The RT accepts a command word if it contains a terminal address that matches the RT address via sending status and data

word. The mode code changes the operation mode of RT and in broadcast mode, all RTs suppress their status word transmission.

The messages transmitted on the data bus comply with the formats in Figure 3-3. The RT responds to a valid command word within the period of 4.0 to 12.0 μs. A terminal operating as a bus controller is responsible for sending data bus commands, participating in data transfers, receiving status responses, and monitoring system status as defined in this standard. The bus controller function is to control the data bus. Only one terminal is in active control of a data bus at any one time.

A remote terminal (RT) operates in response to valid commands received from the bus controller. The RT accepts a command word as valid when the command word meets the criteria and the command word contains a terminal address that matches the RT address. No combination of RT address bits, T/R bits, subaddress/mode bits, and data word count/mode code bits of a command word results in invalid transmissions by the RT. Subsequent valid commands shall be properly responded to by the RT.

Message Formats The standard defines ten types of message transmission formats. All of these formats are based on the three-word types which are defined above. In the figures that follow, the BC words (Command and Data Words) are in blue and RT response words (Status and Data Words) are in amber. The following facts and rules will help explain the caption of the figures:

The bus controller to remote terminal (BC-RT) message is referred to as the receive command since the remote terminal is going to receive data. (Note: For the 'BC Mode Code Without Data Word Message' type, the 'Transmit' bit is set)

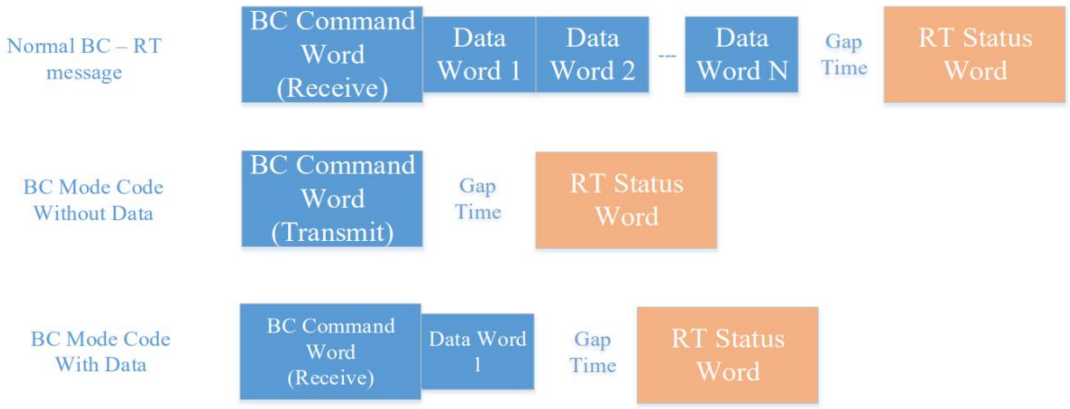


Figure 3-3 BC to RT Messages

The remote terminal to bus controller (RT-BC) message is referred to as a transmit command. The bus controller issues only a transmit command word to the remote terminal. The terminal, on validating the command word, transmits its status word followed by the number of data words requested by the command word.

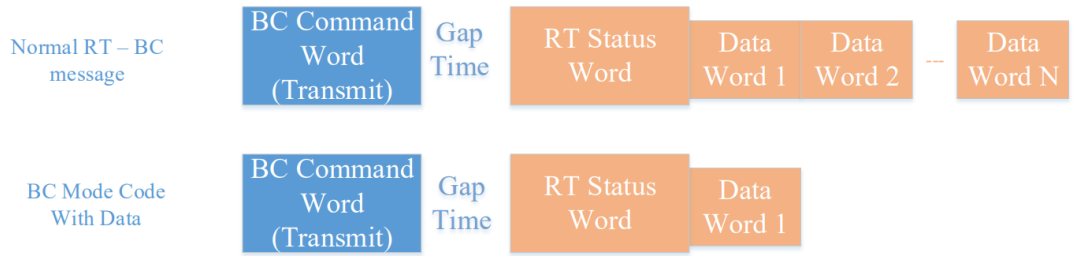


Figure 3-4 RT to BC Messages

The remote terminal to remote terminal (RT-RT) command allows a terminal (the data source) to transfer data directly to another terminal (the data sink) without going through the bus controller. However, the bus controller may also collect the data and use them. The bus controller issues a command word to the receiving terminal immediately followed by a command word to the transmitting terminal.

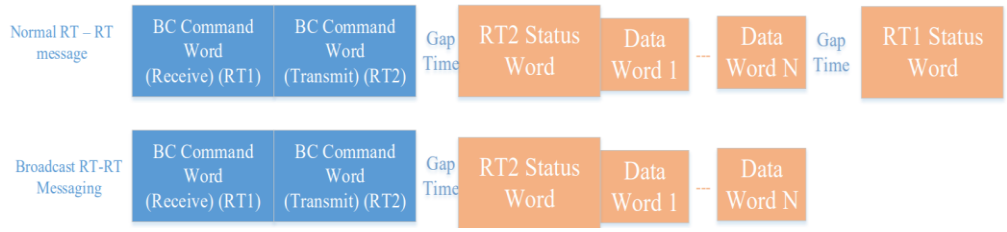


Figure 3-5 RT to RT Messages

BC Broadcast messages are implemented sporadically with different systems. A Broadcast message can save a lot of communication time, but avionics systems tend to like a closed-loop transmission scheme (the central computer wants to know that the message got delivered), so the use of Broadcast Commands is truly system dependent. BC sends a broadcast message via the address field to 11111B. The broadcast message is sent to all remote terminals However, since RTs will not respond the message cannot be resent in case of errors.

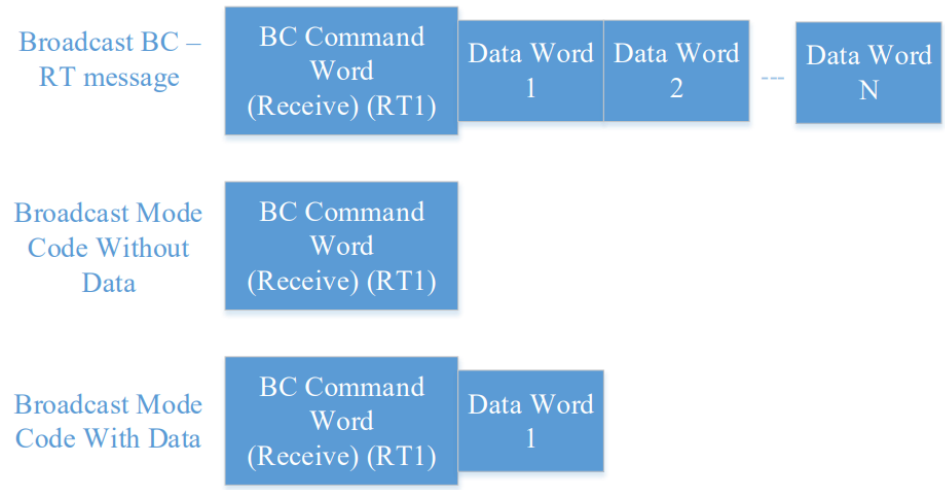


Figure 3-6 BC-RT(s) Broadcast Messages

To start mode code messaging BC sends mode code to RT(s) via the sub-address/mode field is set to 00000B or 11111B and the word count field to the mode code value.

Communication protocol choices of the model can be adjusted with the Model Interface structure.

3.2 Model Interface

For modeling the MIL-STD 1553, Subaddress and Word are needed to be defined in the modeling environment of ATLAS simulator [18]. Avionics messages are modeled defining the properties as in Tables 3.1, 3.2, and 3.3.

3.2.1 MIL-STD 1553 Interface

The interface structure in the MIL-STD-1553 protocol has the following fields.

Table 3.1 MIL-STD 1553 bus

MIL-STD-1553 bus	
Name	Unique name of the MIL-STD-1553 Bus. A Model Interface must have only one bus with the same name.
Endianness	Byte Order of the MIL-STD-1553 Bus Big Endian; MSB byte is stored first. Little Endian; LSB byte is stored first.

3.2.2 MIL-STD 1553 Subaddress

The subaddress structure in the MIL-STD-1553 protocol has the following fields.

Table 3.2 MIL-STD 1553 Subaddress

MIL-STD 1553 bus	
Name	Unique name of the MIL-STD 1553 subaddress. A Bus must have only one subaddress with the same name.
Subaddress No	Number identifier of the MIL-STD 1553 subaddress.
Transmit/Receive Mode	Communication Direction of the MIL-STD 1553 subaddress.
Frequency	To enable periodic communication, in Hertz.

3.2.3 MIL-STD 1553 Word

There are 32 words in a subaddress. The word structure in the MIL-STD-1553 protocol has the following fields.

Table 3.3 MIL-STD-1553 Word

MIL-STD-1553 bus	
Name	Unique name of the MIL-STD 1553 Word. A Subaddress must have only one word with the same name.
Word No	MIL-STD 1553 Word consists of 16 bits.

3.3 Messages

In this thesis to provide a MIL-STD 1553 bus messaging, avionics messages are modeled using ATLAS simulator and a simulator file is created since the equipment that generates these messages can not be fully functional on the ground. Messages are included in data words length of 16 bits.

The message list can be defined in Table 3.4. A detailed message list is given Appendix Message List MIL-STD 1553 scheduler excel file.

Table 3.4 Message List for DOS attack

RT	EQ	Message Description	T/R	SA	Word Count
Address1	V/UHF radio	Preset Sayısı	T	01	32
Address2	Transponder	Control Status	T	01	32
Address3	FADEC	-	T	01	32
Address4	Unknown RT	-	R	01	32

3.4 Bus Scheduling

To define a MIL-STD 1553 bus schedule the messages should be defined and categorized with their frequency. Message sending frequency can be defined as very low, low, medium, high, and urgent. The messages are assigned to time slots. These time slots define the main frames of MIL-STD 1553. These frames are defined by the bus controller software to schedule the communication on the bus. The major frame is the pre-defined time slot which should include all the periodic messages. Minor frames include messages with transmission gap times. The period of the minor frame can be the highest frequency message however major frame period matches the lowest frequency message on the bus schedule. The minor frame period should be less than the minimum period among aperiodic messages.

For the aperiodic messages a fixed time slot is determined in the major frame also. Following response times occur in the command word transmission.

Response time $4\ \mu s < T1 < 12\ \mu s < T2 \leq 14\ \mu s < T3$.

T1: response time

T2: late respose

T3: no response

There is not any time gap between data words that are part of the command message.

Gap times are an important part of the algorithms detecting the anomaly in the dataset.

The bus schedule of a fighter aircraft is defined as following frame times Table 3.5. The message list and frame times are defined in Appendix.

Table 3.5 MIL-STD-1553 Bus Scheduler

Message Timing			Bus Timing																			
			0	60	120	180	240	300	360	420	480	540	600	660	720	780	840	900	960	1120	1180	1240
60 mS	20 Hz	20-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
120 mS	10 Hz	10-1	1		1		1		1		1		1		1		1		1		1	
120 mS	10 Hz	10-1		1		1		1		1		1		1		1		1		1		1
250 mS	5 Hz	5-1	1				1				1				1				1			
250 mS	5 Hz	5-1		1				1				1				1				1		
250 mS	5 Hz	5-1			1				1				1				1				1	
250 mS	5 Hz	5-1				1				1				1				1				1
600 mS	2 Hz	2-1	1										1									
600 mS	2 Hz	2-2		1										1								
600 mS	2 Hz	2-3			1										1							

3.5 Bus Topology

Different topologies are possible for MIL-STD 1553. All terminals and the bus controller can be connected via using bus couplers. Single bus as defined in Figure 3-7 or multiple bus connections can be set up to provide multiple RT connections.

In Figure 3-8, aircraft MIL-STD 1553 architecture is defined. It is a multiple-bus configuration integration. There are three MIL-STD 1553 buses defined as Communication Avionics and FCS MUX buses. These buses provide enough redundancy to complete missions and safe flight. Each bus controller can be different but also in the same bus there can be two bus controllers where a special switching mechanism should be implemented to achieve a master back-up transition if needed. Critical data such as pilot settings, status, or health data on the bus should be forwarded from the master bus controller to backup bus controller via ethernet connection to provide redundancy. Also, there should be mode code or switching mechanisms to provide switching between the two.

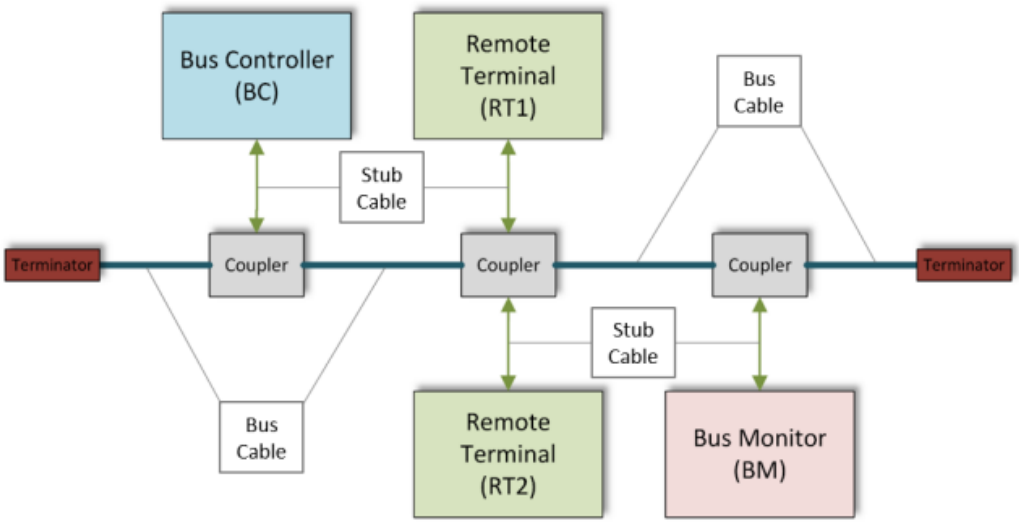


Figure 3-7 Single Bus Topology

In the architecture there are also other interfaces such as ARINC 429, RS 422/485, ARINC 664 and Ethernet. There are various equipment such as EGI, RADALT,

MMR, DVDR, Flight Control Computer, V/UHF Radio, Transponder, CVR/FDR, HUD, UFCP, audio control panel, remote interface unit, interface blanking unit, secondary power distribution units, primary power distribution units, engine control units.

