Feature Engineering for a MIL-STD-1553B LSTM Autoencoder Anomaly Detector

Dakotah Soucy[1] and Brian Lachine[2]
[1]Director Technical Airworthiness and Engineering Support, Ottawa, Canada
[2]Royal Military College of Canada, Kingston, Canada

dakotah.soucy@forces.gc.ca
brian.lachine@rmc.ca

Abstract: The MIL-STD-1553B data bus protocol is used in both civilian and military aircraft to enable communications between subsystems. These interconnected subsystems are responsible for core services such as communications, flow of instrument data and aircraft control. With aircraft modernization, threat vectors are introduced through increased inter-connectivity internal and external to the aircraft. The resulting potential for exploitation introduces a requirement for an intrusion detection capability in order to maintain the integrity, availability and reliability of data transmitted using the MIL-STD-1553B protocol, safety of the aircraft and overall, to achieve mission assurance. Research in recent years has investigated signature, statistical and machine learning based solutions to detect attacks on MIL-STD-1553B buses. Of the different techniques, those based on machine learning have shown extremely good results. The aim of this research is to improve the performance of an existing Long Short-Term Memory Auto-Encoder by refining the feature engineering phase of its pipeline. The improvement in the detector's overall effectiveness was accomplished through feature engineering focused on feature generation and selection. Five different attack datasets were used as the starting point, consisting of four different denial of service attacks and one data integrity attack. From initial feature extraction of 155 features, two feature generation techniques were employed to create over 38,000 features as a starting point. Using five different MIL-STD-1553B datasets and three feature selection techniques, fifteen different Long Short-Term Memory Auto-Encoder models were created, trained and evaluated using common performance metrics and compared to those of the original anomaly detector. This research demonstrated marked performance improvement achieved by the feature engineering refinements made in comparison to those of the original model. Equally important, this research also showed a significant reduction in the number of features required to achieve this performance gain. In the context of miliary air operations, the ability to improve detection capabilities with less data is important to the technical solutions that contribute to the achievement of cyber mission assurance.

Keywords: MIL-STD-1553B, Anomaly Detection, Deep Learning, LSTM Autoencoder, Aviation Cybersecurity

1. Introduction

The MIL-STD-1553B data bus protocol is used to enable communications between subsystems in civilian and military aircraft. These subsystems, referred to as remote terminals (RTs) provide core aircraft services across the data bus. MIL-STD-1553B was introduced in 1978 and was designed for reliability and safety in an air-gapped environment. With aircraft modernization, threat vectors may be introduced through connections such as data links and maintenance diagnostic tools.

These additional threat vectors to the MIL-STD1553B data bus create an opportunity in which adversaries have the ability to exploit vulnerabilities in the MIL-STD1553B protocol. This potential for exploitation of the MIL-STD-1553B protocol introduces a requirement for Intrusion Detection System (IDS) in order to maintain the reliability of the MIL-STD-1553B protocol and safety of the aircraft. Both signature and anomaly-based IDS have recently been researched and provide viable options for monitoring vulnerabilities in the MIL-STD-1553B data bus introduced by these new threat vectors (Bedard, 2019; Genereux et al, 2020; Stan et al, 2020; Onodueze and Josyula, 2020; De Santo et al, 2021; Levy et al, 2022; Banks et all, 2022; Wrana et al, 2022; and Harlow, Lachine and Roberge, 2024).

This research focuses on the feature engineering component of a Long-Short Term Memory (LSTM) MIL-STD-1553B deep learning anomaly detector (Harlow, Lachine and Roberge, 2024) in order to improve its overall effectiveness. Feature generation and selection form the core of this effort.

2. MIL-STD-1553B

The MIL-STD-1553B standard was published in 1978 by the United States Department of Defense (DoD) and defines the mechanical, electrical and functional characteristics of a serial data bus. This standard defines a multi-point, serial communication bus between terminals controlled through a command and response protocol. The typical architecture for MIL-STD-1553B is shown in Figure 1, which consists of terminals connected through a dual redundant communications bus. The standard defines three distinct terminals:

1. Bus Controller (BC): The terminal responsible for initiating and directing information transfer on the bus.
2. Bus Monitor (BM): The terminal responsible for receiving and storing select bus traffic for use at a later time.
3. Remote Terminal (RT): Any terminal not operating as a BC or BM.

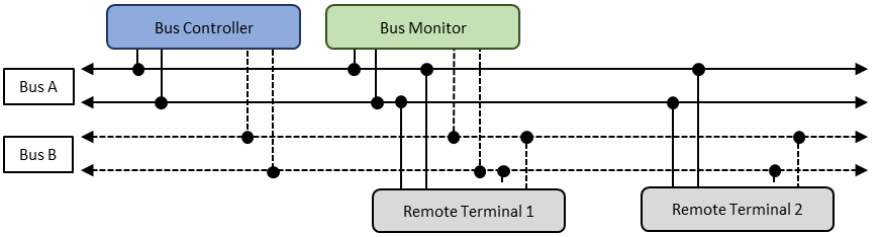


Figure 1: Example of MIL-STD-1553B bus topology with a BC, BM, and two RTs connected by a Dual Redundant Bus

Additionally, there is a maximum of 32 addresses on the MIL-STD-1553B bus. Address 31 is reserved for broadcast transactions and the remaining 0 to 30 addresses are assignable to RTs (U.S. DoD, 1978).

2.1 Data Link Layer

Data transfer on the bus is accomplished through messages which are comprised of 20-bit components called words. MIL-STD-1553B defines three word types: command, data and status. Figure 2 depicts the structure for each type of word. These words combine to form the larger messages and the standard defines two types of message formats: data messages and control messages.

