University of Essex
Department of Mathematical Sciences

MA981: DISSERTATION

Spacecraft Anomaly Detection with Conditional GANs and Explainable AI

Sukanya Arundhati Das

2321248

Supervisor: Dr. Alex Diana

January 5, 2025
Colchester

Contents

List of Figures

List of Tables

Abstract

Anomalies detection in spacecraft is considered crucial in taking safety measures and minimising the risk of mission failures. Most of the time, the imbalanced and high-dimensional nature of spacecraft telemetry data makes the common anomaly detection methods quite inefficient in effectively detecting anomalous data. This research focuses on how Conditional Generative Adversarial Network (CGAN)-generated synthetic anomalous data enhances anomaly detection in spacecraft telemetry using the majority voting concept and how Explainable AI (XAI) can provide insights for anomaly predictions. This study is explored using NASA's two spacecraft telemetry data, where the models are trained and evaluated to detect anomalies in spacecraft operations. The performance of the real data is compared by integrating real data with CGAN synthetic anomalies. Afterwards, anomalies are detected using a majority voting mechanism, where multiple models are trained to classify data points as anomalous or normal. Thereafter, the SHAP (XAI) technique is used to identify the relevant features contributing to anomaly detection to improve interpretability and trust. The proposed methods achieve a high recall with moderate AUC after the integration of CGAN anomalies and show the framework's ability to detect anomalous patterns in spacecraft telemetry, thereby overcoming the lack of labelled data concern. However, the lower precision and F1 score indicate the limitations of reducing false positive rates, which remain an area for further improvement.

Keywords: CGAN, XAI, SHAP, Sliding window, Z-Score, Majority Voting, Synthetic Anomalies

CHAPTER 1

Introduction

Spacecraft systems are highly sensitive and complex in nature. The space system is often impacted due to sudden fluctuations in temperature, energy usage, and inconsistencies in other spacecraft channels, thereby increasing the risk of system failure. These variations in spacecraft sensors combined with the complexity of the systems and remote operation make anomalies quite certain. As mentioned by Hundman et al. [11], to overcome the challenges, real-time status monitoring and tracking of these channels are essential for successful spacecraft missions, considering the high complexity and operational cost. Thus, anomaly or any fault detection system in spacecraft is considered to be crucial for providing real-time insights and automated alerts.

The spacecraft telemetry data are collected from respective channels to evaluate system health and identify potential issues. Any deviation in the telemetry data observed while collecting the measurements is marked as an outlier or anomaly. Generally, the telemetry data contains numerous channel measurements that are voluminous, imbalanced, and diverse in nature, making it difficult for analysis. These data exhibit high dimensionality along with noisy and heterogeneous signals (Hundman et al. [11]; Raturi et al. [18]). In the real-time scenario, the limited availability of labelled telemetry anomalies restricts the use of supervised learning practices, thereby adopting an unsupervised approach, as suggested by Hundman et al. [11].

According to previous studies, anomalies are considered to be of three types, such as point anomalies, contextual anomalies, and collective anomalies (Chandola, Banerjee, and Kumar [5]; Farias et al. [9]). However, in this study the telemetry anomalies are considered as a whole instead of differentiating the anomalies based on specific types. The sole purpose of this research is to detect any abnormal patterns or deviations in the spacecraft operation without adding further complexity based on the anomaly types.

In this paper, the real telemetry data from NASA's two spacecraft, SMAP and MSL, is explored using statistical analysis and visualisation techniques. A hybrid approach is taken into consideration by using the CGAN model, the majority voting concept, and SHAP analysis to minimise the challenges of the unavailability of labelled data for the training set. The primary focus is to assess the effectiveness of CGAN synthetic anomalies by combining them with the real normal telemetry data for training purposes.

Basically, CGANs are a type of generative adversarial neural network (GAN), where GANs emerge from the concept of generative AI, which is widely known as Gen AI. Generative AI comes under artificial intelligence and consists of several methods, including GANs and variational autoencoders (Kumar and Sharma [13]). Generally, GAN architecture represents the generator and the discriminator neural networks, and both the networks contribute to producing more realistic synthetic/fake data. These networks compete in a process in which the generator mimics the real data to generate more accurate fake data, and the discriminator differentiates between actual data and generated synthetic data. Followed by the same concept of GANs, CGANs are a more enhanced type of GANs in which class labels or conditions are taken into account while generating new data.

The proposed method aims to use CGAN anomalies to mitigate the issue of labelled data for anomaly detection tasks and then fit the model using the majority voting concept. The voting concept is used to combine the prediction of multiple models into a single output and differentiate the data point as normal or anomalous. Three models are considered for this purpose, namely isolation forest, one-class SVM, and LSTM autoencoder. As per El-Enen et al. [8], the majority voting concept plays a vital role when there's a lack of labelled data and usually combines multiple unsupervised models to achieve an effective anomaly detection system. Apart from this, explainable AI (XAI) is implemented to provide an overview of how

well the features contributed to the anomaly detection process.

CHAPTER 2

Related Work

The anomaly detection in the space domain is quite crucial for spacecraft mission safety and success. However, common anomaly detection techniques often face challenges due to highly imbalanced classes, the absence of labelled anomalies, and limited explanation for certain irregular patterns. To overcome these limitations, this paper explores the extent to which CGANs, combined with a majority voting strategy, contribute to the anomaly detection process. Additionally, the study emphasises using XAI techniques like SHAP to enhance the explainability of features and make the detection process more transparent, reliable, and actionable.