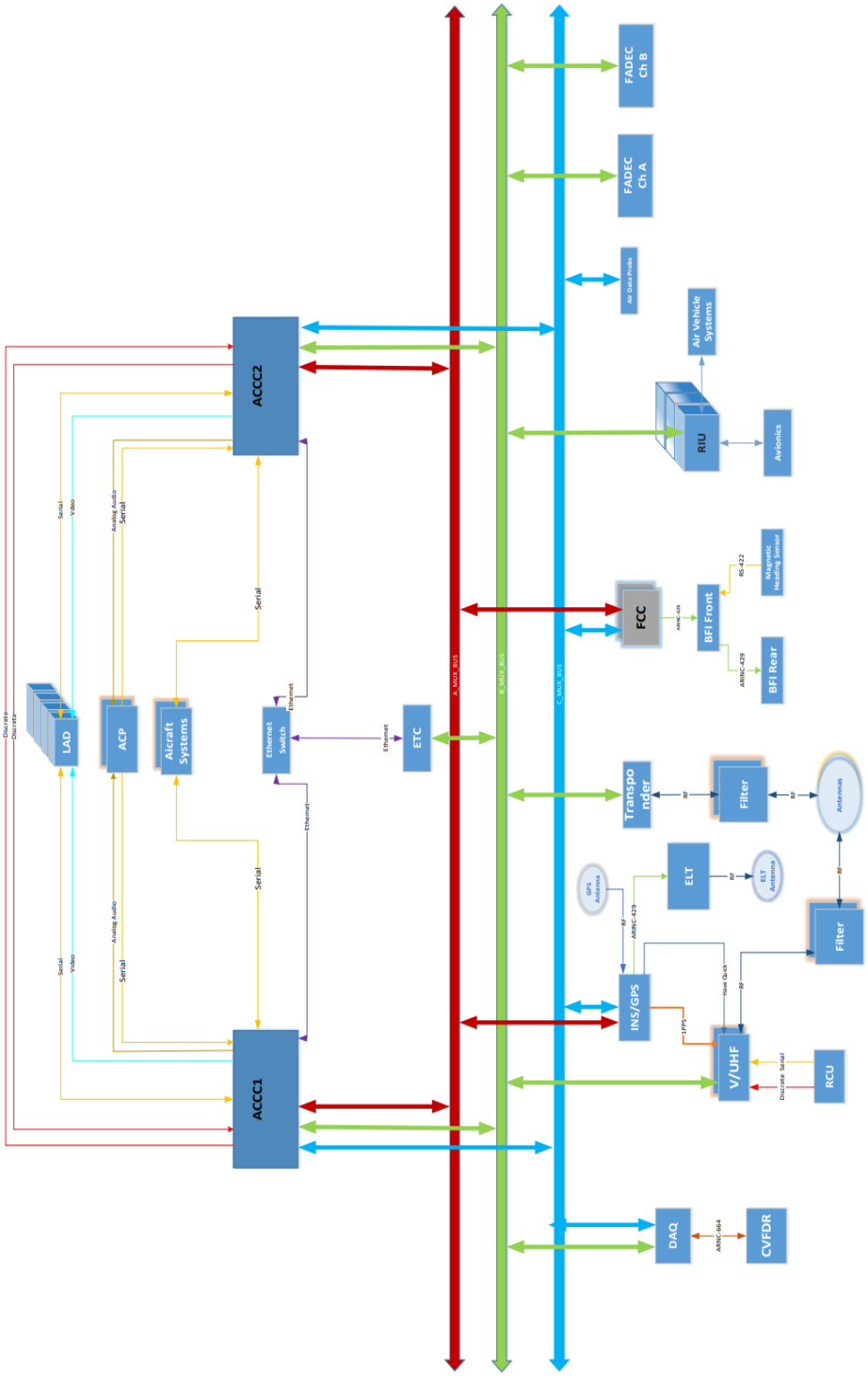


Figure 3-8 Aircraft MIL-STD 1553 architecture

There are other network elements in the bus as bus cable, coupler, stub cable, and terminator.

Table 3.6 Network Equipment

Equipment	Definition
Bus Cable	The cable between the transformer and BC, RT, or Monitor. Recommended length between couplers is >45cm. No maximum length is specified.
Coupler	Transformer interconnect from the bus to the stub of a BC, RT, or Monitor.
Stub Cable	The cable between the transformer and BC, RT, or Monitor. The maximum length is 6 meters for a transformer-coupled bus. Recommended lengths are 45cm to 550cm
Terminator	78 Ohm End Cap at the end of the bus cable or the last coupler on both sides of the bus. Terminators should not be put on stub connections.

3.6 Failure Management in Safety Critical Systems

To increase the reliability of a safety critical system there should be fault prevention and fault-tolerant solutions. To prevent faults design should be implemented as fault free. Also maintenance activities should be planned. At the same time systems or hardware should be replaced to prevent future probable failures. Another mechanism is to design the system as redundant and define the actions which can mask the failure. The next section defines the fault tolerance mechanisms and error detection methods. Also, other databus protocols will be analyzed in terms of error management.

There are various protocols in use in aviation in addition to MIL-STD 1553 such as TTP and AFDX. These protocols are used in safety-critical applications such as aircraft for avionics integration.

Databus communication can be triggered by a predefined schedule or event-driven. TTP and MIL-STD 1553 assign all messages to time slots to transmit messages in a predefined bus scheduler. The schedule is cyclic. However, if there are aperiodic messages these messages should be designed taking their importance and mode of operation, during takeoff aperiodic emergency messages should be sent immediately, into consideration. Avionics Full Duplex Switched Ethernet (AFDX) is a standard defining data communication using IEEE 802.3. It has additional deterministic features in terms of timing and redundancy management. Redundant connections, ports, and switches performing packet delivery, traffic policing, and filtering are used in a network. Time-triggered protocol (TTP) is a real-time communication protocol also used in aviation. It is redundant using replicated two buses. The bus has time synchronization for fault-tolerant communication. Node faults can be isolated and faulty nodes can not inject data to the bus in TTP.

Table 3.7 Communication Protocols

	MIL-STD 1553	**AFDX**	**TTP**
Access Method	Periodic and aperiodic Event driven	CSMA/CD	TDMA
Bus	Bus A and B	Network selector configured over virtual links	Bus A and B
Redundancy Management	Decided during design	Decided during design. End system decision.	Fault tolerance implemented by design.
Bandwidth	1 Mbit/s	100 Mbit/s	5-25 Mbit/s

3.7 Error Management

MIL-STD 1553 is a powerful databus protocol providing flight safety and achieving critical missions. There can be various faults during data transmission or on the RT, BC, or avionics connected to the data bus. MIL-STD-1553 data bus has efficient error and failure management. Command, status, and data words shall be correct and free from error. All the devices on the bus can report errors but only the bus controller decides if there is any error and provides correction actions. In system design, these errors should be analyzed and determined by the necessary actions identified by the bus controller or remote terminal. The remote terminal failures are defined using status words. If there is an error detected by RT, its status word is not activated. RT sends its status and sets its message error bit when BC transmits the mode code including the last status command. Message error bit is activated if there is a data word error, a gap between data words, an unknown command, or a wrong number of data words.

The bus controller decides if data is corrupted or faulty using bit error, word count error, manchester waveform error, or status word. Actions can be different in BC that is decided during designing the bus. Usually bus controller tries to send and receive data using channel B of the bus since the bus is dual redundant channels to provide redundancy.

Table 3.8 Error Determination Approach [20]

Error Identification	Failure Classes	
	Bus System	**Sensor**
Message error	Transmission from the bus controller to a terminal is decoded with an error condition by a receiving remote terminal.	-
Busy	Remote terminal unable to transmit Remote terminal	The remote terminal and sensor are unable to send or receive data.
Subsystem flag	—	Sensor failure preventing proper sensor actions
Terminal flag	Remote terminal failure prevents complete action by a terminal.	The sensor fails preventing complete action by a terminal.
Parity error (incorrect odd parity)	The error set in the status word	Data not usable by a sensor.
Improper sync	Unknown problem	Ignore all data stored from message continue to look for a valid sync
Invalid Manchester	Error in message	Ignore all data stored from message
Improper number of data bits and parity	Error in message	Ignore all data stored from message
Discontinuity of data words	Error in message	Ignore all data stored from message
No status word response	Unknown problem	Further investigation

If there is any data error, various error correction techniques are developed to determine the controller behavior in terms of a fault as defined in Table 3.9. These

behaviors define a set of actions that try to send the message or inform the controller to decide accordingly. In general, actions are taken at the system level considering the system functionality criticality.

Table 3.9 Error Correction Technique [20]

Error Identification Types	**Error Correction Technique**
Bus system failures No status word response Message error Parity error Invalid Manchester Improper number of data bits and parity Discontinuity of data words	Retry message on same alternate bus n times. Transmit reset remote terminal mode code if retry fails.
Busy	Retry message on the same bus after a fixed delay time.
Terminal flag	If necessary, transmit initiate self-test mode code. Transmit BIT mode code. Analyze failure and determine corrective action, which may involve the following mode code commands: Shut down the transmitter (00100 or 10100) Inhibit terminal flag bit (00110)
Improper sync	Transmit reset remote terminal mode code of retry fails. Ignore and reset for a valid sync
Subsystem flag	Normal data communication messages (address/subaddress) to examine sensor BIT discrete or words.
Sensor failure	Analyze failure and determine system-oriented corrective action

Aircraft operational flight program (OFP) of the aircraft control computer has a MIL-STD 1553 error mechanism as defined in Figure 3.8. When there are error flags defined by the 1553 card driver software, or message error defined in the status word or no response from the RT, OFP switches to bus B in the channel and tries to communicate in the same scenario but if there are four consecutive failures

after the each OFP processing cycle in the communication, OFP indicates a communication error to the pilot via caution panel. These actions only depend on the reliability and fault tolerance properties defined above. But there is a need to analyze security problems to realize unauthorized attempts in the system. These attempts focus on cyber attacks on defense platforms. However, the communication is time-based, and detecting anomalies should work on the data.

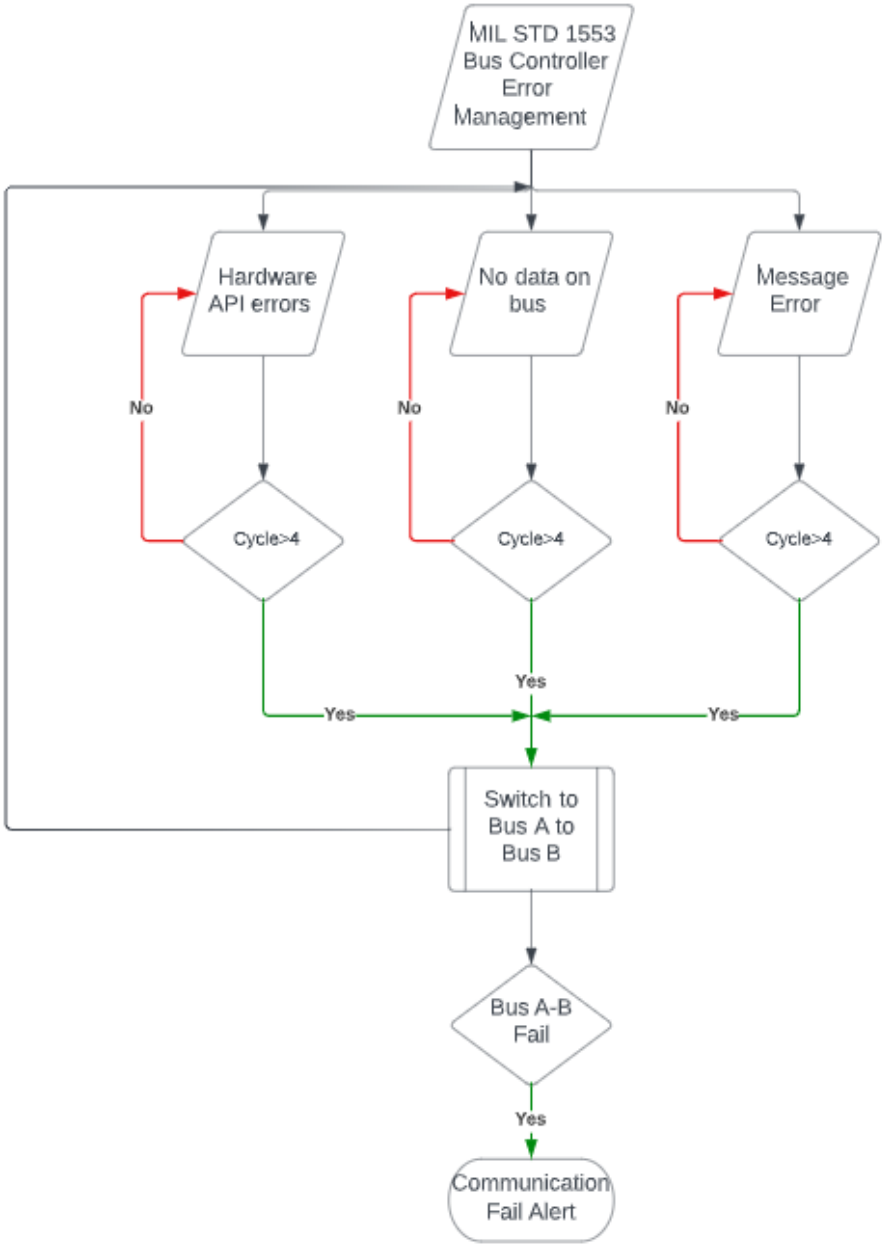


Figure 3-9 Aircraft MIL-STD 1553 error management

As discussed the error management mechanism, these methods do not provide a solution for cyber security problems. Also, there is not any security mechanism to use any authentication or encryption property. That means that when malicious software and hardware are connected on the bus, they can send data on the bus.

The bus itself is prone to unauthorized attacks when any malicious hardware or software is connected to the bus. The bus can be analyzed and any data can be monitored or injected to fail the bus scheduling or to obtain military data. Attack scenarios are defined in Section 5.1.

3.8 Machine Learning Models

Machine learning is the process of using historical data to create a prediction for future data [25]. Monitoring the MIL-STD 1553 bus is very effective for anomaly detection since the bus has various properties for determinism. The experimental approach for the datasets obtained from the real and simulated bus integration environment provides explicit results using the algorithms defined in the thesis. The basic steps in the machine learning approaches are data collection, feature selection, model selection, and producing the training and test dataset. The model can be able to evaluated and monitored based on the dataset.

Machine learning techniques can be used for anomaly detection. They can be categorized as supervised, unsupervised, and reinforcement learning [18]. In supervised learning, machine learning methods using classification methods are used. In unsupervised learning, a training dataset is used for learning and the objects of similar features can be able to grouped.

In the dataset, all the data are not needed and do not make sense to use. It is important to select the minimum amount of necessary data to feed to the training model. Feature selection is used to determine and filter important features. The classification of the data can be done using the various models.

Table 3.10 Learning models

Supervised learning	Unsupervised learning
Uses labeled data	Uses unlabeled data
Tries to predict something	Tries to group things based on their proper
Measure the accuracy	Evaluation is normally indirect
Underlying techniques classification, regression	Underlying techniques clustering

Regarding the anomaly detection for MIL-STD 1553, features are selected as RT address as dataset feature and dependent variable. The confusion matrix is prepared according to the method. With the supervised learning methods the dependent variable is known to be predicted can be known and in unsupervised methods the patterns should be searched for the dependent variable.

The accuracy rate is defined by the correct predictions in total predictions. Homogeneity is the clusters containing class members and completeness means members of the class are assigned to the same cluster can be the main metric to evaluate the clustering results.

There are criteria to achieve successful model selection.

The model should be trained in a reasonable time. Logistic regression and SVM can be trained quickly. Logistics regression and decision trees can analyze large amounts of data. Decision boundaries should provide good classification. The output of the model should be explainable. The decision tree is a good model for this.

3.8.1 Classification models

3.8.1.1 K-NN

K-NN is a simple machine learning algorithm such that features are stored and predictions are achieved using these stored features. A dataset is divided into train

and test sets for this supervised machine learning method. The model is trained and the dataset is tested accordingly. The training test split method separates the tests. Data is defined for train and test sets, separation is chosen randomly for the test set. Distances are computed between test points and trained labels using Euclidean or Manhattan distance. The shortest distance between 2 data points, x and y, is the Euclidean distance. Mathematically, Pythagoras' theorem is used to calculate the Euclidean distance. The sum of squares of the difference equals the squared distance between two points as defined in the following formula. The k value is chosen as the accuracy score of k. The k value defines the number of the nearest neighbors. Since there is no assumption about data in the algorithm, it is useful for non-linear data. It can be used for classification as well as regression. The result's accuracy is high.