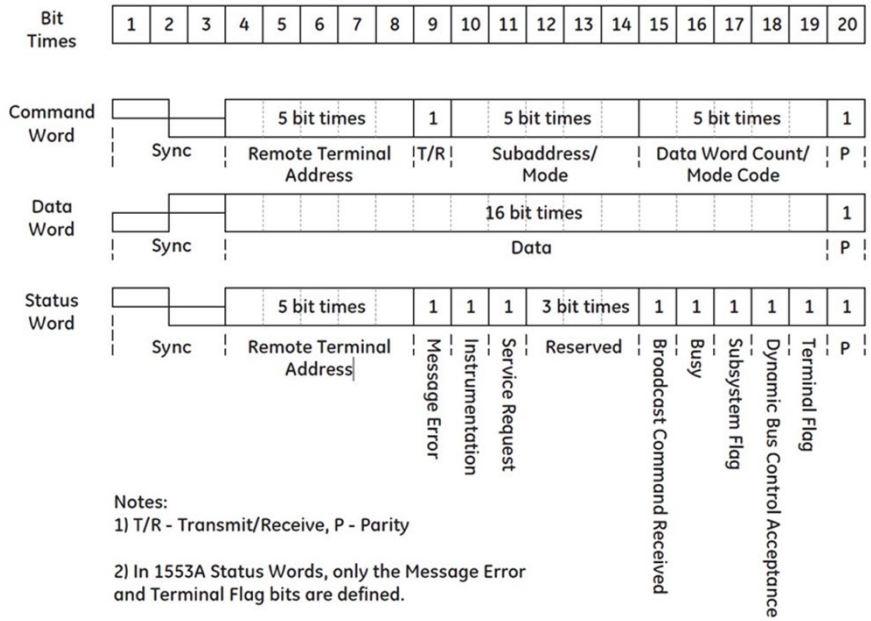


Figure 2: MIL-STD-1553B Word Formats (Abaco-Systems, 2019)

Data messages are initiated by the BC issuing command words on the bus followed by the RTs transmitting and/or receiving the data on the bus. All RTs which are connected to the data bus have the ability to read all messages transmitted, but only the addressed RTs are expected to carry out the command sent by the BC. These data messages are further separated into communications between specific RTs and broadcast communications. There are three data message types between specific RTs: BC to RT, RT to BC and RT to RT. Additionally, there are two broadcast data message types: BC to RTs and RT to RTs.

Control messages enable the BC to monitor and control the bus by issuing mode commands to the RTs. Control messages are a set of predefined functions and can contain command and data words, or solely mode codes

from the BC. When directed to a specific RT, the response can contain status and data words, or solely a status word depending on the initial mode command word. When the control message is a broadcast message, there are no responses from the RT(s).

The BC directs all communications on the bus between RTs following a predefined schedule. This schedule contains two different types of message schedules: periodic and aperiodic. Periodic messages are transmitted at fixed time intervals according to the schedule. Aperiodic messages are conditional and therefore not sent at a fixed interval, although they still retain a fixed time slot in the schedule if they are to be transmitted. Collections of periodic and aperiodic messages are combined to form a minor frame within the schedule. Furthermore, a collection of minor frames forms the main schedule, also known as the major frame.

2.2 MIL-STD-1553B Attack Types and Outcomes

The MIL-STD-1553B data bus protocol was designed for reliability and safety of the system. As stated by the standard (U.S. DoD, 1978), all bus communications follow a predetermined cyclical, real-time schedule controlled by the BC. All RTs manufactured are expected to follow the standards defined by MIL-STD-1553B. However, adversaries are not constrained by these standards and can manipulate the protocol to their advantage to achieve their desired outcome.

Stan et al (2019) specify two main attack methods, message manipulation and behaviour manipulation:

1. Message manipulation: Modification of legitimate words that are transmitted over the data bus.
2. Behaviour manipulation: Altering the normal behaviour of a compromised component such as transmitting fake messages in an unusual timing or order.

Stan et al (2019) then outline three attack outcomes that can be accomplished via the two attack methods. These attack consequences are Denial of Service (DoS), data leakage, and data integrity violation. These attack consequences can be caused by either of the two attack methods discussed above.

3. Feature Engineering

A feature in the context of machine learning is simply an individual measurable property or characteristic of the object being observed (Elgendy, 2020). A feature is derived from raw data and acts as a key input into our model, denoting the importance of considering feature generation and selection.

3.1 Feature Generation

Feature generation is the process of creating new input variables from available data. There are many different approaches for feature engineering and they are specific to the type of data utilized. To generate features, McGaughey et al (2018) used of the processing module netAI (Zander and Williams, 2011) to create network specific flow features. Stan et al (2020) also implemented the use of a feature generation module called *Time Series FeatuRe Extraction on basis of Scalable Hypothesis* (tsfresh) by Christ et al. (2018) that allowed the calculation of features from time-series datasets. Self-Organizing Maps (SOM) were used by da Silva Rodrigues et al (2021) to generate new features with noted improvements in F1-Scores depending on the classification model used. Brownlee (2020) highlighted other feature generation methods such as polynomial transformation. Polynomial transformation utilizes simple mathematical operations to create additional features that may transform the original feature set into more effective features.

The polynomial transformation method was also utilized by McGaughey et al (2018) with results demonstrating that these generated features allowed the model to perform predictions with an improved detection rate and Matthew Correlation Coefficient (MCC) score.