2.1 LSTMs for Spacecraft Telemetry

Hundman et al. [11] addressed the issues in anomaly detection by proposing an unsupervised nonparametric framework that integrates LSTM models with dynamic thresholding, using the NASA dataset. This approach effectively learns temporal patterns to determine whether the present value is impacted by previous value patterns in spacecraft telemetry data and allows smooth anomaly detection. Their strategy, which combines non-parametric thresholding and pruning, showed improved precision by reducing false positives while impacting recall. Despite the approach's effectiveness, the study shows persistent challenges, including a high

false-positive rate. Also, the lack of explainability in the LSTM model remains a challenge for mission operators to understand the detected anomalies.

2.2 Supervised Learning using Modified Test Data

A research by Farias et al. [9] considered supervised learning for spacecraft anomaly detection by modifying the labelled test dataset from the NASA public repository. Their approach uses random forest models to identify anomalies and LIME to provide explanations for individual predictions. Further, the approach resulted in better performance compared to related works. However, the paper completely relies on the labelled datasets, which can be challenging in real-world scenarios. Thus, the current study uses a hybrid approach by leveraging CGAN, a majority voting strategy, and SHAP analysis to address the issues associated with labelled datasets. Furthermore, the feature extraction approach is also influenced by this paper, with minor modifications in the extraction process that will be discussed in the following chapter.

2.3 Generative AI Models for Anomaly Detection

Generative AI models are highly efficient at generating diverse and high-quality output by learning patterns from input data and producing new data patterns that resemble the characteristics of the input data (Kumar and Sharma [13]). The most common Gen AI models are GANs, which are used in the anomaly detection process to learn complex and high-dimensional data patterns. Moreover, recent studies and experiments with GAN-based architecture have shown considerable potential to identify anomalous patterns in time-series data.

The TadGAN, an unsupervised architecture proposed by Geiger et al. [10], shows potential by integrating LSTM layers within the GAN framework to capture temporal and key patterns in time-series data. The anomalies are detected by evaluating the reconstruction errors, providing reasonable metrics on multiple time series datasets, including the NASA spacecraft dataset. However, the study highlights that highly accurate reconstructions can overfit

anomalies and make it difficult to identify anomalous patterns. This insight highlights the significance of further research to find a balance between reconstruction and effective anomaly detection.

In another study, Song et al. [23] described a novel approach to identify anomalies in spacecraft telemetry data based on GANs. According to the research, this method is more effective than traditional methods, as the approach uses a GAN-based anomaly detection method, where LSTM networks are used to capture temporal patterns and reconstruct time-series data. The anomaly score, known as GDScore, is determined by the generator's reconstruction error and the discriminator's output. The approach is assessed on NASA's telemetry datasets, and it performed well compared to common unsupervised techniques. However, the paper highly relies on the threshold for the anomaly detection outcomes and does not suggest a systematic approach to determine the optimal threshold for the datasets. This study further encourages experimenting with CGANs to generate a wide range of anomalies and expose the model to different patterns in order to minimise the threshold dependency.

A different study Raturi et al. [18] uses GANs to generate synthetic data that replicates normal patterns and a binary classifier to detect anomalies. This method achieved high metrics, including F1 score, precision, and recall, on the Yahoo! Webscope S5 time series dataset. In addition, the study highlights the potential of GANs in enhancing datasets with minimal data or imbalanced classes and suggests further study with different datasets and scenarios to expand the GAN framework. Based on this concept, the CGAN framework is taken into consideration in our research with a diverse NASA dataset to generate synthetic anomalies for spacecraft telemetry.

Furthermore, GANomaly, another research conducted by Akcay, Atapour-Abarghouei, and Breckon [2], is a semi-supervised anomaly detection framework that uses an encoder-decoder-encoder pipeline leveraging a CGAN approach. In this approach, the model learns normal data distribution during training and detects anomalies by measuring reconstruction distances in image and latent spaces. Further, it uses adversarial, contextual, and encoder losses to generate realistic outputs and minimise reconstruction discrepancies. Despite achieving high accuracy, GANomaly exhibits limitations in detecting anomalies with high similarity to normal data in the UBA datasets and struggles with minor deviations and overfitting.

Furthermore, the model lacks interpretability and explanations for its anomaly scores. Based on the study of GANomaly, our research extends the same generator-discriminator concept using the CGAN framework along with associated losses to generate realistic and diverse synthetic anomalies for spacecraft telemetry.

In addition, Vega-Márquez et al. [24] propose the use of CGANs to generate synthetic data using class labels for a real-world credit card fraud dataset. The generated data from CGAN showed minor correlation with the original dataset while achieving similar performance on a classification task using the XGBoost algorithm. The outcome suggests that synthetic data generated by CGAN for tasks like fraud detection acts as a suitable approach to train machine learning models, thus maintaining data privacy. Moreover, this paper suggests further exploration of GAN architectures and parameter tuning to improve synthetic data quality and reliability.

2.4 Anomaly Detection Using Ensemble-Based Methods

The majority voting technique is an ensemble approach that combines predictions from multiple models to provide a single output based on majority vote and reduce dependency on individual models.