Prediction is slow since it stores all the training data. It is very sensitive to the scale of data as well as irrelevant features. KNN is used for classification and prediction problems in industry.

$$Euclidian\ distance = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

3.8.1.2 Naive Bayes

Naïve Bayes using statistical assumptions and features can be chosen independently. This algorithm can predict the malicious RT with 100 % accuracy. It can be used for malicious RT detection since the algorithm can manage a large number of features providing fast predictions.

The algorithm for this method is as follows. Data is imported, train and test sets are defined. Naive Bayes classifier is built. The result is predicted based on the classifier. Accuracy of the model as a percentage is defined. As an option the results of the trained and test set are also visualized.

This method is a probabilistic method achieving the classification. It can be defined as the following formula P(B|A) representing B occurs if A is true. P(A) is the probability of A occurring and P(B) is the probability of B occurring. Using this formula Naive Bayes model determines labels for a set of features.

$$P(A|B) = \frac{P(B|A).P(A)}{\mathrm{P(B)}}$$

Multinomial, Bernoulli and Gaussian classifiers can be used. Gaussian anomaly detection is an effective classifier of Naïve Bayes model to find out anomalies in continuous data. For the dataset, the anomaly is found considering the Gaussian anomaly density function of the Gaussian distribution. The best threshold is found in estimating the parameters of the Gaussian distribution using the data representing a probability distribution of random variables where σ^2 is the variance, μ is the mean. The location of data is μ and σ is the data's scale. The mean converges to the normal value as the number of observations increases. Continuous variable x in the following formula keeps changes but mean and variance are constant for each distributed curve. The probability density for Gaussion distribution function can be defined as following.

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\Pi\sigma^2}} e^{-\frac{(\mathrm{x}-\mu)^2}{2\sigma^2}}$$

3.8.1.3 SVM

SVM divides the classes by finding a line between two classes. It defines the best hyperplane that separates the classes of data. The line between these points is called the support vector. If dividing classes by a straight line is not possible non-linear hyperplanes such as kernels can be used for prediction. Various kernels are special functions such as Gaussian, sigmoid or polynomial kernels. Complexity grows if the training samples are increased. Choosing the right SVM kernel can be difficult to decide.

3.8.1.4 Logistic Regression

Logistic regression is an algorithm that can be trained for various features. Logistic regression tries to model accurate probability estimates by analyzing the input variables. This causes to provide accurate classification. It can be used to predict if data is malicious or not using a binary outcome because it can manage large datasets. Logistic regression can be used for cyber security problems, such as attack detection. The logistic function which is called a sigmoid function returns values between 0 and 1 for the dependent variable.

$$f(x) = \frac{1}{1 + e^{-x}}$$

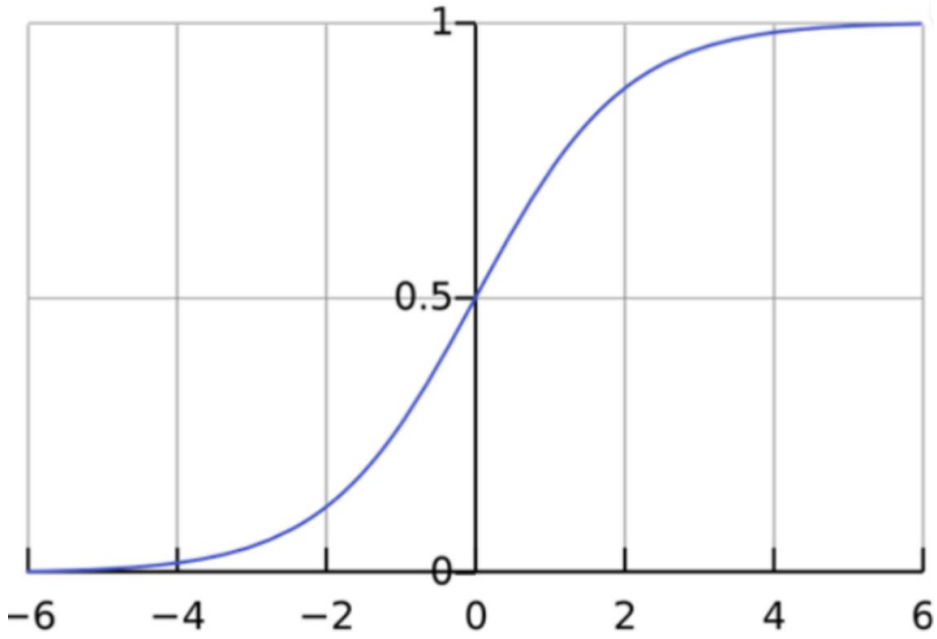


Figure 3-10 Sigmoid function

3.8.1.5 Decision Tree

A decision tree is a tree structure to achieve a decision. It can be used to predict categorical or numerical values without any normalization. Every prediction can be

expressed as a Boolean condition tracing a path from the root node to the leaf node. It performs well with large datasets. To decide the decision node entropy that is the amount of uncertainty in the dataset is used.

$$E(S) = -p_+ \log p_+ - p_- \log p_-$$

p+ is the probability of a positive class, p– is the probability of a negative class, and S is the subset of the training example.

3.9 Attack Scenarios

Cyber security issues make the systems implementing MIL-STD 1553 vulnerable to modern threats such as denial-of-service (DoS), dead time in the bus, spoofing attacks and susceptible to attacks. Bus controller functions can be emulated or bus controller components can be controlled on the bus. One of the bus controller functions is simulated or emulated data on the bus can be started or replaced.

Message spoofing can be done by the broadcast mode command or messages can be sent on the bus disrupting the messaging between remote terminals on the empty communication slots. DoS attacks can be achieved to prevent communication or try to achieve collisions. It is done by sending the data words which is an answer to the command words according to bus schedule. In detail, malicious devices send periodically the same command word for specific RT. A malicious remote terminal can act as a member of the bus remote terminal to change the value of a request to generate false output or stop communication.

However, once the words are on the bus, it is not easy to change those words since there is a very small inter-message gap time. But malicious data on the bus causes collisions and MIL-STD 1553 is very vulnerable to these kinds of attacks.

Data leakage is the transfer of data from a remote terminal to an external component by changing the terminal address or by increasing word count

increasing data. To achieve attacks message injection can be applied on the bus to block communication.

3.10 Features and Dataset

Considering attack scenarios, the detection of the anomaly considering the bus structure can be achieved by machine learning algorithms providing feature detection and analyzing which feature to use for detection. Also, it enables larger volumes of data to be analyzed. The attack scenarios can be various and the hidden pattern or malicious data on the multiple busses and multiple messages MIL-STD 1553 environment cannot be manually detected. Bus records as train sets for machine learning algorithms can be used to develop efficient models. These records need to be updated and machine learning models should be revised to increase the performance of the detection. One approach for anomaly detection is using detection software and hardware as passive and offline without an active role in the databus. Taking the records and analyzing the outcome and reporting if there is malicious integrated software or hardware in the system. Another approach is using the detection software and hardware as a bus member and alerting the pilot or maintenance personnel to take action.

In a typical operation of the 1553 system, a malicious remote terminal can act as a member of the bus remote terminal to change the value of a request to generate false output or stop communication. Since this standard is implemented in mission-critical systems, it was designed to provide a high level of fault tolerance. To meet the security communication requirements for MIL-STD 1553, Intrusion Detection methods can be used for the bus to prevent threats. and identify anomalous data transmissions.

Classifying a dataset including a malicious message which is is 1 % of all samples hardens the classifier performance. This is an unbalanced dataset causing the classifier to perform poorly. Scaling can be used to standardize data by giving it a

mean of 0 and a deviation of 1, resulting in a normal distribution to increase classification performance. Random Undersampling can be used to match the number of anomalies to balance the imbalanced dataset. Random oversampling also helps to have a balanced dataset. Different performance metrics defined in Section 3.12 should be used. Feature selection is used to increase model performance. Highly correlated features are combined to solve feature correlation. Logistics regression, SVMs, and decision trees use methods to determine feature selection. Manual techniques can be used to leave out features and observe the output.

3.11 Simulation Software

A simulation should provide a realistic integration environment. The test should be conducted using LRU or its simulator/emulator model interacting with OFP. Simulation software tools should work together with commercial off-the-shelf software tools. Simulation software should be able to change, monitor, compare, wait until a defined time, data on the database, use system time, provide warning, able to manage submode and replay the recorded data. To simulate flight conditions flight simulators should be integrated with the test design. The Avionics data bus is monitored to verify the data transfer. Test points for the selected signals help to understand integration problems. Data to be monitored should be converted to engineering data via configuration files. Various interfaces such as MIL-STD 1553, ARINC-429, ARINC-708, ethernet, RS-422, RS-232, RS-485, discrete, analog, synchro, TTL, CAN bus are used in avionics architecture.

It is important to model the simulator in a fast and flexible way. Automation is needed to accelerate the test such as generating test cases and test steps automatically from the model and generating the documents.

Simulator development with ATLAS consists of several steps. The message structures (static requirements) described in detail in the software requirements

have been expanded with behavioral models (dynamic requirements). These requirements become the input of ATLAS for use in the phase. Static requirements are modeled with the help of the generic user interface as a result message structures are formed, then the dynamic models are developed by coding.

The code generator provided by ATLAS parses the simulator models and generates code according to language selection. A binary library file (.dll) is created by combining the implementation codes of dynamic requirements. This file becomes the input of ATLAS Test Suite as a simulator. The steps taken up to this stage have been made to create the simulator, it is followed once as long as the requirements are not changed.

ATLAS Test Suite creates a configuration by taking one or more of the simulators mentioned in the previous steps. An automatic user interface has been generated from modeled message structures also each different message type for example enumeration and geolocation, are presented with different layouts. All process is explained in Figure 3-11.

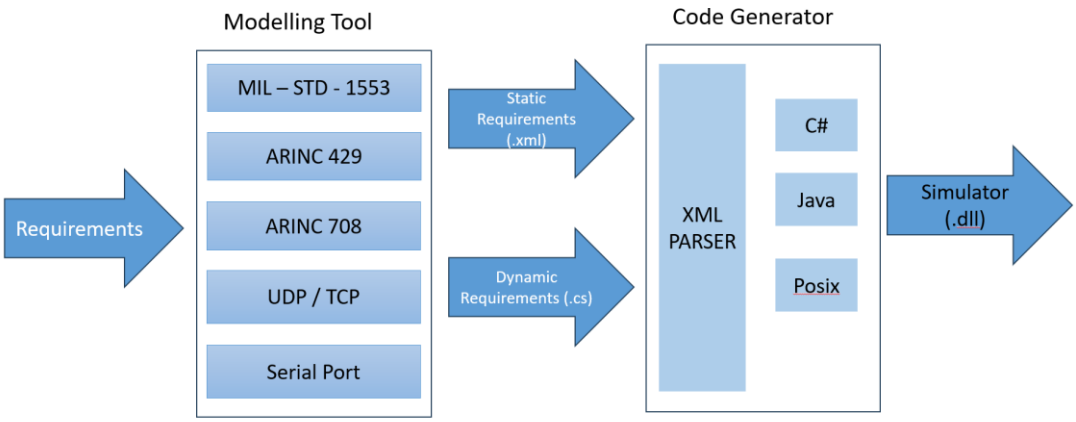


Figure 3-11 Simulator Life Cycle

The simulator software characteristics are as follows:

• It consists of avionics models based on LRUs. These models provide LRU functional simulation as well as ICD messages for testing purposes.

• Monitor selected avionics signals. Interface with the MIL-STD 1553, ARINC429, Serial, and Discrete HWCIs and initialization of these HWCIs.

Simulator architecture is composed of presentation business, hardware access and common layers. The modeling tool provides a script development structure for non-modelable objects. The test engine displays the test reports by communicating with the simulator [17].

Table 3.11 Simulation Software Functions

Software Function	Definition
Simulation Control	Activate flight model and simulation
Data entry	Entry panel, graphical user interface
Data display	Multi-Function Display, Head Up Display, Control Panel, Master Warning Panel
Data Analysis	Data monitor, record, replay, report
Hardware-Software Interface	Simulation computer, hardware driver, RS-422, analog, discrete, MIL-STD 1553, Arinc 429, Arinc 708 adapter software
Models	Static/Dynamic models, avionics
Model Control	Failure activation, model ON/OFF control
Script preparation	Simulation automaton
Data conversion	Engineering unit conversion

3.12 Classifier Performance

When the dataset is examined for the predictions, every predicted data belongs into four categories:

- True Positive TP: positive
- True negative TN: negative
- False Positive FP: negative mistakenly classified as positive

- False Negative FN: positive mistakenly classified as negative

Accuracy is the ratio of correctly classified data points to total data. The detection rate is the number of attack samples detected by the model to the total number of attack samples in the data. They are visualized as a confusion matrix.

Model accuracy $= \frac{\text{TN+TP}}{\text{TN+TP+FP+FN}}$

Error rate= 1 -Accuracy

Precision = $\frac{FP}{FP+TN}$

Sensivity or True Positive Rate = $\frac{TP}{TP+FN}$

False Positive Rate = $\frac{FP}{FP+TN}$

Recall =$\frac{TP}{TP+FN}$

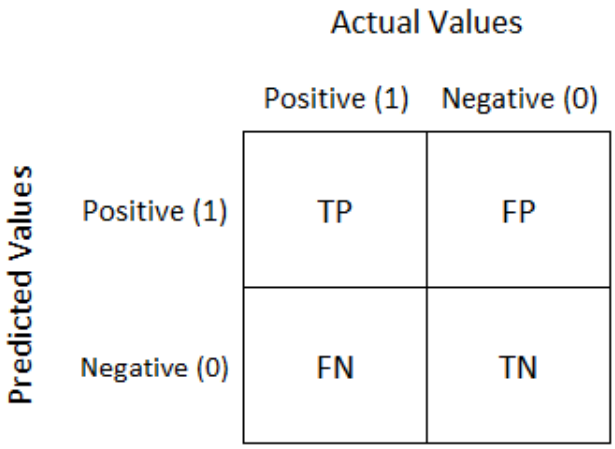


Figure 3-12 Confusion Matrix

Accuracy can be the criteria for performance evaluation in traditional classification models. However, there are other metrics such as precision, recall, F-measure and ROC area defined for a single-label multi-class classification problems [29]. The prediction can be correct or partially correct or incorrect for a collection of labels.

Multiclass classification is the classification with more than two classes. Each sample is labeled as one class. One-vs-the-rest (OvR) multiclass strategy is implemented in to fit one classifier per class.

Classification reports define recall as the ratio of true positives to the sum of true positives and false negatives, F1 score. The F1 is the weighted harmonic mean of precision and recall. The closer the value of the F1 score is to 1.0, the better the expected performance. If there is an imbalance in the dataset, F1 score is chosen since high F1 score defines both precision and recall. High F1 means false positives and false negatives are low.

The supervised classification algorithm outputs a score. There is a need to set a threshold to select the related class that has the highest score. The highest score can be defined according to the false positive to false negative ratio or precision or recall target. Receiver operation characteristic (ROC) plots and area under the curve (AUC) computation are done with FPR on the x-axis and TPR on the y-axis.