3.2 Feature Selection

Feature selection has been an intrusion detection research focus for decades (Mukkamala and Sung, 2002; Thakkar and Lohiya, 2023) and is a technique of selecting a subset of features that will provide the most relevant data for input into the model (Brownlee, 2020). Feature selection can be grouped into two main categories; unsupervised and supervised. Supervised feature selection is used in this research and can be further grouped into three categories; intrinsic, wrapper and filter methods (Guyon and Elisseeff, 2003). Intrinsic feature selection refers to machine learning models that have embedded processes for selecting the best features, such as Least Absolute Shrinkage and Selection Operator (LASSO) which uses penalization functions or decision trees

(Muthukrishnan and Rohini, 2016). The wrapper feature selection method recursively selects a subset of the features, trains the model on these features then evaluates the performance El Aboudi and Benhlima, 2016). Lastly, the filter method selects features independent of the machine learning algorithm, and instead uses statistical methods to determine which features to use (Abujazoh et al, 2023).

4. Design

In order to understand how model effectiveness may be impacted by feature engineering, the entire pipeline for a deep learning anomaly detector needs to be designed and constructed. The same MIL-STD-1553B datasets and general model used by Harlow, Lachine and Roberge (2024) were used as the starting point to which two feature generation techniques and three feature selection techniques were added. The first feature generation method chosen was polynomial expansion that leverages the *sklearn* library. The second technique is based on time series characteristics using *tsfresh*, an application also used by Stan et al (2019). After feature generation, selecting the most useful features would then need to be conducted. The feature selection methods selected are Analysis of Variance (ANOVA), Fast Orthogonal Search (FOS) (Korenberg and Paarmann, 1989; McGaughey et al, 2018) and Predictive Power Score (PPS) (Wetschoreck, 2020, Demertzis et al, 2021). The evaluation of the models utilizes common machine learning model evaluation metrics.

4.1 Data Collection and Munging

The raw data collected by Harlow, Lachine and Roberge (2024) forms the datasets used for this research and includes both baseline data and anomalous data. The anomalous data was created using the evaluation tool created by Paquet (2014). The datasets are further divided into two main groups based on anomalous or attack types: DoS and data integrity violation. These datasets have been collected using the Abaco BusTools-1553 software suite in a simulated aircraft environment. Abaco BusTools-1553 records data bus traffic in the Bus Monitor Data Files Extended (BMDX) message format. Within each of the n messages, the data contained in the words is in the format outlined in Figure 3.

```
typedef struct api_bm_mbuf
    {
    BT_U32BIT           messno;                 // Message number (generated by API, 1-based)
    BT_U32BIT           int_status;             // Interrupt status from board
    BT1553_TIME         time;                   // Time of message (48-bits, 1 us LSB)
    BT1553_COMMAND      command1;               // 1553 command word #1 (Rx for RT→RT)
    BT_U16BIT           status_c1;              // 1553 command word #1 error status
    BT1553_COMMAND      command2;               // 1553 command word #2 (Tx for RT→RT)
    BT_U16BIT           status_c2;              // 1553 command word #2 error status
    BT1553_BMRESPONSE   response1;              // 1553 response time #1 (byte)
    BT1553_BMRESPONSE   response2;              // 1553 response time #2 (byte)
    BT1553_STATUS       status1;                // 1553 status word #1 (Transmit for RT→RT or
                                                // Broadcast RT→RT)
    BT_U16BIT           status_s1;              // 1553 status word #1 error status
    BT1553_STATUS       status2;                // 1553 status word #2 (Receive for RT→RT,
                                                // NULL for Broadcast RT→RT)
    BT_U16BIT           status_s2;              // 1553 status word #2 error status
    BT_U16BIT           value[BT1553_MBUFCOUNT];    // 1553 data words
    BT_U8BIT            status[BT1553_MBUFCOUNT];   // 1553 status for data words
    }
API_BM_MBUF;
```

Figure 3: Abaco BusTools-1553 BMDX Message Structure (reformatted from Abaco-Systems, 2019)

The data structure fields from in Figure 3, were munged to provide more granular information such as RT address, transmit/receive, sub-address and number of words to better correlate to how words are typically presented in MIL-STD-1553B. This is accomplished by separating the data contained in the command word(s), data word(s) and/or status word(s) into their respective formats as outlined in section 2.1.

4.2 Feature Engineering

After data collection and munging, feature engineering is the next pipeline step in order to prepare the data for use in the anomaly detection model. This step first consists of feature generation followed by feature selection as outlined in Figure 4.