The approach suggested by El-Enen et al. [8] for fraud detection in medical insurance claims used a majority voting concept for 18 unsupervised anomaly detection models with no labelled data. The study shows the ability of majority voting by combining unsupervised models and achieved a high accuracy. Thus, the study suggests that this voting technique is capable of dealing with high-dimensional and complex data patterns, which makes it appropriate for our research to detect anomalies in spacecraft telemetry data.

Further, Mateo et al. [16] developed a framework to enhance the efficiency of an expert system for predictive maintenance in the packaging industry. This framework uses unsupervised anomaly detection methods that include One-Class SVM (OCSVM) and Minimum Covariance Determinant (MCD) to enhance the outputs of a Random Forest-based predictive maintenance system. This method is applied to time-series data, where the voting ensemble method

achieved the best performance in terms of F1 score compared to individual classifiers and shows the potential of combining unsupervised models. Hence, the same voting approach is applied for our research by integrating models like Isolation Forest, One-Class SVM, and LSTM to enhance anomaly detection processes.

Another research by Agyemang [1] compared five unsupervised algorithms to identify anomalies. The models used for this purpose are one-class SVM, isolation forest, local outlier factor, robust covariance, and one-class SVM with SGD. The study used a synthetically generated dataset to assess each model's performance in outlier detection. The models, like one-class SVM, isolation forest, and robust covariance, are effective, and isolation forest achieved a balanced F1 score. Further, one-class SVM with SGD resulted in high precision but with a low recall value. These results verify the efficacy of models like isolation forest and one-class SVM, which are also used as part of our ensemble framework in detecting anomalous patterns. In addition, the study also suggests validating the effectiveness of these methods on real-world datasets. Thus, the methods are considered as part of our research and applied to the NASA spacecraft telemetry dataset.

2.5 Explainable AI in Anomaly Detection

Generally, machine learning models that are highly complicated and efficient often act as black boxes, and this makes it challenging to interpret the predictions done by these models as stated by Li, Zhu, and Van Leeuwen [14]. Hence, the lack of transparency and explainability often reduces trust and reliability in anomaly detection systems, resulting in delayed actions on certain irregular patterns, mainly in major sectors like space. To overcome these issues, Explainable AI (XAI) is taken into consideration, which allows users to understand the reason behind the occurrence of anomalies. This helps the operators to take corrective measures on time, thereby reducing the chance of any major incidents.

Furthermore, a review discussed by Kumar and Sharma [13] shows the requirement of transparency and explainability for generative AI models in major sectors like healthcare and finance. Thus, this study shows the need for explanation in anomaly detection systems in order to use the models efficiently in real-world scenarios.

In addition, Farias et al. [9] suggest the significance of XAI in spacecraft anomaly detection, emphasising the need to understand specific data that are marked as anomalous. This study uses the Local Interpretable Model-Agnostic Explanations (LIME) framework to interpret predictions made by a random forest model for supervised anomaly detection, as previously discussed in this review. Basically, the work highlights the ability of XAI using the LIME framework in enhancing the spacecraft's safety. While this study validates the effectiveness of supervised learning approaches, it also shows a significant knowledge gap in unsupervised and generative frameworks, where feature-related explanations are equally necessary but remain mostly unexplored.

A survey conducted by Li, Zhu, and Van Leeuwen [14] provides a detailed classification of eXplainable Anomaly Detection (XAD) techniques, based on certain criteria such as locality (global & local explanations), data perspective (feature, sample, or feature & sample), specificity (model-agnostic & model-specific), and so on. The study also highlights the importance of post-model explanation techniques like SHAP (SHapley Additive exPlanations) for anomaly detection. Furthermore, SHAP provides feature-based insights by identifying key telemetry parameters or features that contribute to the anomalous behaviour.

Additionally, the research conducted by Roshan and Zafar [20] highlights the importance of kernel SHAP and refers to it as a model-agnostic XAI technique that can be used for any machine learning model. In their study, the SHAP technique is used to determine the most relevant feature contributing to the autoencoder's reconstruction error. The process involves using a subset of instances and then calculating SHAP values to determine feature relevance. However, the study suggests the time complexity of kernel SHAP with high-dimensional datasets and larger background sets. Thus, this study suggests considering a smaller subset to overcome performance issues, thereby using the SHAP XAI technique.

Overall, the previous studies and analysis highlight the benefits of using GANs to produce accurate and diverse fake data for spacecraft telemetry. These synthetic data can handle the issues of imbalanced datasets and enhance the performance of the anomaly detection framework. Furthermore, in real-world scenarios, obtaining labelled data is often challenging, which limits the feasibility of using supervised learning approaches for anomaly detection. This highlights the necessity of using more robust techniques in the current research that can

detect and handle anomalies in unsupervised or hybrid approaches, such as CGANs, rather than relying largely on labelled data. Moreover, relying on a single model to detect anomalies increases the risk of misclassification. To avoid these challenges, this research proposes using a majority voting ensemble approach that can handle complex and high-dimensional data well. This approach will combine the outcomes of various models like Isolation Forest, One-Class SVM, and LSTM AE to ensure robustness and reliability.

On the other hand, a major gap in the existing studies is the limited focus on explainability frameworks for anomaly detection, mainly in generative and unsupervised models. Though LIME is useful in improving the transparency within supervised learning systems, their integration with GANs and unsupervised anomaly detection frameworks remains unexplored. This research addresses this gap by applying SHAP to the predictions of a majority voting ensemble, including CGAN anomalies alongside real dataset anomalies, to provide feature-level explanations for the detected anomalies. This integration enhances the explainability of anomaly detection systems, building trust and providing actionable insights in safety-critical domains such as spacecraft telemetry.