There is not any labeling in clustering, features are observed and clusters are created. The performance of a clustering algorithm is different from supervised learning such as the precision and recall. Clusters are evaluated based on the distance between points. The evaluation metric for clustering algorithms is the Silhouette coefficient which is defined as the following formula where a is the mean distance between the sample and other points in the same cluster and b is the mean distance between the sample and other points in the next nearest cluster. The score is between -1 and +1. If the clusters are dense it is higher meaning well separated.

$$s = \frac{b - a}{\max(a, b)}$$

3.13 Tools for Machine Learning

There are several software tools for Machine Learning available in the market including Google COLAB, Saturn Cloud including Jupyter and RStudio, Amazon SageMaker, Microsoft Azure, VS Code for various algorithms and features such as Sci-kit Learn, Pytorch, Tensorflow, Weka, Knime, Apache Mahout, Accors.net, Shoguni Keras and Rapid Miner. In this thesis, Google Colab is used. It is a cloud service that supports Python and google drive can be integrated. Python has various libraries to support machine learning algorithms. Pandas, matplotlib, numpy array processing library, sciPy scientific computing, scikit-learn contains supervised and unsupervised ML algorithms and tensorFlow as well. Microsoft.NET provides to add machine learning algorithms to NET applications. Using this capability data on the MIL-STD 1553 bus can be analyzed to make predictions. Classification, categorization, regression, prediction, anomaly detection and recommendations can be achieved using these algorithms ML.NET includes data loading and preparation providing transformations. Training algorithms such as binary classification, multiclass classification, anomaly detection, ranking, regression, and recommendations algorithms. These algorithms are used to create model operations.

CHAPTER 4

SIMULATION ENVIRONMENT

The test set-up has various hardware and software components. Hardware components included in the setup are defined in Table 4.2. ATLAS simulator is used to create the messages of avionics on the set up to have a realistic scenario since some functionality can not be activated if the aircraft is not in the air. The message list in the databus is listed in Table 4.1 for attack scenarios 1, 2 and 3. The bus architecture is defined in Figure 3.8.

The columns of Table 4.1 are defined as follows:

R/T Add: Address of the remote terminal

SA: Subaddress of the remote terminal

LRU: Line replaceable unit

Message name and description : Name of the message and its description on the bus.

T/R: Transmit or receive message.

Word count: The number of words for messaging

CW: Command word of bus controller

SW: Status word of remote terminal

DW: Data words sent by bus controller or remote terminal

Int Msg gap: Time for next message word as defined in Figure 4-13

Response Time: Time for status word defined in Figure 4-13

Frequency: Message frequency

Table 4.1 Dataset Message List

LRU	Message Name	Description	T/R	Word Count	CW	SW	DWs	Int_Msg Gap	Resp. Time	Total (uSec)	Freq. (Hz)	SA	R/T Add
EQ1	01R	Configuration Loading	R	15	20	20	300	50	10	400	1	1	16
EQ2	02R	Position and Time Information	R	11	20	20	220	50	10	320	16	2	16
EQ3	EQ3*01	Preset Bilgileri	T	32	20	20	640	50	10	740	1	1	9
EQ3	EQ3*03	Cihaz İçi Test	T	10	20	20	200	50	10	300	1	3	9
EQ3	EQ3*05	Telsiz Ayarları	T	11	20	20	220	50	10	320	1	5	9
FADEC A/B	T12	BACKUP ENGINE STATUS/DATA	T	20	20	20	400	50	10	500	16	12	9
FADEC A/B	T13	ENGINE DATA	T	20	20	20	400	50	10	500	16	13	9
CVFDR	SA1	Message 2 Hz	R	32	20	20	640	50	10	740	2	1	27
CVFDR	SA2	Message 2 Hz	R	32	20	20	640	50	10	740	2	2	27
RIU_1	RD_INFO	RD_INFO	T	23	20	20	460	50	10	560	1	1	20
RIU_1	ANALOGO	WR_ANALOGO	R	23	20	20	460	50	10	560	1	30	20
RIU_2	RUN_TIME_WRITE	RUN_TIME_WRITE	R	2	20	20	40	50	10	140	1	4	21
RIU_2	WR_DISCIO	WR_DISCIO	R	2	20	20	40	50	10	140	2	26	21

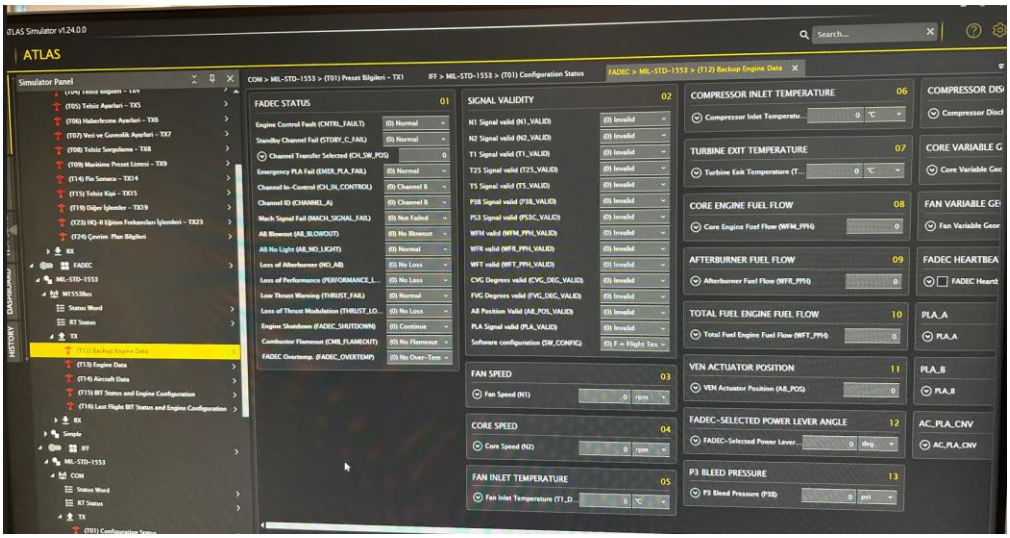


Figure 4-1 ATLAS simulator GUI

4.1 Simulation Setup Hardware

All equipment start communication on avionics data buses on power-up. Interfaces such as discrete, analog, synchro, TTL, and discrete are connected to PCIe to route the signals to the related LRU. Altadata PCIe 1553B interfaces the avionics bus to simulate avionics functionality.

Table 4.2 Simulation Hardware

Hardware Components	Definition
Simulation Computer	PC
Data bus tool	Commercial off the shelf products such as Ballard, Altaview
Equipment	Avionics
Monitoring	Bus tool, Oscilloscope
Power Supply	115 V AC, 220 V AC, 28 V DC power supply, circuit breaker

When a simulation is activated, equipment starts to receive data from the simulator.. Cabling is compatible with the aircraft environment. Various testbeds are designed to have a clear dataset for the data on the bus.

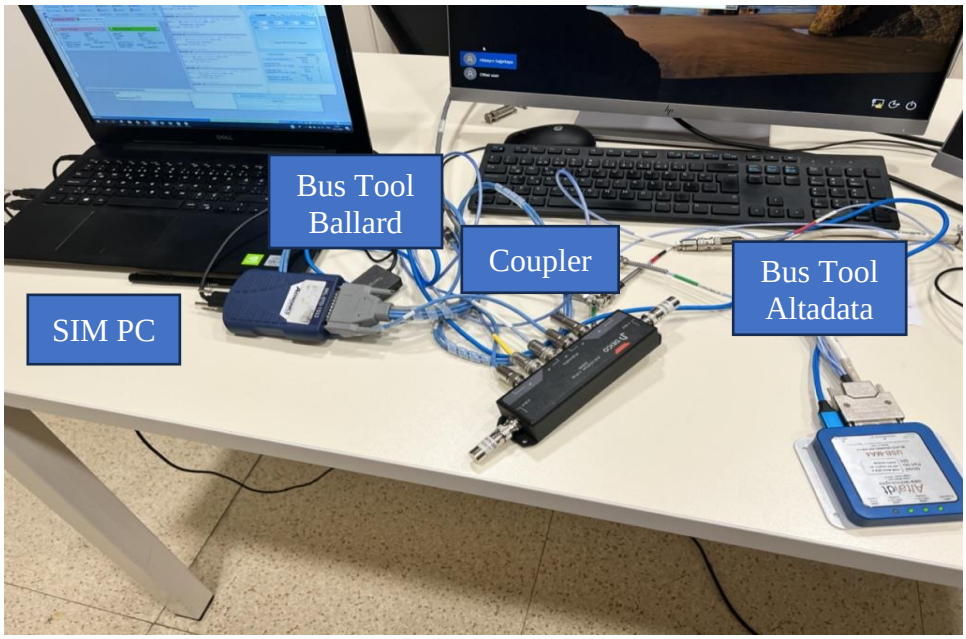


Figure 4-2 Test Setup for Dataset

The following figure is the environment for the MIL-STD 1553 architecture defined in Figure. 3-9.

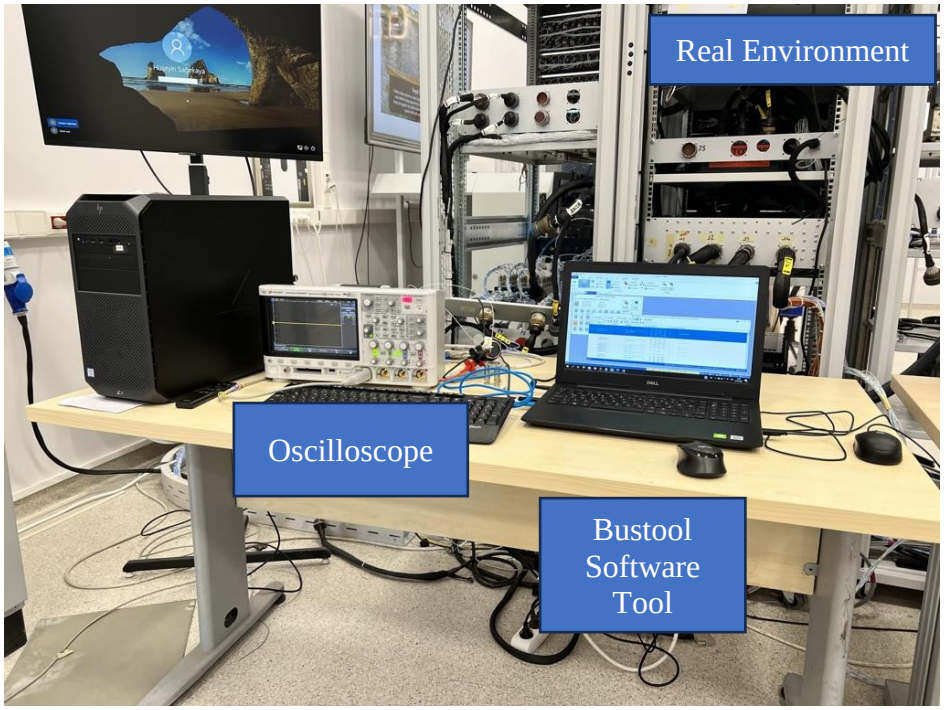


Figure 4-3 Test setup for real dataset recording

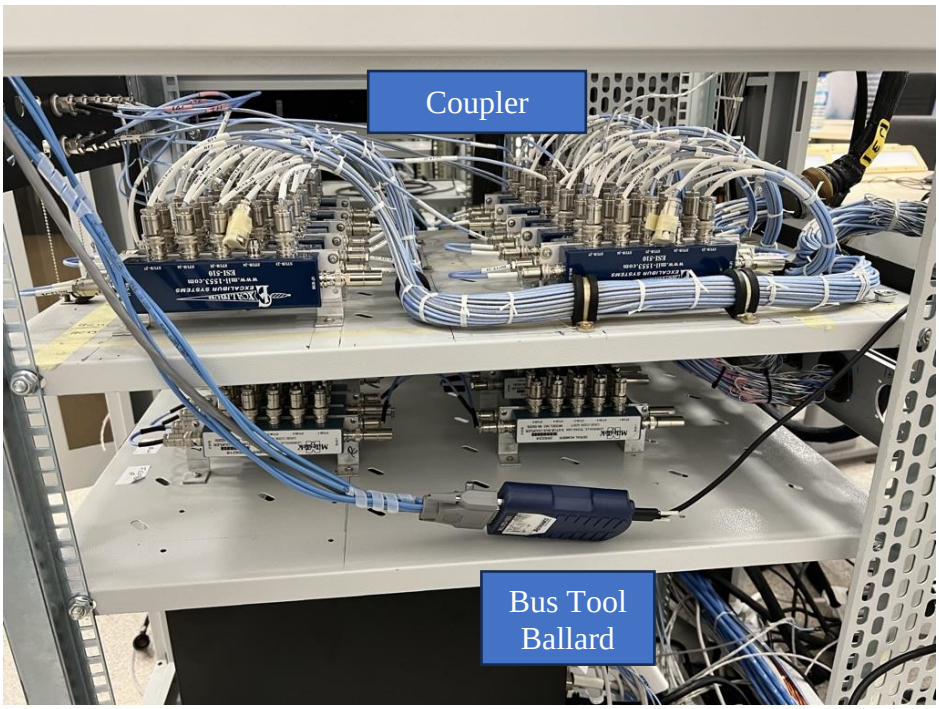


Figure 4-4 Integration of Bus consisting of bus couplers

Figure 4-5 Aircraft Control Computer- Bus Controller

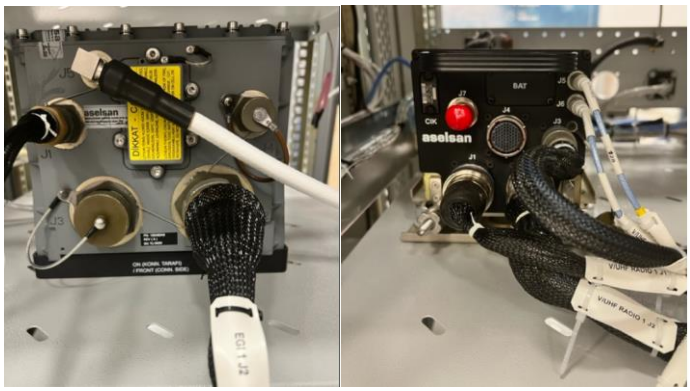


Figure 4-6 Avionics EGI and V/UHF radio – Remote Terminals

4.2 Driver Integration

In this thesis, a driver has been developed for interfacing with the API of the PCIe card which can provide a record in csv file format as a dataset. For a card to function and have an interface with the host computer's software, the driver that is specific to the card is required to be installed on the host computer.

API, Application Programming Interface, is an interface between two software programs. APIs are used as a standardized means to work together. Programs interface with each other using predefined commands. Altadata uses a layered structured API.

APIs are provided by the card manufacturer to have different levels of control over the card. To effectively use the equipment, the developer must comprehend the API functions and utilize them in the ATLAS code. The API documentation explains the mechanisms of the card and sometimes gives examples of the methods via code blocks written in several languages like C, C++, and C#.