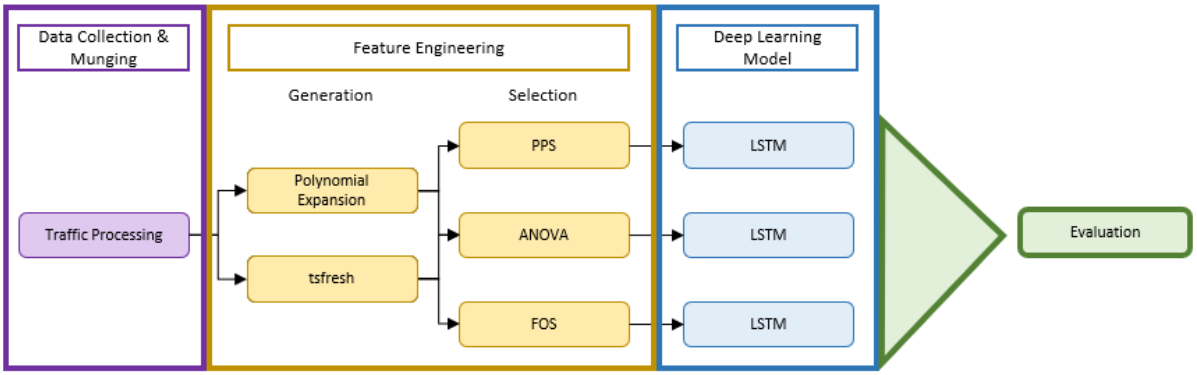


Figure 4: Overview of Feature Engineering Focused Pipeline

4.2.1 Feature Generation

In this step, the primary features created in the data collection and munging step are used to generate multiple new features. The first method of feature generation utilizes the scikit learn *sklearn.preprocessing.PolynomialFeatures* module (Buitnick et all, 2013). This module generates a new feature matrix with all polynomial combinations of the features up to the degree specified. For the purpose of this research the maximum polynomial degree of two is used due to processing speed and resource consumption. The second method utilizes the python tool *tsfresh*. *tsfresh* is a python package that generates time-series features using 78 different feature calculation modules (Christ et al, 2018). *tsfresh* does this by running the different calculation modules with the data from the features in the dataset, and then creates additional features based on the results from the calculation modules.

4.2.2 Feature Selection

After feature generation, feature selection is used in order to select a smaller number of features to be used as input into each discrete run of the learning model. The feature selection methods ANOVA, PPS and FOS are used to select a subset of features in parallel from the same starting features created in the previous phase.

4.3 Evaluation of Deep Learning Model

In order to measure the effectiveness of the anomaly detection method, the results were evaluated using multiple methods. The methods utilized metrics derived from the confusion matrix: precision, recall, accuracy, Area Under Receiver Operating Characteristic Curve (AUROCC), and Matthews Correlation Coefficient (MCC). These metrics were used to measure the performance of the anomaly detector. These results were then compared against recent work in the field, specifically the original LSTM detector by Harlow, Lachine and Roberge (2024).

5. Results

The data collected comprised both of baseline and anomalous datasets. All of the datasets were representative of an aircraft in the cruise phase of flight. There was a total of three baseline datasets collected, all of which utilized the same master schedule. Through investigation of these three benign datasets and as expected, it was confirmed that they were identical. As such, only one baseline dataset was selected for use in this experiment as the others would not contribute any additional information. The anomalous data collected comprised of a total of five datasets across two attack categories as outlined section 2.2:

1) DoS
   a) NetDisrupt statusword 250820 (*Disrupt*): Network *disrupt*ion
   b) RT-SA deny statusword rt18 sa32 250820 (*Deny*): RT deny
   c) RT-*SA* deny statusword rt18 sa1 250820 (*SA deny*): RT subaddress deny
   d) RT-*SA* deny statusword rt18 sa1 rec2 250820 (*SA deny 2*): RT subaddress deny

2) Data Integrity Violation
   a) *Hijack* rt18 sa6 w56 250820 (*Hijack*): RT *hijack*

These anomalous datasets are named using the following convention: the first part is the type of attack, followed by the associated RT details and ending in the day, month, year recorded. In the remainder of this paper, these datasets are referred to by their abbreviated forms outlined in parentheses above. Through exploratory data analysis, it was noted that the *SA deny* and *SA deny 2* datasets contained very similar data. As such, *SA deny* was utilized for feature selection and *SA deny 2* was used solely for generating model performance metrics. Both the baseline and anomalous datasets contain just over 1 million MIL-STD-1553B messages each. The anomalous datasets are imbalanced, containing little anomalous traffic compared to normal traffic. The specific amount of anomalous traffic for each dataset is as follows: *Disrupt* - 17.2%, *Deny* - 2.1%, *SA deny* - 5.7%, *SA deny 2* - 4.2%, and *Hijack* - 17.2%.

5.1 Feature Generation

A complete list of 155 primary features was created after the data munging was completed. Next, the feature generation techniques were then applied to these munged datasets. Polynomial expansion utilizing the *sklearn.preprocessing.PolynomialFeatures* tool created a total of 12,247 derived features from the existing 155 primary features. *tsfresh* created an additional 26,071 derived features from the existing 155 primary features. Additionally, *tsfresh* automatically eliminates generated features that contain no additional information, such as no variance in the values generated. Adding the features from the two generation techniques to the original primary feature set resulted in a total of 38,473 unique features.

5.2 Feature Selection and Models

After creation of the extended datasets, the feature selection techniques were implemented to determine the features to be used for each model. Model development consisted of two approaches; attack specific and general. The first approach was to use a specific model for each attack type as shown in Figure 5.