CHAPTER 3

Dataset

In this paper, the dataset used is a publicly available NASA dataset shared on Kaggle (NASA Anomaly Detection Dataset on Kaggle). The dataset consists of separate folders for training and testing, along with a labeled.csv file that contains anomaly labels associated with the test dataset. In addition, the dataset includes three more folders: smoothed errors, models, and yhat. However, these folders are not utilised in this research, as the main focus is on the core components, training data, test data, and anomaly labels for the test set, which provide relevant data pre-processing and modelling processes for identifying anomalies in the telemetry data.

3.1 Data Description

The training and testing folders contain telemetry data with time-series values for individual channels stored in .npy files, representing time-series data recorded from the spacecraft's subsystems. The labeled.csv file provides details of channels and corresponding anomaly sequences over time.

Channel ID (chan_id): Maps to specific .npy files in train and test sets.

Spacecraft Names (spacecraft): Soil Moisture Active Passive (SMAP) satellite and the Mars Science Laboratory (MSL) rover, Curiosity.

Anomaly Interval (anomaly_sequences): Start and end time of each anomaly.

Anomaly Category (class): Point, contextual, or collective anomalies.

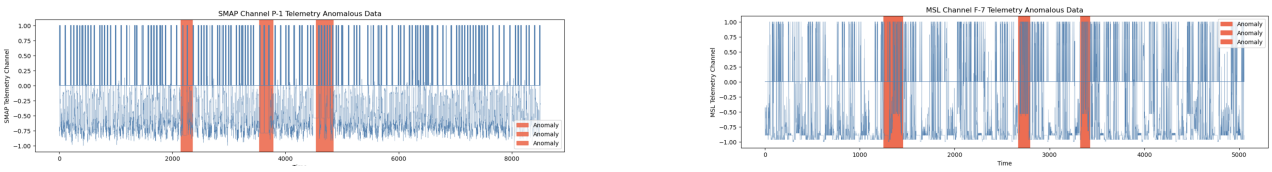
Telemetry Values (num_values): number of telemetry values in each .npy file of the test set.

The SMAP and MSL spacecraft consist of complex monitoring systems, with numerous channels tracking various subsystems and anomalous patterns. SMAP includes 55 channels, and MSL contains 27 channels, where each channel represents measurements of different spacecraft subsystems and suggests potential anomalies.

3.2 Exploratory Data Analysis

3.2.1 Data visualization (Test Set)

In fig 3.1 (a), for SMAP Channel P-1, the dataset includes 8505 values with three anomalous sequences observed in the range of [2149, 2349], [3539, 3779], and [4536, 4844]. The plot shows anomalies in red as deviations from the normal pattern. For MSL Channel F-7 fig 3.1 (b), the dataset contains 5054 values, with anomalies recorded in the range of [1250, 1450], [2670, 2790], and [3325, 3425]. These anomalies are shown in the plot as extreme deviations and abnormal behaviour in the telemetry data.



(a) SMAP Channel P-1

(b) MSL Channel F-7

Figure 3.1: Test Set Visualization

3.2.2 Feature Extraction and Scaling

Sliding Window Technique

Kulanuwat et al. [12] suggest the concept of a sliding window to capture abnormal data patterns within a specific time frame or window, and the similar sliding window approach is applied in this paper. Generally, the sliding window technique is used to divide the time-series data into overlapping or non-overlapping segments. These segments are commonly known as windows, where each window captures a small subset of data points. This technique focuses on key patterns rather than the entire dataset, as anomalies are difficult to identify across the whole dataset. The smaller segments make it easier to detect anomalies and retain key patterns of the data. The window size specifies the data points per segment, and the step size specifies how far the window moves forward after each iteration. For this study, a window size of 100 and a step size of 50 are considered. Each window moves forward by 50 data points, capturing an overlapping segment of 100 data points at a time. Each extracted window is considered an independent input, which is then utilised for subsequent processing steps. Specifically, in the context of the CGAN framework and Long Short-Term Memory (LSTM) network, to a single timestep, i.e., timestep = 1.

Feature Extraction

Within each window, relevant features are extracted to capture the specific behaviour of the data. The features extracted are the waveform changes, frequency fluctuations, and the variation in magnitude. Afterwards, these features are used to transform raw telemetry data into a structured format for analysis that is suitable for machine learning models.

Feature 1 (Waveform Changes)

The initial feature is calculated to capture the waveform change and calculates the prediction error by comparing the real telemetry values with predicted ones using a moving average predictor and a moving average window size of 5. Afterwards, the standard deviation of these errors is considered to be the waveform changes over time. Farias et al. [9] considers

the raw prediction error as the feature. However, there's a slight change in this paper, where the standard deviation is used to provide a more robust measure of waveform outliers by summarising the prediction errors rather than relying on individual errors. According to Farias et al. [9], the equations can be calculated as:

Moving Average Prediction:

$$\hat{y}_{ma}(t) = \frac{1}{n}\sum_{i=0}^{n-1} x(t-i) \tag{3.2.1}$$

Where,

- $\hat{y}_{ma}(t)$: Predicted sample at t
- x: Series value
- t: Time Period
- n: Window size