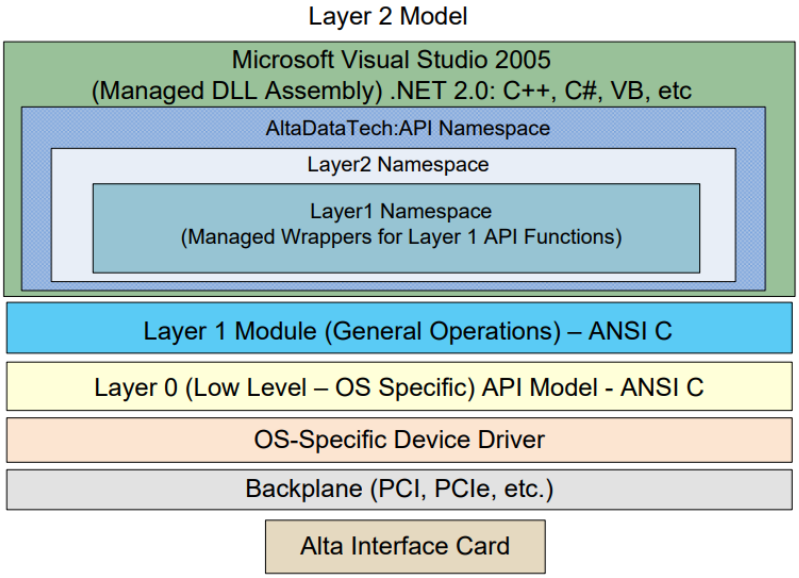


Figure 4-7 Altadata API Layers [30]

Layer 1 is the functional core of the API. This layer is written in ANSI C for portability to any environment and provides all functions needed to control the Alta hardware. Layer 2 uses the Layer 1 API on Layer 0 API to communicate with the Alta hardware. This layer uses higher-level programming languages to encapsulate the Layer 1 functions for object-oriented programming.

4.2.1.1 MIL-STD 1553 Implementation Flow

The implementation flow for a remote terminal is defined in Figure 4.8 below for Altadata MIL-STD 1553 multifunction (BC, RT, BM) card. The method names will not be the same for a different producer's card. However, the general logic would be similar.

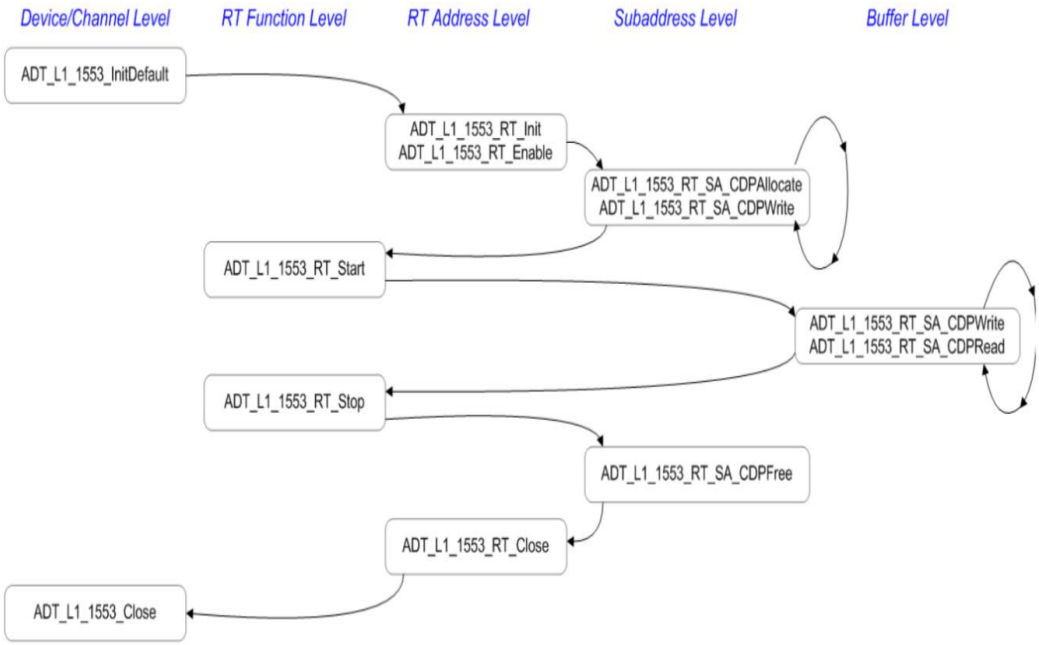


Figure 4-8 Basic Remote Terminal Programming Flow

If we enable the RT (ADT_L1_1553_RT_Enable) and start RT operation (ADT_L1_1553_RT_Start), the RT responds to commands and all sub-addresses would be “wrapped”. Accordingly, if the BC sends a BC-RT message with a given set of data, then sends a RT-BC command, it will get the same set of data back in the RT-BC message.

4.3 Dataset

In this thesis 3 attack scenarios are simulated. The transmitted messages and data are defined in the Appendix.

Various attack vectors have been simulated as follows.

1. Scenario 1: Spoofing Attack with same RT address; attacks imitate the legitimate RT. Here the malicious RT tries to force communication with BC.

   The spoofing attack is achieved by sending messages with a duplicate address of 20 to the bus in the real integration environment.

In this scenario the attack messages are generated by the simulator and transmitted on the bus using the AltaData USB MIL-STD 1553 converter as shown in Figure 4-9. BC and RTs are the real equipment of the aircraft.

2. Scenario 2: Spoofing Attack within dead (unused) bus time; attacks can use spoofing to behave as a normal message without disturbing normal communication. The bus is loaded with malicious messages during unused bus time. This scenario is achieved by a simulation computer simulating all the messaging in a real integration environment sending aperiodic messages.

   In this scenario the attack messages are generated by the simulator and transmitted on the bus using the AltaData USB MIL-STD 1553 converter as shown in Figure 4-9.

   In this scenario the attack messages are generated by the simulator and transmitted on the bus using the AltaData and Ballard USB MIL-STD 1553 converters. BC and RTs are simulated by ATLAS software.

3. Scenario 3: DOS attack; Bus is flooded with high volume malicious messages to achieve denial of service. Malicious remote terminal sends unexpected malicious messages to other remote terminal or bus controller causing failures. This causes collisions on the bus and causes incorrect operation of other RTs. This scenario is same with scenario 2 but malicious RT sends unauthorized messages periodically.

   In this scenario the attack messages are generated by the simulator and transmitted on the bus using the AltaData and Ballard USB MIL-STD 1553 converters. BC and RTs are simulated by ATLAS software.

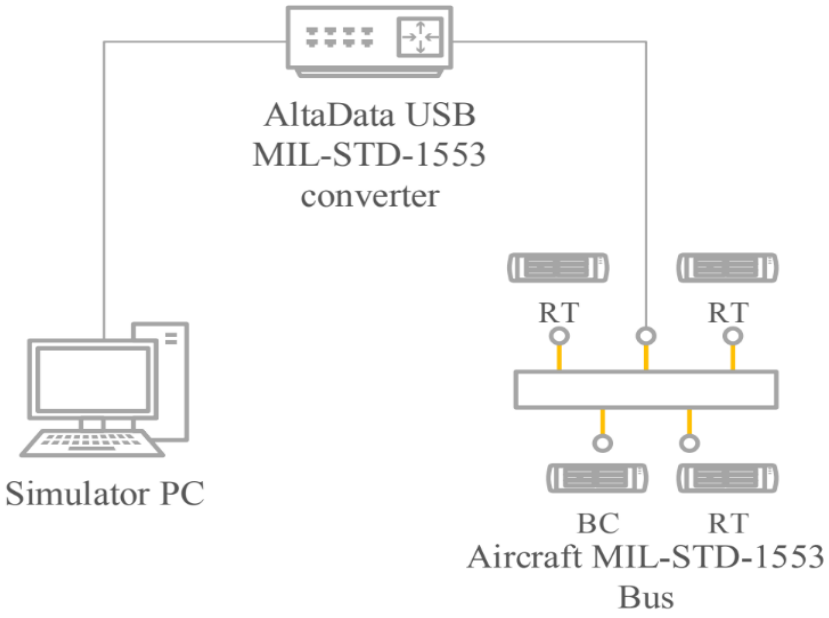


Figure 4-9 Test setup using bus-tools for scenario 1.

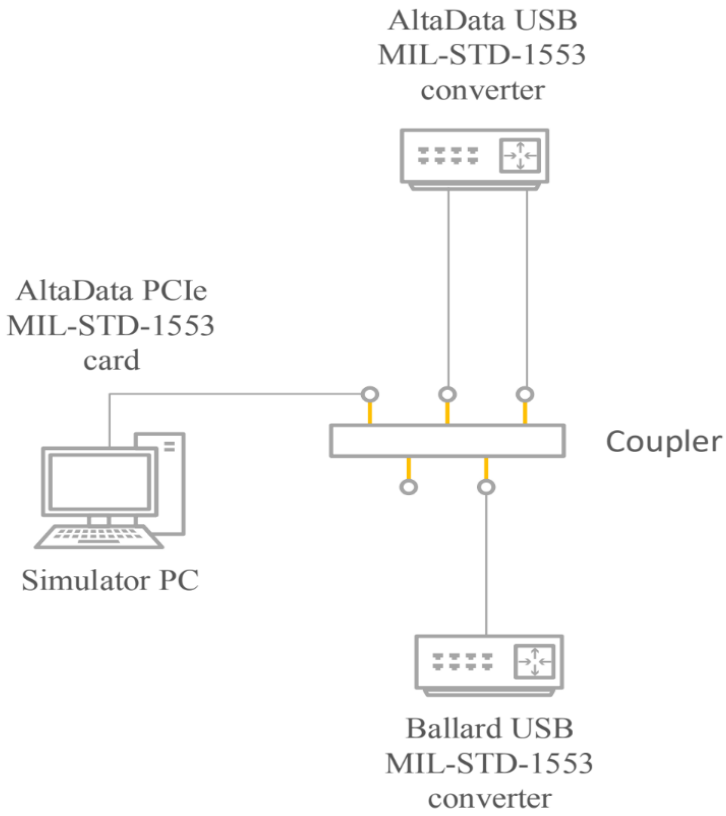


Figure 4-10 Test setup using bus-tools for scenario 2 and 3.

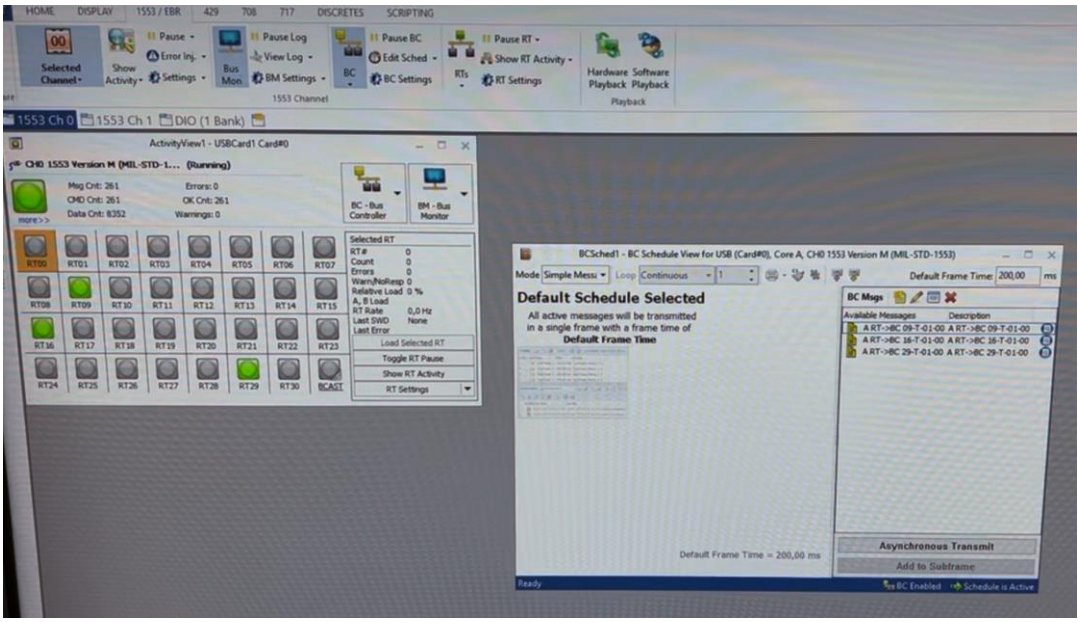


Figure 4-11 Scenario 3 attack scenario Ballard bustool view

RT is defined on the bus sending the 32-word data on subaddress 10. When communications start it is seen that the command word error is defined on the bus. This demonstrates the spoofing attack causing the error on the bus.

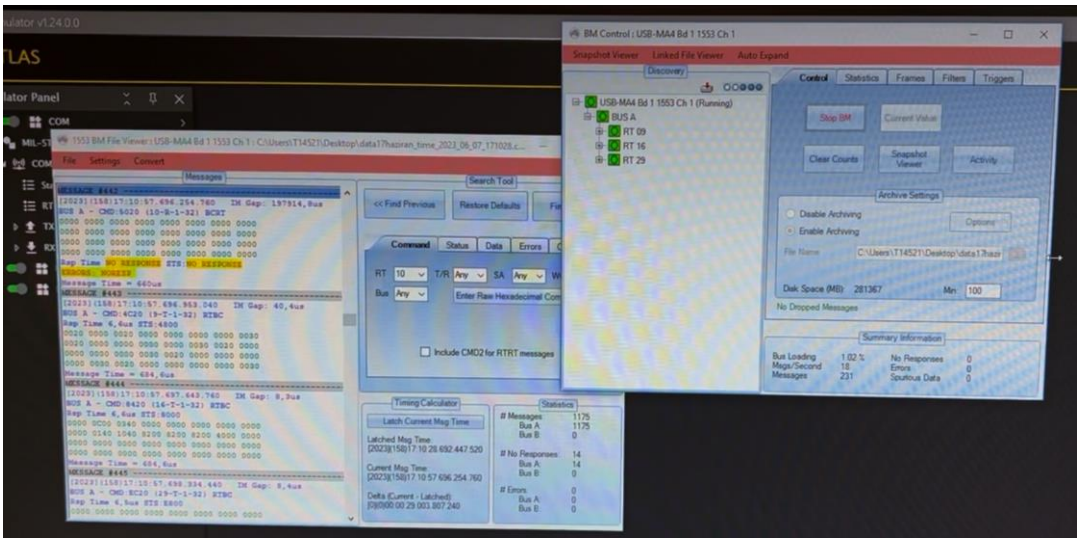


Figure 4-12 Scenario 3 attack scenario Altadata bustool view (RT10 unauthorized access – Scenario 2)

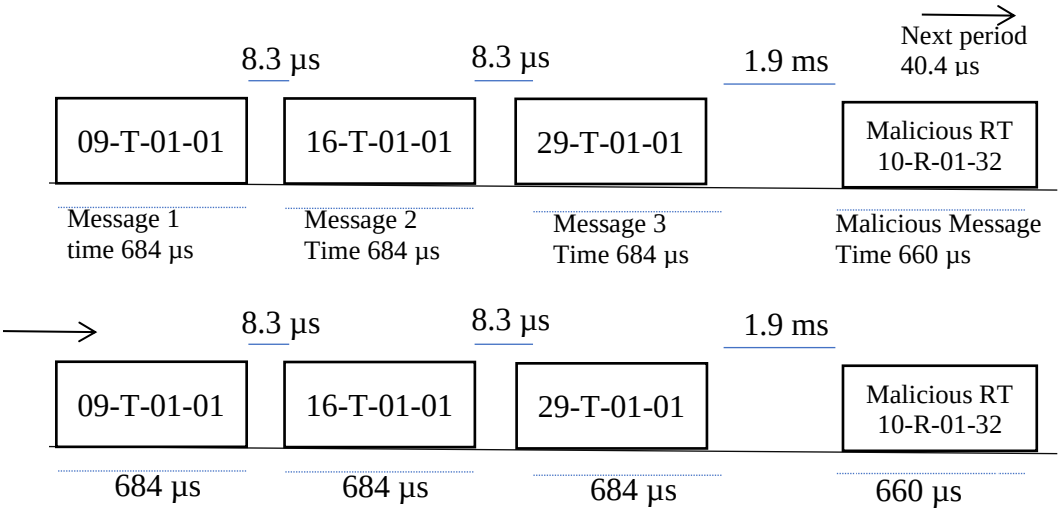


Figure 4-13 Major frame and malicious message for Scenario 2

Message time calculation can be defined as follows:

MIL-STD 1553 clock works with a timing of 1MHz on the bus, meaning one microsecond (µs) per bit, resulting in 20 bits per word. BCRT messaging consists of command and status words and also a maximum of 32 data words. The intermessage gap time is 4 µs and the response time for a status word is 4 µs. A 32-word message takes (34 words * 20 µs/word) + 4 µs, resulting in 684 µs to send the message as defined in Figure 4-12.