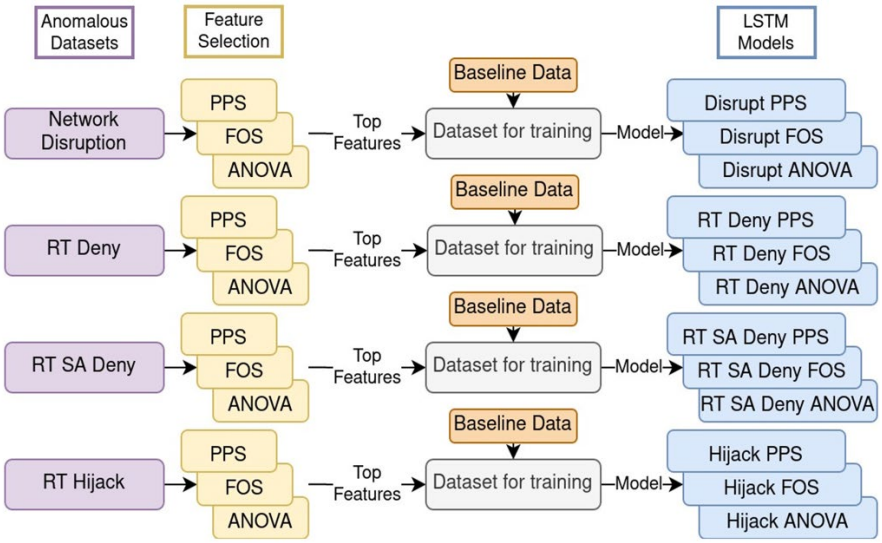


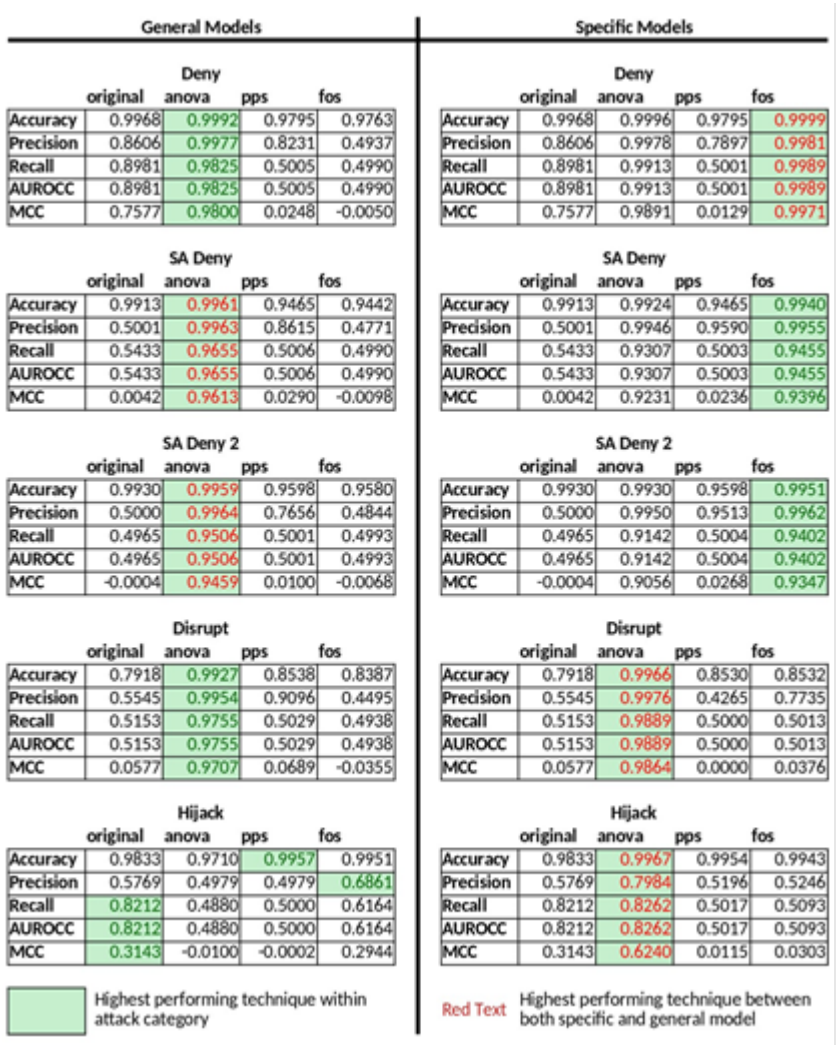
Figure 5: Attack and Feature Selection Specific Models

Each labelled attack type dataset was fed into the three feature selection technique tools as outlined in Figure 5. The top features output from each feature selection technique were then used to define the features selected from the baseline dataset to train the models. The process for selecting the top features utilized elbow curves. Upon completion, there was a specific model for each attack type and each feature selection method, resulting in 12 specific models.

The second approach was to create a general model by using the three feature selection techniques to select the top features for each attack type. These top features for each attack type were then combined and used as the features from the baseline to train the general model. This resulted in a total of 3 general models, one for each feature selection method. The purpose of the general model was to compare the results of a general model to the specific model to evaluate the ability for a more streamlined approach for feature selection and model generation. The number of top features for all models was determined through the analysis of elbow curves.

5.3 Model Performance

The resulting performance metrics of the fifteen models were then compared to the research originally conducted by Harlow, Lachine and Roberge (2024) which demonstrated an overall improvement in the effectiveness of the anomaly detection pipeline as illustrated below in Figure 6.



General Models

Deny

	original	anova	pps	fos
Accuracy	0.9968	0.9992	0.9795	0.9763
Precision	0.8606	0.9977	0.8231	0.4937
Recall	0.8981	0.9825	0.5005	0.4990
AUROCC	0.8981	0.9825	0.5005	0.4990
MCC	0.7577	0.9800	0.0248	-0.0050

SA Deny

	original	anova	pps	fos
Accuracy	0.9913	0.9961	0.9465	0.9442
Precision	0.5001	0.9963	0.8615	0.4771
Recall	0.5433	0.9655	0.5006	0.4990
AUROCC	0.5433	0.9655	0.5006	0.4990
MCC	0.0042	0.9613	0.0290	-0.0098

SA Deny 2

	original	anova	pps	fos
Accuracy	0.9930	0.9959	0.9598	0.9580
Precision	0.5000	0.9964	0.7656	0.4844
Recall	0.4965	0.9506	0.5001	0.4993
AUROCC	0.4965	0.9506	0.5001	0.4993
MCC	-0.0004	0.9459	0.0100	-0.0068