Prediction Error:

$$e_{ma}(t) = x(t+1) - \hat{y}_{ma}(t) \tag{3.2.2}$$

Where,

- $e_{ma}(t)$: Prediction error at t
- $x(t+1)$: Actual value at the next time step

Finally,

$$Feature_1 = \text{std}(e_{ma}(t)) \tag{3.2.3}$$

Feature 2 (Frequency Changes)

The second feature, frequency, is determined by aggregating the frequency components derived from the Short-Time Fourier Transform and calculating their standard deviation over time. This approach aligns with the work done by Farias et al. [9], and the equation is given by:

$$Feature_2 = \text{std}\left(\sum_f |x(f,t)|\right) \tag{3.2.4}$$

Where,

- $|x(f,t)|$: STFT component for frequencies

Feature 3 (Magnitude Changes)

Furthermore, the third feature, magnitude, identifies anomalies in the magnitude by initially considering the prediction error using a moving average predictor and then smoothing the error using Exponentially Weighted Moving Average (EWMA) with a smoothing factor of 0.3 as performed by Farias et al. [9], and again this concept is also considered based on the analysis performed by Raval [19]. This approach replaces the use of smoothed LSTM predictor error done by Farias et al. [9] with smoothed moving average predictor error, i.e., $e_{ma}(t)$, and is adopted to avoid the complexity of the LSTM predictor, which is complex and time-consuming.

In the sliding window and feature extraction process, the training set is treated as normal data, and anomaly labels are not mapped for the training set, whereas the test set is mapped with the labelled anomalies. These labelled anomalies provide the ground truth to measure the performance of the model in identifying anomalous behaviour.

Scaling

Min-Max scaling is used on the training data to ensure feature consistency and compatibility, thereby normalising all features to a uniform range of 0 to 1. The same scaling parameters derived from the training data are used for the test set in order to prevent data leakage and ensure the integrity of the evaluation process.

3.2.3 Data visualization (Train Set)

SMAP Train Data

Scatter Plot

The scatter plot in fig. 3.2 shows the correlation between waveform and frequency and the magnitude as a colour gradient. The majority of the points are clustered and show clear patterns in the telemetry data, where the magnitude gradually increases from low to high. Also, the spread of the data points indicates variation in the feature interactions. Thus, the plot provides an overview of feature relationships with some outliers in the data, which can help with the analysis in the following steps.

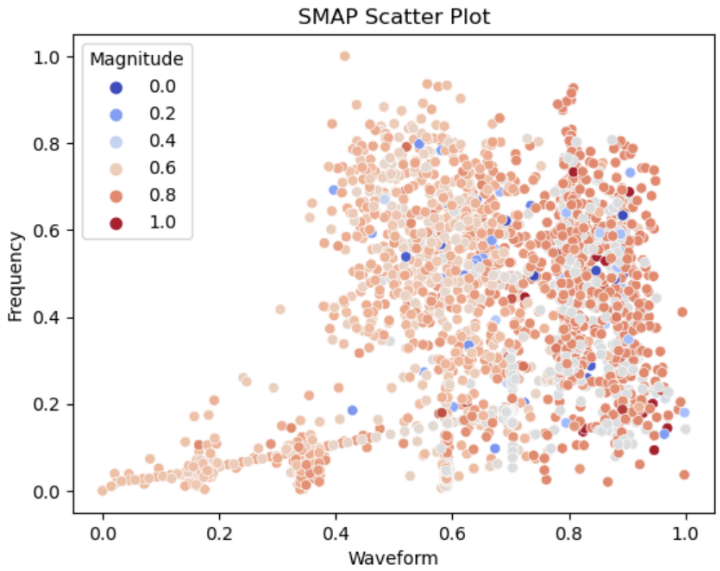


Figure 3.2: SMAP Scatter Plot

Again, to validate the findings, additional visualisation techniques are used to identify potential outliers or anomalous patterns in the dataset.

Box Plot

The box plot in fig. 3.3 shows most of the outliers for waveform below the lower range and few outliers for magnitude in the lower part. These outliers in waveform and magnitude suggest the occurrence of anomalies in the dataset. For frequency, no notable outliers are observed, and it also shows a wide range with more variation than the other features. Moreover, this plot shows better visualisation of feature variation along with the outliers, which is required to verify prior to the outlier removal procedure.

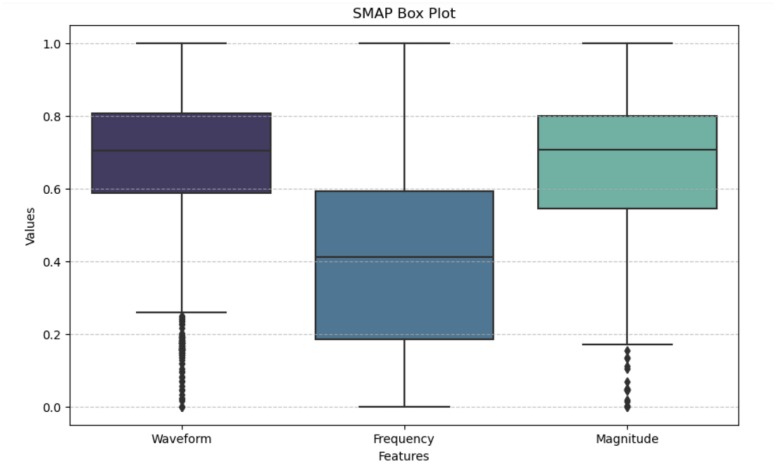


Figure 3.3: SMAP Box Plot

Hence, the box plot indicates that the dataset is well-prepared and scaled for further research. Additionally, a time series plot is also taken into consideration in the following step to observe the fluctuation in features across time.