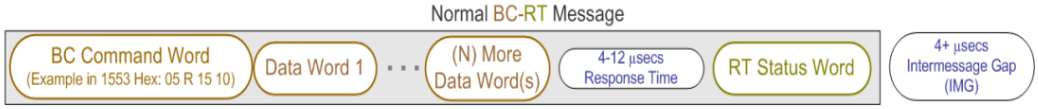


Figure 4-14 BC-RT Messaging [27]

4.4 Parsing

To evaluate the data which is active on the bus, the dataset is created using a parser implemented in C#. The parser takes the ASCII dump file which is the output of the Altadata bus tool and converts it to engineering data as seen in Table 4-5 as excel or csv format.

To design a parser bus tool format should be known. In the thesis, the Altadata bus tool is used and its word structure is defined. The parser code is given in Appendix. ASCII dump file contains status and command words in ASCII format. In the parser masking is used to extract specific values defined in the dataset and related data is isolated from the rest. For example to extract RT address message in a command word, a bitwise and operation with F800 hex is used to extract first five bits in a 16 bit command word.

Table 4.3 Parser output of ASCII Dump Data (Dead Bus Time).

Cmd Rt Addr	Cmd Transmit Receive	Cmd Sub Address	Cmd Word Count	Sts Rt Address	Sts Messge Error	Sts Inst	Sts Service Req	Sts Broadcst Commnd Received	Sts Busy	Sts Subsyt Flag	Sts Dynamic BusCntrol Acceptnce	Sts Termnal Flag	Is Error Data	Gap (ms)	Time
2	1	1	0	2	0	0	0	0	0	0	0	0	00,99313		[2022](318) 18:04:28.192.082
2	0	1	0	2	0	0	0	0	0	0	0	0	0	8,40E-05	[2022](318) 18:04:28.192.777
3	0	1	0	3	0	0	0	0	0	0	0	0	0	0,986178	[2022](318) 18:04:28.292.082
2	1	1	0	2	0	0	0	0	0	0	0	0	00,99313		[2022](318) 18:04:28.392.082
2	0	1	0	2	0	0	0	0	0	0	0	0	0	8,40E-05	[2022](318) 18:04:28.392.777
2	1	1	0	2	0	0	0	0	0	0	0	0	00,99313		[2022](318) 18:04:28.592.082
2	0	1	0	2	0	0	0	0	0	0	0	0	0	8,40E-05	[2022](318) 18:04:28.592.777
3	0	1	0	3	0	0	0	0	0	0	0	0	0	0,986178	[2022](318) 18:04:28.692.082
2	1	1	0	2	0	0	0	0	0	0	0	0	00,99313		[2022](318) 18:04:28.792.082
2	0	1	0	2	0	0	0	0	0	0	0	0	0	8,40E-05	[2022](318) 18:04:28.792.777

Table 4.4 ASCII dump data (Spoofing attack)

IMGap	CMD1	STS1	DTA 1	DTA 2	DTA 3	DTA 4	DTA 5	DATA06	DATA07	DATA08	DATA09	DATA10	DATA11	DATA12	DATA13	DATA14	⋯ DATA32	TimeStmp
0x 001E33C2	0x 4C20	0x 4800	0x 0020	0x 0000	0x 00200x0000		0x 0000	0x 0000	0x 0000	0x 0030	0x 0020	0x 0000	0x 0000	0x 0000	0x 0000	0x 0030	0x 0030	[2023] (158) 17:10:28. 692
0x 0000003F	0x8 420	0x 8000	0x 0000	0x 0C00	0x 0340	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0140	0x 1040	0x 8200	0x 8200	0x 8200	0x 0000	[2023] (158) 17:10:28. 693
0x 0000003F	0x EC20	0x E800	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	[2023] (158) 17:10:28. 693
0x 001E33C3	0x 4C20	0x 4800	0x 0020	0x 0000	0x 0020	0x 0000	0x 0000	0x 0000	0x 0000	0x 0030	0x 0020	0x 0000	0x 0000	0x 0000	0x 0000	0x 0030	0x 0030	[2023] (158) 17:10:28. 892
0x 0000003E	0x 8420	0x 8000	0x 0000	0x 0C00	0x 0340	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0140	0x 1040	0x 8200	0x 8200	0x 8200	0x 0000	[2023] (158) 17:10:28. 893
0x 0000003F	0x EC20	0x E800	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	0x 0000	[2023] (158) 17:10:28. 893

The dataset features are inter-message gap times, the number of data words, and the period that is the time between the bus controller and remote terminals. We present the correlation between the period and the data in the figures presented in the next section.

CHAPTER 5

ANOMALY DETECTION

In this thesis, we propose using machine learning algorithms to prevent threats and identify anomalous data transmissions such that the security communication requirements for MIL-STD 1553 are met. Supervised learning methods are used to predict future anomalies can be as in the training set of the machine learning method.

The dataset is prepared by monitoring the message traffic on the bus. The dataset is constructed for various RT in-flight configurations. Traffic on the test setup is recorded and each 2-minute stream is defined in the dataset. The dataset on the bus is defined with the properties below. Command and Status words are defined in the dataset since the effect of DoS attack on these words is analyzed.

- Timestamp
- Gap between the data
- Command Word RT Address
- Command Word RT SubAddress
- Command Word WordCount
- Status Word RT Address
- Status MessageError
- Status Word Instrumentation
- Status Word ServiceRequest
- Status Word BroadcastCommandReceived
- Status Word Busy
- Status Word SubsytemFlag
- Status Word DynamicBusControlAcceptance
- Status Word TerminalFlag
- Is Error Data

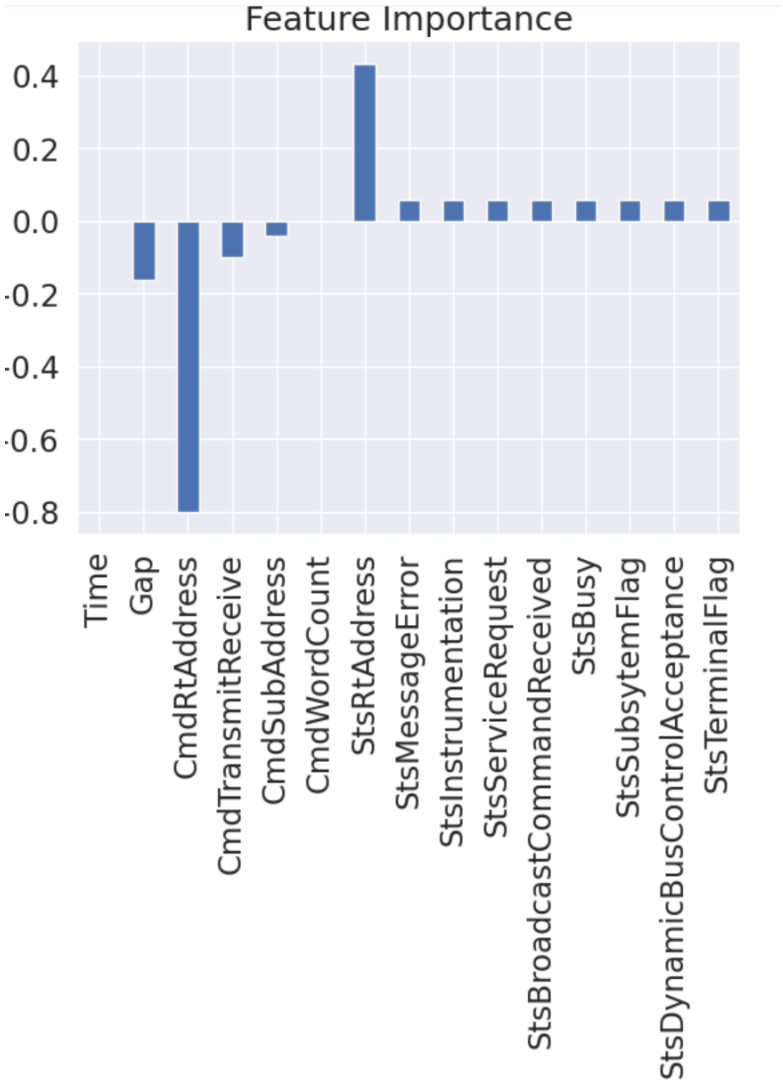


Figure 5-1 Dataset feature importance

To select the most suitable feature, the dataset needs to be analyzed. Feature selection directly impacts the performance of the machine learning model. Different feature selection methods can be implemented such as chi-square test, variance threshold and recursive feature elimination. Dataset feature importance is visualized in Figure 5-1. Chi-square test is a statistical test explaining the difference between the observed and the expected data. It can be used to find the correlation between variables in data. Chi-square test results two values. Feature dependency increases when chi-square value is larger or p-value is lower.

Gap	7.013578e-01
CmdRtAddress	1.335023e-01
CmdTransmitReceive	2.699796e-03
StsRtAddress	2.015412e-09
Time	1.602880e-84
StsMessageError	8.047395e-179
StsInstrumentation	8.047395e-179
StsServiceRequest	8.047395e-179
StsBroadcastCommandReceived	8.047395e-179
StsBusy	8.047395e-179
StsSubsytemFlag	8.047395e-179
StsDynamicBusControlAcceptance	8.047395e-179
CmdSubAddress	NaN
CmdWordCount	NaN

Figure 5-2 Scenario 3 dataset statistics- pvalues

Figure 5-2 above, it is seen that according to the chi-square test, features except CmdSubAddress and CmdSubAddress are dependent to label, error.

From the feature selection evaluation, it is decided that time-based properties and error flags are used for unsupervised and supervised learning methods To provide features for detection. Figure 5-2.

Table 5.1 Dataset statistics for Scenario 3

Dataset statistics for DOS attack	
Number of variables	16
Number of observations	2548
Total size in memory	318.6 Kib
Average record size in memory	128.1 B
Categorial variable types	16

Attack data has a feature attribute called IsErrorData Label defining the error in the dataset. This defines if there is an anomaly in the message. The label is described in Table 5-2.

Table 5.2 Scenario 3 Attack data characteristics

Is Error Data	Count	Frequency
0: No Error	1896	74.4 %
1: Error	652	25.6 %

All the features are highly correlated with each other.

Table 5.3 Dataset Statistics for Scenario 2

Dataset statistics for dead time attack	
Number of variables	16
Number of observations	1175
Total size in memory	147 Kib
Average record size in memory	128.1 B
Categorical variable types	15
Numeric	1

Categorical variable has two or more values that has nominal or ordinal variables. ML algorithms except decision tree are unable to operate using categorical data. Encoding techniques are used to canvert data to numeric.

Table 5.4 Scenario 2 Attack data characteristics

Is Error Data	Count	Frequency
0: No Error	1161	98.81 %
1: Error	14	1.19 %

Table 5.5 Dataset statistics for Scenario 1

Number of variables	16
Number of observations	747
Total size in memory	93.5 Kib
Average record size in memory	128.2 B
Categorial variable types	15

Table 5.5. (Cont'd)

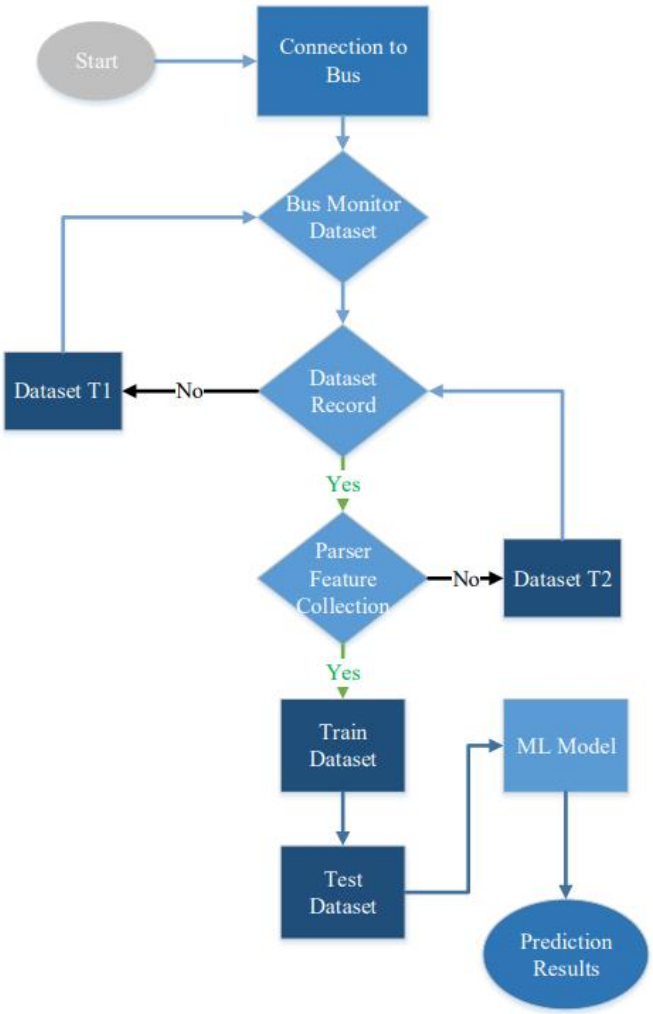
Numeric	1

Table 5.6 Scenario 1 attack data characteristics

Is Error Data	Count	Frequency
0	589	78.85 %
1	158	21.15 %

The most difficult part of solving a machine learning problem can be finding the right model. Different models are better suited for different types of data and problems. In the performance section, results to decide this evaluation is discussed.

Using labeled data, in this case, IsErrorData, classification methods in supervised learning can detect the anomalies in the bus. Furthermore, it is seen that dataset includes categorical features. Working with categorical variables requires working with classification methods. On the other hand, these methods are also searched in case labeled data is not used causing a multi-classification. So, we see the models performance on classifying various classes. In this thesis, supervised machine learning (ML) algorithms K-NN, Naïve Bayes, SVM, logistics regression and decision tree as classification models are investigated considering the attack scenarios. Accuracy and precision metrics are analyzed.



t

Figure 5-3 Anomaly detection strategy with Machine Learning Models

The method can be presented as a flow chart in Figure 5-3. T1 defines the dataset recording time. The anomalies in the dataset should be in the data during the time T1. If the recording is not successful a new dataset should be recorded at time T2. Using the dataset machine learning model is applied. The model is trained with realistic training data and the test dataset created by attack scenarios is investigated to define the anomaly.

CHAPTER 6

PERFORMANCE ANALYSIS

We use K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Naïve Bayes, Logistic Regression and Decision Tree machine learning models for anomaly detection for MIL-STD 1553 networks and evaluate performance metrics such as FPR and TPR. It is seen that K-NN uses the proximity of the data to each other, and SVN determines the data class via slicing. Naïve Bayes and logistic regression uses the probability of the class occurance. Decision tree forms a tree representing data defining with decision and leaf nodes representing a condition using entropy. Generally, data normalization and scaling are used to increase the performance. So dataset should be encoded as one hot or frequency encoding replacing the dataset with numbers or frequency. To select the correct model for the dataset, relations between data in the dataset that have various features and the target data are established. The main concern is to find out if the features are efficient to create classes and separation between them. If features are well connected to the target data linear models work well or tree-based models if data is in separate classes. If the results of the performance models are analyzed it is seen that anomalies change the accuracy of the machine learning models. To reduce the effect of anomalies scaling and normalization are used. Also, they decrease the training time and better accuracy. Test size is selected between 30 percent of the dataset since data being available for training the model which is between 70 is beneficial to deal with limited dataset.