Disrupt

	original	anova	pps	fos
Accuracy	0.7918	0.9927	0.8538	0.8387
Precision	0.5545	0.9954	0.9096	0.4495
Recall	0.5153	0.9755	0.5029	0.4938
AUROCC	0.5153	0.9755	0.5029	0.4938
MCC	0.0577	0.9707	0.0689	-0.0355

Hijack

	original	anova	pps	fos
Accuracy	0.9833	0.9710	0.9957	0.9951
Precision	0.5769	0.4979	0.4979	0.6861
Recall	0.8212	0.4880	0.5000	0.6164
AUROCC	0.8212	0.4880	0.5000	0.6164
MCC	0.3143	-0.0100	-0.0002	0.2944

Highest performing technique within attack category

Specific Models

Deny

	original	anova	pps	fos
Accuracy	0.9968	0.9996	0.9795	0.9999
Precision	0.8606	0.9978	0.7897	0.9981
Recall	0.8981	0.9913	0.5001	0.9989
AUROCC	0.8981	0.9913	0.5001	0.9989
MCC	0.7577	0.9891	0.0129	0.9971

SA Deny

	original	anova	pps	fos
Accuracy	0.9913	0.9924	0.9465	0.9940
Precision	0.5001	0.9946	0.9590	0.9955
Recall	0.5433	0.9307	0.5003	0.9455
AUROCC	0.5433	0.9307	0.5003	0.9455
MCC	0.0042	0.9231	0.0236	0.9396

SA Deny 2

	original	anova	pps	fos
Accuracy	0.9930	0.9930	0.9598	0.9951
Precision	0.5000	0.9950	0.9513	0.9962
Recall	0.4965	0.9142	0.5004	0.9402
AUROCC	0.4965	0.9142	0.5004	0.9402
MCC	-0.0004	0.9056	0.0268	0.9347

Disrupt

	original	anova	pps	fos
Accuracy	0.7918	0.9966	0.8530	0.8532
Precision	0.5545	0.9976	0.4265	0.7735
Recall	0.5153	0.9889	0.5000	0.5013
AUROCC	0.5153	0.9889	0.5000	0.5013
MCC	0.0577	0.9864	0.0000	0.0376

Hijack

	original	anova	pps	fos
Accuracy	0.9833	0.9967	0.9954	0.9943
Precision	0.5769	0.7984	0.5196	0.5246
Recall	0.8212	0.8262	0.5017	0.5093
AUROCC	0.8212	0.8262	0.5017	0.5093
MCC	0.3143	0.6240	0.0115	0.0303

Red Text — Highest performing technique between both specific and general model

Figure 6: Results for Specific and General Models

The metrics used included accuracy, precision, recall, AUROCC and MCC. To obtain the evaluation metrics, each anomalous dataset was processed through the respective LSTM autoencoder, which was trained on the baseline traffic. Scores were recorded for each anomalous dataset and their respective models based on reconstruction error. The green highlighted cells indicate the feature selection technique that resulted in the highest performance for each attack in both the general and specific models and the red text indicates the technique that resulted in the highest performance between the general and specific models. The *original* columns refer to the results from Harlow, Lachine and Roberge (2024).

5.4 Features Selected

There were few to no selected features shared between the *hijack* dataset and the *Deny*, *SA deny* and *Disrupt* datasets. Figure 7 shows an example of this, where the blue highlighted fields represent common features between at least two datasets within the ANOVA selection technique. Additionally, primary features were only selected by all feature selection techniques on average 13% of the time, as illustrated by the large number of generated features shown in Figure 7. The fact that the majority of the selected features were generated features highlights the importance of feature generation.

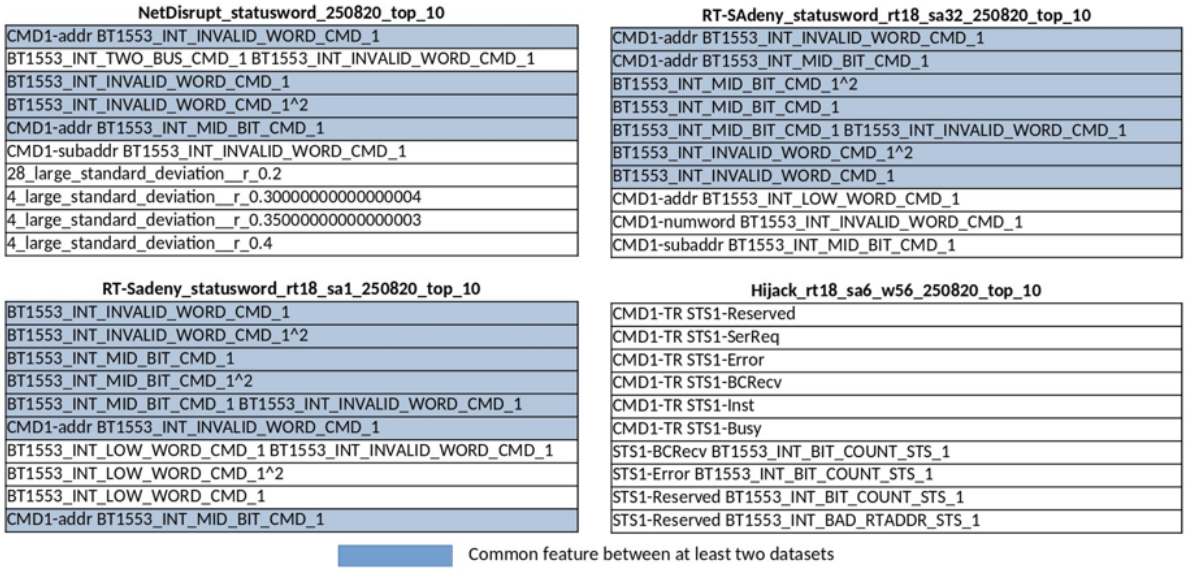


Figure 7: Top 10 Features Selected by ANOVA

6. Discussion

The resulting performance metrics of the fifteen models were then compared to the research conducted by Harlow, Lachine and Roberge (2024) which demonstrated that there was an overall improvement in the effectiveness of the anomaly detection pipeline. These metrics included accuracy, precision, recall, AUROCC and MCC.