Time-Series Plot

The time series plot in fig. 3.4 shows that each feature acts differently and shows different patterns along with fluctuations across time. The magnitude shows high values with a

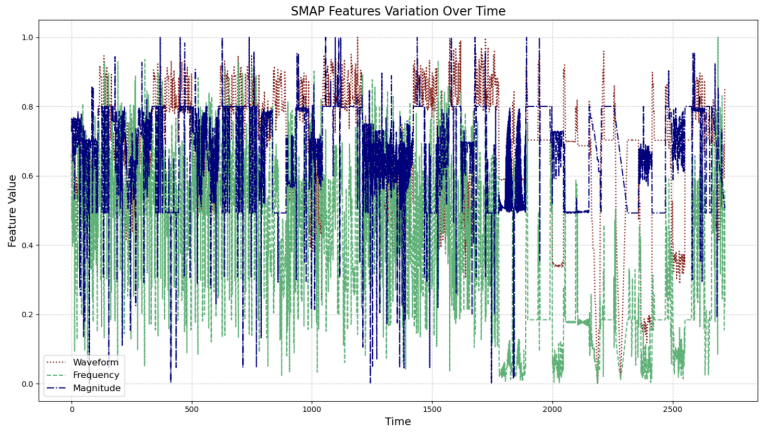


Figure 3.4: SMAP Time-Series Plot

consistent pattern with some variations and occasional drops. Again, the frequency varies significantly over time and shows more variations than the other two features. The waveform shows an inconsistent pattern with sharp peaks and shows shifts across time. Hence, the plot

highlights the temporal dependencies and fluctuations throughout time, which makes the dataset relevant for further analysis in anomaly detection tasks.

In general, the variation of features suggests that the dataset contains relevant information to distinguish between normal and anomalous patterns. Moreover, the features, waveform, frequency, and magnitude are scaled across the entire dataset and well-suited for subsequent steps.

MSL Train Data

Scatter Plot

The scatter plot in fig 3.5 shows that the data points are mostly clustered and packed in the lower region of the plot with smaller values for both waveform and frequency. As shown in the plot, some points that deviate from this pattern indicate outliers or anomalies within the dataset. Also, some data points show a higher value of waveform or frequency, often

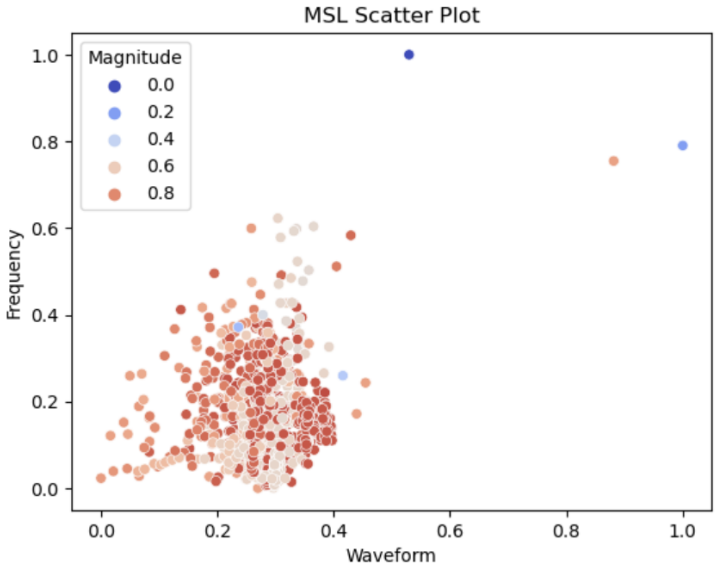


Figure 3.5: MSL Scatter Plot

associated with a consistent high magnitude. Again, additional visualisations are used to verify the dataset for anomalies or abnormal patterns.

Box Plot

The box plot in fig. 3.6 shows significant outliers for the waveform above the upper whisker

and below the lower whisker. Similarly, frequency shows outliers above the upper region. All three features exhibit these outliers, with the waveform and frequency pointing to the most extreme values. The magnitude covers a wider range and variability across the dataset. Overall, this plot shows the features relations and the presence of outliers, which is required for the subsequent steps.

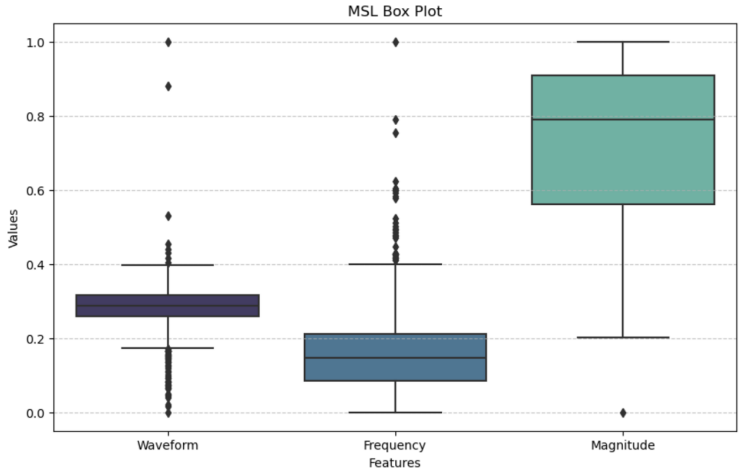


Figure 3.6: MSL Box Plot

Time-Series Plot

The time series plot in fig 3.7 shows the fluctuations in waveform, frequency, and magnitude over time. The waveform shows relatively consistent patterns with occasional peaks, and