6.1 Performance Analysis Scenario 1

In this scenario since there are two same addressed RTs on the bus, error features should not be used in case error flags can be activated. Using the labeled features decrease the error of the models.

Table 6.1 Performance Analysis of Classification Models Scenario 1

	Label	**Accuracy**	**Recall**	**F1**	**Precision**
KNN	Gap	0.63	0.63	0.56	0.52
	Is Error	1	1	1	1
SVM	Gap	0.54	0.54	0.71	0.79
	Is Error	1	1	1	1
Naïve Bayes	Gap	0.56	0.56	0.56	0.56
	Is Error	1	1	1	1
Logictic	Gap	0.62	0.62	0.5	0.57
Regression	Is Error	1	1	1	1
Decision Tree	Gap	0.84	0.84	0.80	0.78
	Is Error	1	1	1	1

Various detection algorithms are evaluated in terms of accuracy, F1 and especially precision and recall. The results are summarized in the tables above. As can be seen, the models trained via supervised algorithms correctly identified all the benign and anomalies in all the attack scenarios using labeled data. When other features such as Gap data are used it is seen that there are high bias and low variance causing underfitting since it is not a sufficient predictive feature. Decision tree performs well since it is very powerful in multi-class classification problems.

6.2 Performance Analysis Scenario 2

The accuracies of the models using Scenario 2 dataset are listed in Tables 6-2 below. The accuracy of the models changes with the selection of features in the dataset. It is important to select highly correlated features with related labels to have accurate model output. Machine learning models achieve high accuracy and precision values. Naive Bayes, Logistics Regression, and Decision Tree models fit well to detect the anomalies in the dataset under the proposed anomaly detection strategy in Figure 5-3. It is seen that there are no differences between the models in terms of accuracy and processing time to detect the anomalies in the dataset following the method in Figure 5-3.

The accuracy of the K-NN and SVM models are 99 %, indicating models achieve correct predictions as 99 % using the features in the dataset. However, given the imbalanced dataset (anomalies are rare), accuracy can be a misleading metric. So, F1 values are evaluated. Naive Bayes, Logistics Regression, and Decision Tree have better performance since their F1 values are higher. The lower recall for anomaly indicates that the model is less capable of identifying anomalous messages.

Table 6.2 Performance Analysis of Classification Models Scenario 2

	Label	**Accuracy**	**Recall**	**F1**	**Precision**
K-NN	Is Error	0.99	0.99	0.99	1
	Gap	0.59	0.59	0.7	0.77
SVM	Is Error	0.99	0.99	0.99	0.99
	Gap	0.51	0.51	0.66	0.74
Naïve Bayes	Is Error	1	1	1	1
	Gap	0.51	0.51	0.66	0.74
Logistic Regression	Is Error	0.98	0.98	0.97	0.97
	Gap	0.59	0.59	0.72	0.78
Decision Tree	Is Error	1	1	1	1
	Gap	0.55	0.55	0.69	0.76

6.3 Performance Analysis Scenario 3

The performance results for attack scenario 3 are given below.

Table 6.3 Performance Analysis of Classification Models Scenario 3

	Feature	Accuracy	Recall	F1	Precision
K-NN	Is Error	0.99	0.99	0.99	0.99
	Gap	0.54	0.54	0.44	0.38
SVM	Is Error	0.98	0.98	0.98	0.98
	Gap	0.60	0.60	0.47	0.60
Naïve Bayes	Is Error	1	1	1	1
	Gap	0.60	0.60	0.47	0.40
Logistic	Is Error	1	1	1	1
Regression	Gap	0.62	0.62	0.49	0.43
Decision Tree	Is Error	1	1	1	1
	Gap	0.59	0.59	0.47	0.40

6.4 Graphical Representation of Results

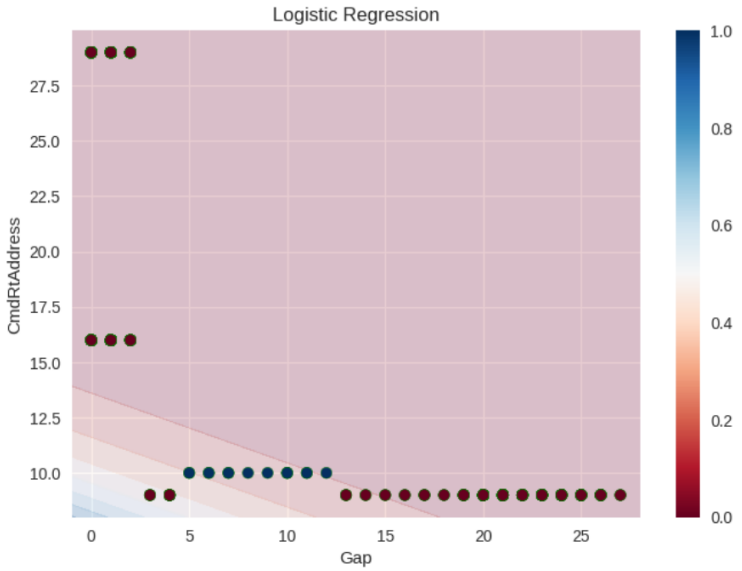


Figure 6-1 Logistic Regression Graph for Scenario 2

Logistic Regression determines the decision boundary so that it is possible to predict the class of data. In the figure 6-1, it can be seen that labels and the color shows the confidence of the model. RT 10 is responsible for anomalies.

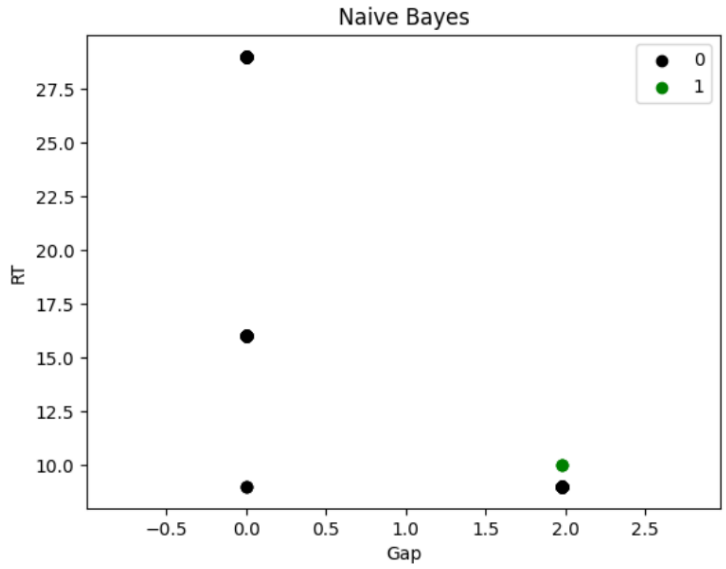


Figure 6-2 Naïve Bayes Graph for Scenario 2

It is seen that there are malicious data injection with RT10 in the Naïve Bayes method. It can be seen that there is an activity for RT addresses 9, 10, 16 and 29. From the analysis RT 10 can not transmit and receive data within 6.2E-05 ms, it can be defined as malicious RT. It is seen that precision and recall are 1 for labeled data to find out the anomaly which can be seen that RT10 is periodically sending messages in the dataset processing lower than 1 seconds for a 300 KiB data. It means that the model has successfully identified all the positive instances in the dataset, and there are no false negatives. It is visualized in Figure 6-2 where the green indicator are identified attacks.

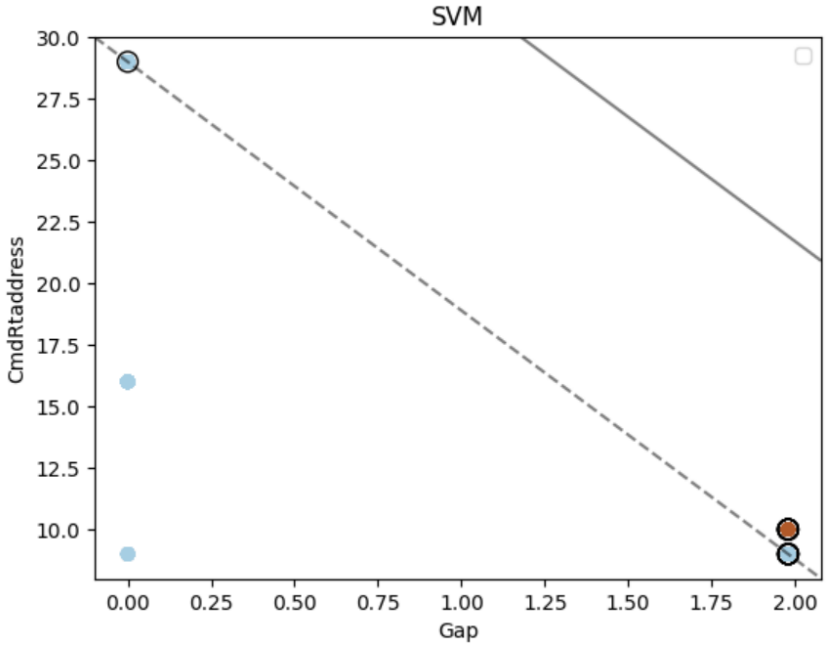


Figure 6-3 SVM Graph for Scenario 2

SVM divides the classes by finding a line in case of multiple dimensions. It defines the best hyperplane that separates the classes of data. Support Vector Machines can be used to define the separation between benign and anomalous data. Complexity grows if the training samples are increased. Choosing the right SVM kernel can be difficult. The aim is to create the best-separating hyperplane. Training an SVM with the Radial Basis Function kernel or linear kernel, C, and gamma parameters must be considered. Low C causes the decision surface smooth, while a high C classifies samples correctly, and Gamma determines kernel function width. In Figure 6-3, it is seen that separation can be achieved for RT 10 as orange color, its precision is 1 with a 0.99 recall value. It is displayed between the benign region hyperplane and optimal hyperplane. It is also seen that anomaly and benign data are very close to each other. There are also two support vectors touching the hyperplane.

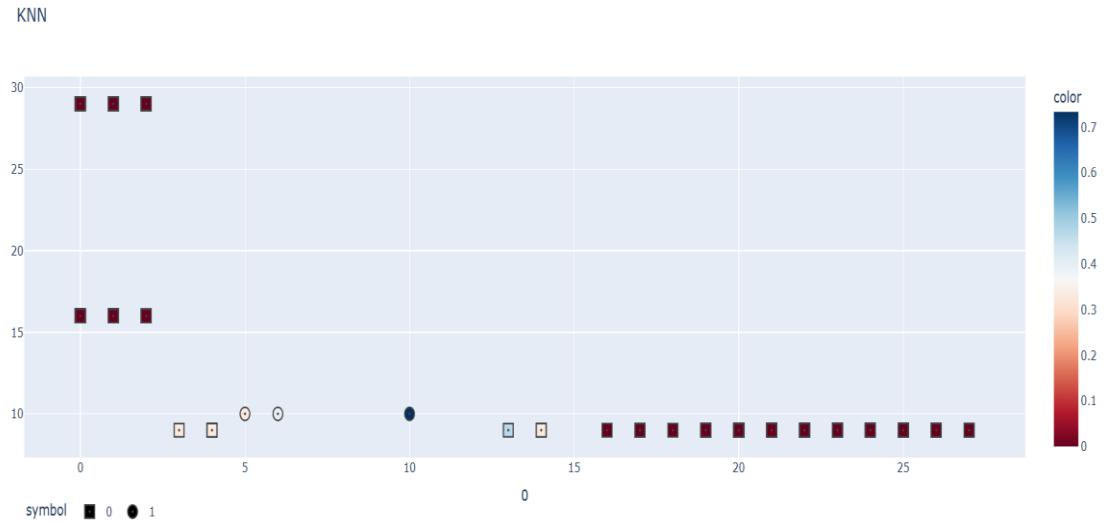


Figure 6-4 K-NN Graph for Scenario 2

In the figure 6-4, labels and the color displays the confidence of the model. Malicious RT10 activity can be easily noticed with blue color circular symbol.

K-NN is a simple machine learning algorithm that performs the classification according to the stored features. The results are achieved from K-NN implementation in Python in Figure 6-4. A dataset is divided into train and test sets for this supervised machine learning method. Its accuracy is 0.99 with a precision value of 1 and a recall value of 0.99. RT 10 is identified as malicious RT.

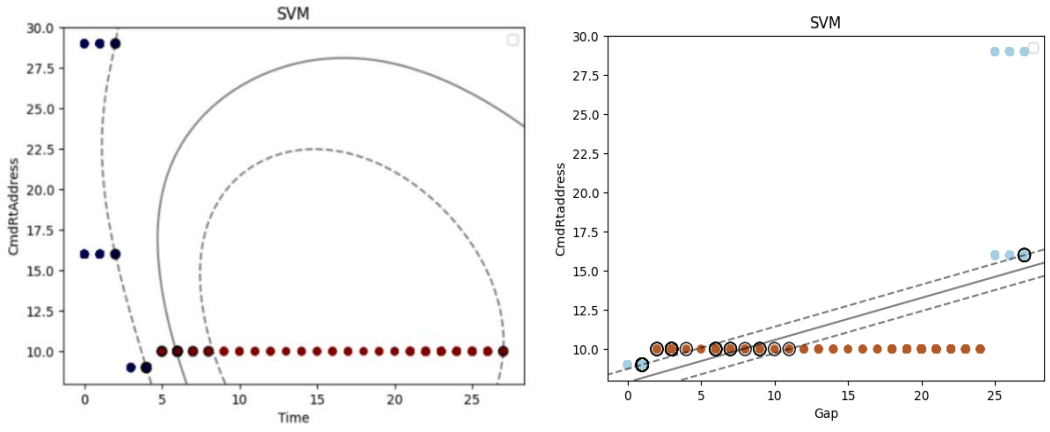


Figure 6-5 SVM Graphs for Scenario 3

The SVM model has two types methods as linear SVM and non-linear SVM. Linear SVM classifies linear data seperation. Non-linear SVM uses the kernel for a high-dimensional workspace to solve non-linear problems. In Figure 6-5, radial basis function kernel (RBF) kernel and linear kernel is the used to classify dataset.

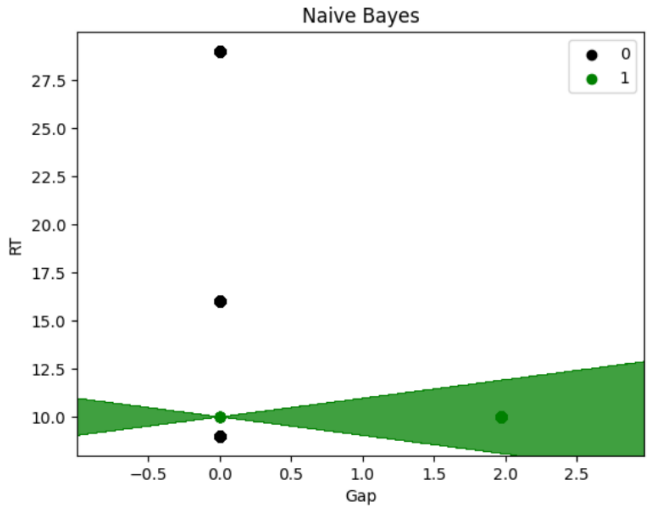


Figure 6-6 Gaussian Distribution Graph for Scenario 3

In figure 6-6, Naïve Bayes model defines the malicious RT using Gaussian Distribution. Green label defines the malicious RT activity considering the Gap value.