6.1 ANOVA

The ANOVA general model outperformed all other general models in the DoS type attack (*Disrupt*, *Deny*, *SA deny*, *SA deny 2*). The specific ANOVA models: *Disrupt* and *Hijack* outperformed all other specific models for these same attack types, as seen in Figure 6. The *Deny* and *Disrupt* specific models performed marginally better than the general model, although in the case of the *SA deny* dataset, the general model performed better by 4%. Furthermore, the *Deny*, *SA deny*, *SA deny 2*, and *disrupt* datasets utilizing the general model obtained similar results. This is attributed to the four datasets utilizing the DoS attack method, and as such would contain similar style traffic. Due to this result, it shows that these four attacks could be combined into a single DoS model, allowing a more streamlined pipeline while still yielding effective results when compared to the model by Harlow, Lachine and Roberge (2024). In the case of the *Hijack* dataset metrics, the specific *Hijack* model significantly outperformed the general *Hijack* model.

6.2 FOS

FOS performed well for the specific models: *Deny* and *SA deny*, and performed relatively poorly for the *Disrupt* and *Hijack* models. FOS also performed poorly for all of the general models as seen in Figure 6. Additionally, the remainder of the FOS model's metrics demonstrated poor performance. The feature scores from FOS resulted in an elbow curve that typically suggested only one feature, whereas the majority of other feature selection methods suggested 5 or more. Due to the elbow method only suggesting a single feature, it is suggested that other methods be explored for selecting the cutoff for the number of features. Except for precision, the metrics from the *Disrupt* dataset demonstrated similar performance between the general and specific models, although with poor performance. Finally, with the metrics from the *Hijack* dataset, the general model outperformed the specific model, although again with overall poor performance. These results highlight the possible need for more anomalous traffic to be used for selection of the features.

6.3 PPS

PPS related performance metrics did not perform well compared to ANOVA and PPS. Although on closer analysis of the specific PPS models, three distinct grouping of a larger Mean Absolute Error (MAE) value were easily

identified. Therefore, reducing the MAE threshold for anomalous traffic in the model or utilizing other methods such as standard deviation could enable effective detection for the PPS models.

7. Conclusion

This research extended the original work by Harlow, Lachine and Roberge (2024) by focusing on feature generation and selection. The approach in this work resulted in fifteen distinct models based on the original design, results were calculated using the following performance metrics: accuracy, precision, recall, AUROCC and MCC. These metrics demonstrated the improved performance solely from this focus on feature engineering. The comparison of the results revealed there was a marked improvement in 8 of the 15 models created during this research. In addition, the improved results can be achieved with fewer features that in turn reduces overall processing time. These results highlight the promise of this improved model and the future work needed to develop an in-production solution for military aircraft.

7.1 Future Work

The next phase of this research will focus on creating anomalous datasets with an increased number and variety of attacks that aim to provide three related improvements. First, a more balanced dataset will improve the feature selection phase by reducing potential bias and generating a clearer decision boundary. Second, increased diversity in attacks will explore the ability to create more general models at the outset, reducing overall processing overhead and set the conditions for applying this model to different modes of flight. Third, a more balanced dataset would provide additional events to be categorized as anomalous, allowing for a more comprehensive validation of the detection methodology.

References

Abaco-Systems (2019) "Software Reference Manual BusTools/1553-API", Publication No. 1500-038 Rev. 5.12.

Abujazoh, M., Al-Darras, D., Hamad, N.A., Al-Sharaeh, S. (2023), "Feature Selection for High-Dimensional Imbalanced Malware Data Using Filter and Wrapper Selection Methods", International Conference on Information Technology (ICIT), pp. 196–201, https://doi.org/10.1109/ICIT58056.2023.10226049.

Banks, J., Kerr, R., Ding, S., Zulkernine, M. (2022), "SV1DUR: A Real-Time MIL-STD-1553 Bus Simulator with Flight Subsystems for Cyber-Attack Modeling and Assessments", 2022 IEEE Military Communications Conference (MILCOM), pp. 522–528, https://doi.org/10.1109/MILCOM55135.2022.10017663.

Bedard, C. (2019), "An Application of Network Security Monitoring to the MIL-STD-1553B Data Bus", MASc Thesis, Royal Military College of Canada, https://espace.rmc.ca/jspui/handle/11264/1823.

Brownlee, J. (2020), *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python*", Machine Learning Mastery, 1st ed.

Christ, M., Braun, N., Neuffer, J., Kempa-Liehr, A.W. (2018) "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)", Neurocomputing, 307, pp. 72–77, https://doi.org/10.1016/j.neucom.2018.03.067.

Da Silva Rodrigues, E., Martins, D.M.L. and Buarque de Lima Neto, F. (2021), "Automatic Feature Engineering Using Self-Organizing Maps", IEEE Latin American Conference on Computational Intelligence (LA-CCI), pp. 1–6, https://doi.org/10.1109/LA-CCI48322.2021.9769788.