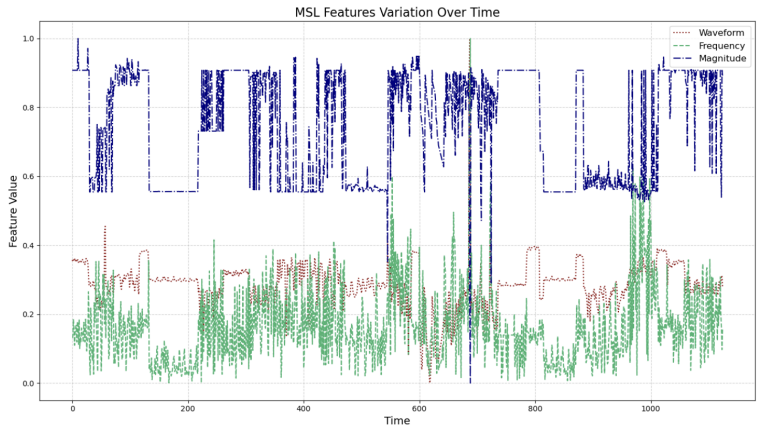


Figure 3.7: MSL Time-Series Plot

frequency shows frequent variations. Further, magnitude has a stable pattern for longer time durations with few sudden peaks and indicates that the data might contain fewer deviations.

This variation across features suggests that the dataset is appropriate for additional detection tasks in subsequent steps.

In general, both the SMAP and MSL datasets show some variations and deviations over time, along with some irregular data patterns that might refer to an anomaly or outlier. Hence, these observed outliers and irregular patterns are further statistically analysed using the Z-score technique in the subsequent phases.

3.2.4 Statistical Analysis

Z-Score Technique

This paper uses a statistical approach, i.e., Z-score analysis, to verify the findings obtained in the above sections using data visualisation techniques. As per Ramesh Kumar [17], the Z-score is a widely used statistical technique in the anomaly detection process that measures the distance of a data point from the mean with respect to standard deviations. For this purpose, a threshold is set, where any point beyond that threshold is marked as an anomaly. In this study, a threshold value of 3 is considered to determine the Z-score of the individual features like waveform, frequency, and magnitude independently.

According to Chikodili et al. [7], the Z-score is given by:

$$Z(i) = \frac{x_i - \mu}{\sigma}$$

Where,

- x_i: Data point in the distribution
- μ: Mean of the distribution
- σ: Standard deviation of the distribution

SMAP Train Data Analysis

In SMAP training data, as shown in tab 3.1, all the extracted features, waveform, frequency, and magnitude consist of zero missing values. waveform and magnitude show higher mean values compared to frequency and suggest variation in the behaviour across features. The standard deviations are low and show most values are close to their respective means. The

SMAP Train Data			
Parameters	**Waveform**	**Frequency**	**Magnitude**
Mean	0.678	0.398	0.671
Standard deviation	0.179	0.230	0.143
Skewness	-0.97	0.04	-1.02
Kurtosis	0.93	-1.18	1.64
Z-Score	22	0	36

Table 3.1: Statistical Analysis of SMAP Train Data

waveform and magnitude are slightly negatively skewed and suggest that most of the data points are quite high, with only a few deviating to lower ranges. On the other hand, the frequency is nearly symmetrical and balanced. Further, the kurtosis value for magnitude indicates a distribution with heavier tails, meaning the distribution consists of more extreme data points than a normal distribution.

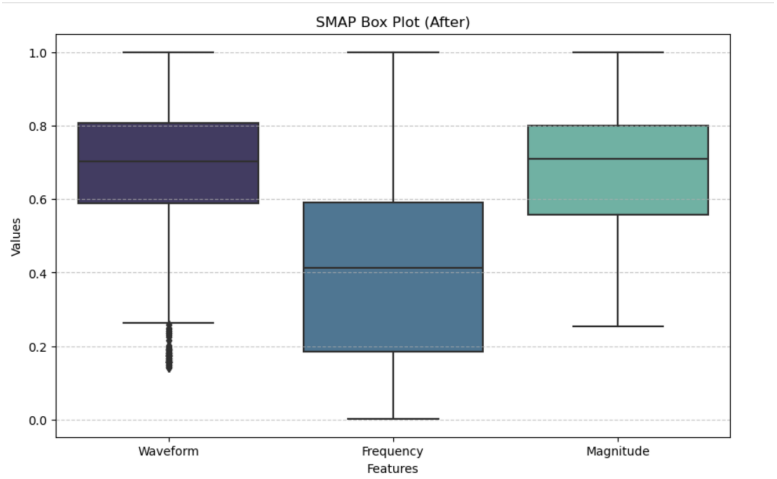


Figure 3.8: SMAP Box Plot (After Outlier Removal)

Additionally, Z-scores identify outliers in waveform and magnitude, but no outliers are detected in frequency. Again, these outliers are removed to enhance the data quality and reduce noise in the dataset. In fig. 3.8, a post-outlier removal box plot is used to represent a more compact range for waveform and magnitude. In general, the SMAP training data is well suited for the anomaly detection process based on the statistical analysis.