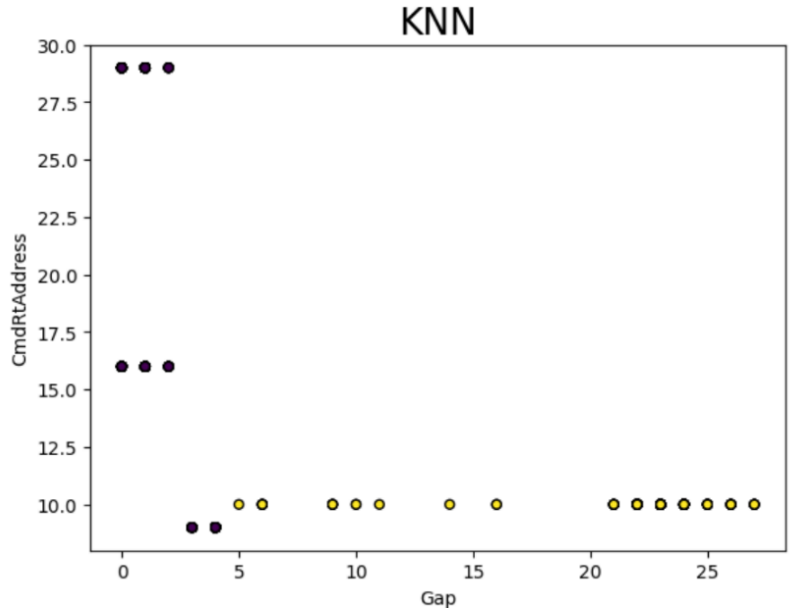


Figure 6-7. K-NN Graph for Scenario 3

In figure 6-7, RT 10 is defined as malicious RT injecting anomaly messages to the bus.

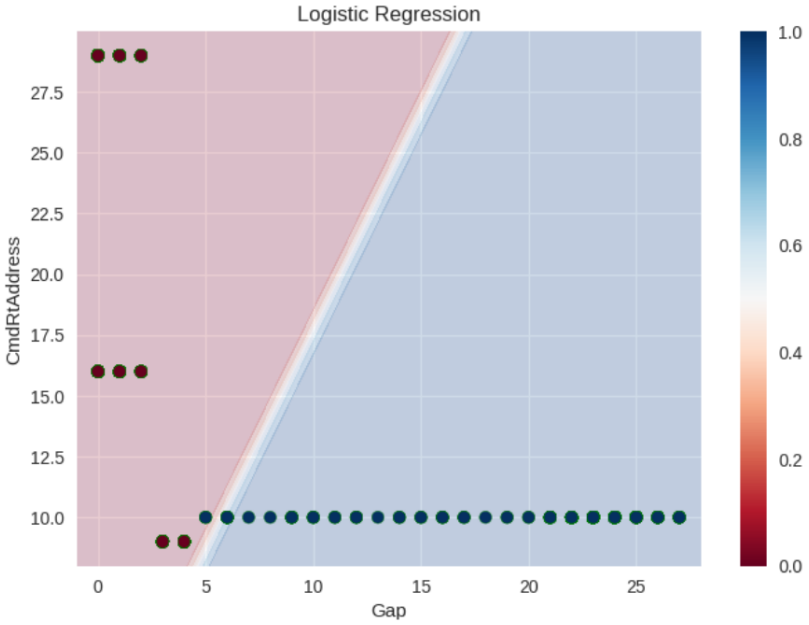


Figure 6-8. Logistic Regression Graph for Scenario 3

In figure 6-8, it can be concluded that the malicious RT activity can be defined more successful since dataset includes more anomaly than scenario 2.

6.5 Timing Analysis

Computation time for various features using different machine learning algorithms can be defined below. In practice using a large dataset requires a large storage capacity and slows down the computation.

Tables show the time needed to build the model. Selecting the correct feature using the correct model decreases the computation time and increases the model processing time.

Table 6.4 Performance Analysis of SVM method

Labels	Features	Processing time
Is error	Gap Cmdaddress	1 ms
Cmdaddress	Gap Cmdaddress	1 ms
StsRtAddress	Gap Cmdaddress	1 ms
Gap	Gap Cmdaddress	2-5 ms
Time	Gap Cmdaddress	7 min

Table 6.5 Performance Analysis of Naïve Bayes method

Label	Features	Processing time
Is error	Time Gap	0.1s
	CmdRtAddress	
Is error	Subaddress	0.1s
	Gap	
Is error	CmdRtAddress	0.1s
Gap	Time Gap	0.1s
	CmdRtAddress	
Gap	Subaddress	0.1s

Table 6.6 Performance Analysis of K-NN method

Label	Features	Processing time
Is error	Time Gap	0.1s
Gap	CmdRtAddress Subaddress	0.1s
Gap	Time Gap	1s
Is error	CmdRtAddress Subaddress	0.1s
Is error	Gap CmdRtAddress	0.1s

Table 6.7 Performance Analysis of Regression method

Label	Features	Processing time
Is error	Gap	0.1s
Gap	CmdRtAdress	10 min

Table 6.8 Performance Analysis of Decision Tree method

Label	Features	Processing time
Is error	All	0.1s
Gap	All	0.1s

6.6 Discussion

In Scenario 1, the models are trained via supervised algorithms, and it is seen that all anomalies in recorded data are identified with the highest accuracy using labeled feature. For the performance analysis, gap feature is also used in order to detect the anomaly. However the gap feature causes multi-class classification results. Classification machine learning methods assume that the class values are unordered. Gap values can be evaluated without discarding the ordering information by using ordinal classification methods. Ordinal classification can be used for a multiclass classification problems for the classes that has an order between them. So the predictive performance of a classifier can be improved. For the performance values of gap feature analysis, weighted scores are used. It should be noted that if a limit value for the gap is feature is defined, the performance results can be seen in the classification report. If gap feature is used, it is seen that decision tree performed better since it captures non-linear relationship between features and the target data. A non-linear relationship may not cause a constant change in the second variable even if there is a change in the first variable. It is seen that increasing the number of features increases all model performances. The results also show high bias and low variance meaning underfitting. It is unable to capture the relationship between the features and labels accurately. On the other hand, using labeled feature such as IsErrorData results low bias and low variance meaning model can detect anomaly.

In scenario 2, using a labeled data performance metrics such as recall and precision are equally high in all ML models however if other feature such as Gap is used it can be concluded that since the dataset does not have a predefined Gap value, models cannot detect the anomaly. For the analysis using other features unsupervised models should be evaluated.

In scenario 3, Naïve Byes and Decision Tree provide higher recall and precision values using labeled feature. Models are 100 % precise achieving perfect detection to prevent incorrect detection as anomaly.

When K-NN is used for anomaly detection by calculating the distance between data and their k-nearest neighbors, anomalies are needed to be far away from their nearest neighbors. So, it has a disadvantage when there is not such condition.

Naïve Bayes needs independent features and follows specific probability distributions (e.g., Gaussian for continuous features). In practice, some continuous features may not follow the assumed distribution, leading to suboptimal results.

Logistic Regression for scenario 2 shows that it is not the most suitable option for complex or highly imbalanced datasets with non-linear relationships.

SVM finds boundary enclosing most of the data in the dataset. This is useful for anomaly detection where anomalies are rare and are not explicitly labeled in the training data. SVM is less prone to overfitting, that is a critical need when working with limited training data or imbalanced datasets.

As a result, supervised machine learning models are effective for labeled dataset since performance metrics and processing time is optimal. Decision tree better handles imbalanced data and can capture complex relationships between features.

CHAPTER 7

CONCLUSION

MIL-STD 1553 is a widely used communication bus for military avionics systems. It is designed to be fault-tolerant and robust, but it does not have built-in security features. This is a concern because the bus is becoming increasingly connected to the outside world, which could make it vulnerable to malicious attacks.

To have a secure 1553 system against cyber attacks, authentication and encryption are needed to include in the protocol. But it is very hard to implement because there are lots of system components in operation. There is only a parity check in MIL-STD 1553 system in terms of failure detection mechanism.

In this thesis, we propose to introduce machine learning (ML) algorithms to increase the cyber resiliency of the bus. These algorithms can provide anomaly detection and warn the pilot or maintenance personnel via recording capabilities. We assume that the machine learning algorithms work offline on the previously collected data. It is seen that all anomalies in recorded data are identified with the highest accuracy using labeled Is Error label.

We identified attack scenarios and message features that can be used to detect anomalies on the MIL-STD 1553 bus. We built a testbed with real avionics hardware and a simulator that can generate attack messages. We injected the messages generated by the simulator into the bus using a PCIe card connected to the PC with the simulator. We also used bus monitoring and analysis tools to collect data. We modified the driver of the PCIe card and wrote software to parse and analyze the traffic data. We then performed anomaly detection with K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), Naïve Bayes, Logistic Regression and Decision Tree machine learning (ML) methods and compared their results.

As a result, decision tree is a more effective anomaly detection method since performance metrics and processing time is optimal. These models are better handle imbalanced data and can capture complex relationships between features.

Additional properties like periods and response time can be analyzed in detail using related machine learning algorithms. Various new attack vectors can be implemented on the bus to detect the anomaly. It should be noted that errors in the anomaly detection algorithms cause important problems however other applications of machine learning models can be just for advisory purposes.

The ML models are implemented and run offline to detect if any malicious software and hardware are placed in the aircraft. Machine learning models is compared in terms of correctness and speed for a two minutes recorded dataset. Features are analyzed and it is seen that selecting the correct feature affects the performance. As a result, the outcome of the study showed that the realistic scenarios in which any cyber attack can cause an anomaly in the MIL-STD 1553 bus that can be detected by the machine learning models.

REFERENCES

[1] B. Losier, R. Smith, and V. Roberge, “Design of a time-based intrusion detection algorithm for the MIL-STD 1553,” *Royal Military College of Canada, Rep.*, No. DTAES-8 2102, Jan. 2019. (Accessed on 12/04/2019).

[2] S. JJ. Genereux, A. KH. Lai, C. O Fowles, V. R. Roberge, Guillaume PM Vigeant, and Jeremy 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, 2019.

[3] F. Onodueze and D. Josyula, “Anomaly detection on mil-std- 1553 dataset using machine learning algorithms,” *IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 592–598, 2020.

[4] O. Stan, Adi 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, 2019.

[5] F. Levy et al. “AnoMili: Spoofing Prevention and Explainable Anomaly Detection for the 1553 Military Avionic Bus,” Ben-Gurion University of the Negev, 2022.

[6] 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”. *arxiv Prepr*., no. 1707.05032, 2017.

[7] R. Yahalom, A. Steren, Y. Nameri, M. Roytman, A. Porgador, and Y. Elovici, “Improving the effectiveness of intrusion detection systems for hierarchical data,” *Knowledge-Based Syst*., vol. 168, pp. 59–69, 2019.

[8] R. Yahalom et al. “Datasets of RT spoofing attacks on MIL-STD- 1553 communication traffic”. *Data Br.*, vol. 23, pp. 103863, 2019.

[9] “RT Spoofing Attacks on MIL-STD 1553 Communication Traffic”, https://doi.org/10.17632/jvgdrmjvs3.3. (Accessed on 9/12/2023).

[10] D. De Santo, CS. Malavenda, SP. Romano, and C. Vecchio, “Exploiting the MIL-STD 1553 avionic data bus with an active cyber device,” *Computers & Security*, vol.100:102097, 2021.

[11] T. D. Nguyen, “Towards mil-std 1553b covert channel analysis,” Technical report, Naval Postgraduate School Monterey CA, 2015.

[12] N. T. Van, T. N. Thinh, and L. T. Sach, “An anomaly-based network intrusion detection system using deep learning,” *Int. Conf. Syst. Sci. Eng.*, pp. 210–214, 2017.

[13] U. and Q. H. Mahmoud, “A filter-based feature selection model for anomaly based intrusion detection systems,” *IEEE Int. Conf. Big Data*, pp. 2151–2159, 2017.

[14] Y. Du, H. Wang, and Y. Pang, “A hidden Markov models-based anomaly intrusion detection method,” *5th World Congr. Intell. Control Autom.*, vol. 5, pp. 4348–4351, 2004.

[15] V. Chandola, A. Banerjee and V. Kumar, “Anomaly detection: A survey. ACM computing surveys,” 2009.

[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*, 2021

[17] H. Sağırkaya, G. Durgun, “Avionics Simulation Environment,” *International Test Conference*, 2021.

[18] H. Sağırkaya, K. Polat, Y. E. Yilmaz, “Avionics Modeling Environment,” *The IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications*, 2021.

[19] U. Qamar, M. Summair Raza, Data Science Concepts and Techniques with Applications, Springer, 2020.

[20] “MIL-STD-1553 Designer’s Guide.” http://www.ddc-web.com. (Accessed on 9/12/2023).

[21] M. Srinivas, G. Sucharitha, A. Matta. Machine Learning Algorithms and Applications, John Wiley, 2021.

[22] K. Sachdev, H. S. Saad, I. Traore, K. Ganame, and O. Boudar, “Unsupervised Anomaly Detection for MIL-STD 1553 Avionic Platforms Using CUSUM,” *ECPSCI.*, vol. 2, 2022.

[23] P. M. Hayden, D. K. Woolrich, K. D. Sobolewski, “Providing Cyber Situational Awareness on Defense Platform Networks.” *The Cyber Defense Review*, vol. 2, no. 2, pp. 125-140, 2017.

[24] “MIL-STD 1553C Digital Time Division Command/Response Multiplex Data Bus,” Department of Defense Interface Standard.

[25] C. Chio, D. Freeman. Machine Learning& Security Protecting Systems with Data and Algorithms. O’Reilly, 2018.

[26] “Simsoft Information Technologies Corporate Catalogue. Advanced Test Lab with Automated Simulators.” https://www.simsoft.com.tr/brosur/Simsoft-ATLAS.pdf. (Accessed on 9/12/2023).

[27] “Alta Data Technologies LLC.” https://www.altadt.com/products/mil-std-1553. (Accessed on 9/12/2023).

[28] “Astronics Corporation.” https://www.astronics.com/portable-avionics-interfaces-for-usb-2. (Accessed on 9/12/2023).

[29] M. S. Sorower. A Literature Survey on Algorithms for Multi-label Learning. Computer Science, 2010.

[30] “Alta Data Technologies LLC.” AltaAPI Users Software Manual. (Accessed on 9/12/2023).

APPENDICES

A. Drivers for MIL-STD-1553 bustools

The content of M1553DataChannel (.Cs File)

B. ML Algorithms

ML Algorithms K-NN (Jupyter Notebook File)
ML Algorithms SVM (Jupyter Notebook File)
ML Algorithms Naive Bayes (Jupyter Notebook File)
ML Algorithms Logistics Regression (Jupyter Notebook File)
ML Algorithms Decision Tree (Jupyter Notebook File)

C. Parser

Parser for MIL-STD-1553 dataset (.Cs FILE)

D. Message List

Message list of MIL-STD 1553 Scheduler for the Dataset (Excel File)

E. Dataset

Simulator Scenario 1 (Raw- Parsed Excel and Csv File)

Simulator Scenario 2 Dataset (Raw- Parsed Excel and Csv File)

Simulator Scenario 3 Dataset (Raw- Parsed Excel and Csv File)

F. CONFIGURATION

Configuration of Simulator and Machine Learning Libraries