De Santo, D., Malavenda, C.S., Romano, S.P., Vecchio, C. (2021) "Exploiting the MIL-STD-1553 avionic data bus with an active cyber device", Computers & Security, 100, p. 102097, https://doi.org/10.1016/j.cose.2020.102097.

Demertzis, K., Tsiknas, K., Takezis, D., Skianis, C. and Iliadis, L. (2021), "Darknet Traffic Big-Data Analysis and Network Management for Real-Time Automating of the Malicious Intent Detection Process by a Weight Agnostic Neural Networks Framework", Electronics, 10(7), p. 781, https://doi.org/10.3390/electronics10070781.

El Aboudi, N. and Benhlima, L. (2016), "Review on wrapper feature selection approaches", International Conference on Engineering & MIS (ICEMIS), pp. 1–5, https://doi.org/10.1109/ICEMIS.2016.7745366.

Elgendy, M. (2020) *Deep Learning for Vision Systems,* Manning, New York.

Généreux, S.J.J., Lai, A., Fowles, C., Roberge, V., Vigeant, G.P.M., Paquet, J. (2020), "MAIDENS: MIL-STD-1553 Anomaly-Based Intrusion Detection System Using Time-Based Histogram Comparison", IEEE Transactions on Aerospace and Electronic Systems, 56(1), pp. 276–284, https://doi.org/10.1109/TAES.2019.2914519.

Guyon, I. and Elisseeff, A. (2003) "An introduction to variable and feature selection", The Journal of Machine Learning Research 3, pp. 1157–1182, https://dl.acm.org/doi/10.5555/944919.944968.

Harlow, A., Lachine, B. and Roberge, V. (2024), "Anomaly Detection for the MIL-STD-1553B Multiplex Data Bus Using an LSTM Autoencoder", 19th International Conference on Cyber Warfare and Security, to be published.

He, D, Liu, X., Zheng, J., Chan, S., Zhu, S., Min, W. and Guizani, N. (2020), "'A Lightweight and Intelligent PPSIntrusion Detection System for Integrated Electronic Systems", IEEE Network, 34(4), pp. 173–179, https://doi.org/10.1109/MNET.001.1900480.

Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J. (2019), "Survey of intrusion detection systems: techniques, datasets and challenges", Cybersecurity 2, Article 20, https://doi.org/10.1186/s42400-019-0038-7.

Korenberg, M.J. and Paarmann, L.D. (1989), "Applications of fast orthogonal search: Time-series analysis and resolution of signals in noise", Annals of Biomedical Engineering, 17(3), pp. 219–231, https://doi.org/10.1007/BF02368043.

Levy, E., Maman, N., Shabtai, A., and Elovici, Y. (2022), "AnoMili: Spoofing Prevention and Explainable Anomaly Detection for the 1553 Military Avionic Bus", https://doi.org/10.48550/arXiv.2202.06870.

McGaughey, D., Semeniuk, T., Smith, R. and Knight, S. (2018), "A systematic approach of feature selection for encrypted network traffic classification", IEEE International Systems Conference (SysCon), pp. 1–8, https://doi.org/10.1109/SYSCON.2018.8369567.

Mukkamala, S. and Sung, A.H. (2002), "Identifying key features for intrusion detection using neural networks", in Proceedings of the 15th international conference on Computer communication (ICCC), pp. 1132–1138.

Muthukrishnan, R. and Rohini, R. (2016), "LASSO: A feature selection technique in predictive modeling for machine learning", IEEE International Conference on Advances in Computer Applications (ICACA), pp. 18–20, https://doi.org/10.1109/ICACA.2016.7887916.

Onodueze, F. and Josyula, D. (2020), "Anomaly Detection on MIL-STD-1553 Dataset using Machine Learning Algorithms", IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 592–598, https://doi.org/10.1109/TrustCom50675.2020.00084.

Paquet J. (2014) "Uncovering MIL-STD-1553 vulnerabilities: exploitability of military aircraft networks", MASc Thesis, Royal Military College of Canada.

Stan, O. et al. (2019), "On the Security of MIL-STD-1553 Communication Bus", in B. Hamid et al. (eds) Security and Safety Interplay of Intelligent Software Systems. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 153–171, https://doi.org/10.1007/978-3-030-16874-2_11.

Stan, O. et al. (2020) 'Intrusion Detection System for the MIL-STD-1553 Communication Bus', IEEE Transactions on Aerospace and Electronic Systems, 56(4), pp. 3010–3027, https://doi.org/10.1109/TAES.2019.2961824.

Thakkar, A. and Lohiya, R. (2023), "Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System", Information Fusion, 90, pp. 353–363, https://doi.org/10.1016/j.inffus.2022.09.026.

U. S. DoD (1978), "MIL-STD-1553B, Aircraft Internal Time Division Command/Response Multiplex Data Bus", https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=36973

Wetschoreck, F. (2020), "RIP correlation. Introducing the Predictive Power Score", *towardsdatascience.com*, https://towardsdatascience.com/rip-correlation-introducing-the-predictive-power-score-3d90808b9598.

Wrana, M.M., Elsayed, M., Lounis, K., Mansour, Z., Ding, S. and Zulkernine, M. (2022), "OD1NF1ST: True Skip Intrusion Detection and Avionics Network Cyber-attack Simulation", ACM Transactions on Cyber-Physical Systems, 6(4), p. 33:1-33:27, https://doi.org/10.1145/3551893.

Zander, S. and Williams, N. (2011) "netAI - Network Traffic based Application Identification", http://caia.swin.edu.au/urp/dstc/netai/.