MSL Train Data Analysis

In MSL training data, as shown in tab 3.2, all the extracted features, waveform, frequency, and magnitude consist of zero missing values. Magnitude consists of a high mean value across the dataset. Again, all features show moderate standard deviations and suggest relatively close grouping around their means. Further, the waveform and frequency have positive skewness

MSL Train Data			
Parameters	**Waveform**	**Frequency**	**Magnitude**
Mean	0.285	0.164	0.743
Standard deviation	0.067	0.109	0.161
Skewness	0.95	1.58	-0.28
Kurtosis	17.94	5.43	-1.28
Z-Score	21	14	2

Table 3.2: Statistical Analysis of MSL Train Data

values and suggest right-tailed distributions. On the other hand, the magnitude suggests a left-tailed distribution with a negative skewness value. The high kurtosis of waveform and frequency indicates the presence of outliers within the dataset.

In addition, Z-score analysis is used to detect potential outliers in all the features. Afterwards, these outliers are removed to reduce noise and enhance data quality for further tasks. The updated box plot in fig 3.9 shows a more compact data range for the waveform and frequency. The magnitude remains to have a wider range, and after the removal of outliers, its distribution shows a few extreme values.

In general, both the SMAP and MSL datasets are cleaned and refined with a well-structured

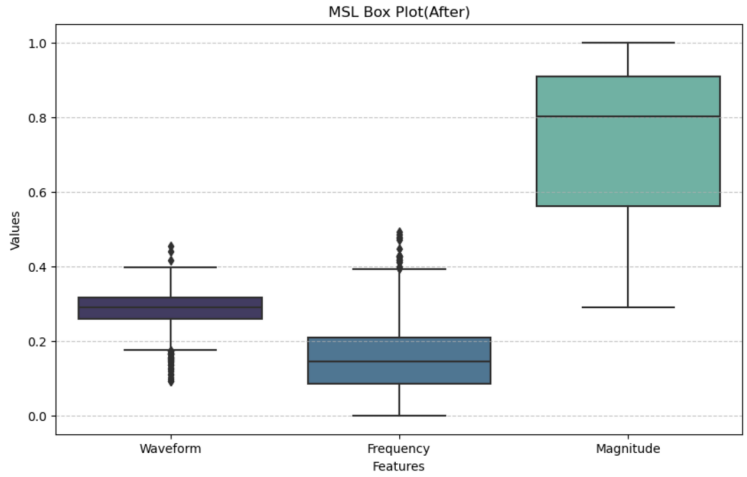


Figure 3.9: MSL Box Plot (After Outlier Removal)

distribution across their respective features and enhanced data quality. In addition, the datasets are ready for further anomaly detection and modelling tasks.

CHAPTER 4

Conditional GAN (CGAN) Framework

This study uses the conditional GAN framework to generate synthetic anomalies in order to enhance the anomaly detection process in space sectors. The concept of the framework is designed based on the approach described by Sevyeri and Fevens [22] and Brownlee [3]. The CGAN framework includes three key components, namely the generator, the discriminator, and the combined CGAN model, which are considered for time-series data.

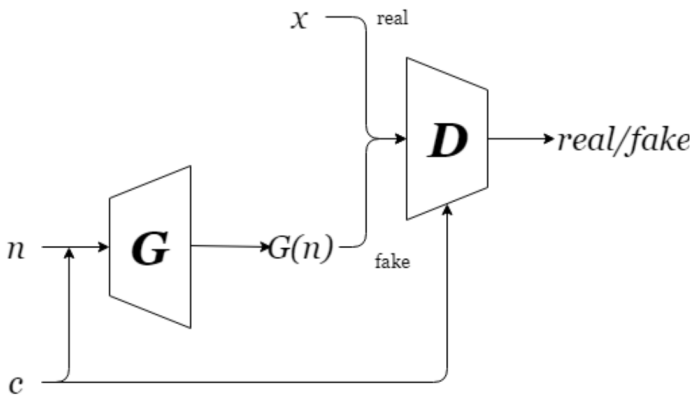


Figure 4.1: CGAN Framework by Vega-Márquez et al. [24]

4.1 Generator

The generator is used to generate fake irregular patterns based on noise and conditions as input. The input uses random noise that is generated based on the mean and standard deviation of the real normal data in order to deviate from the normal patterns. The concept of adding noise deviations using statistical attributes is inspired by the approach described by Schlegl et al. [21], where the analysis used GANs to identify anomalies using deviations from a learnt normal pattern. Again, the conditional labels are used to differentiate the normal and anomalous data. The generator's framework includes LSTM layers, which are used to learn the key and temporal patterns in the telemetry data. Additionally, dropout layers are added to reduce overfitting, and batch normalisation is considered to stabilise the training process. Also, leaky relu activation layers are used to avoid the problem of vanishing gradients. Apart from this, L2 regularisation is used in the LSTM layers to avoid overfitting and improve further stability. The output layer consists of a dense layer, and next to it, a reshape operation is added to match the generated sequence with the dimensions of the real telemetry data.

Parameter setting for the generator network:

Layer	**Details**
LSTM (Layer 1)	80 units; return sequences = True; L2 regularization = 0.00005
LeakyReLU (Layer 1)	Alpha = 0.2
Dropout (Layer 1)	0.3
LSTM (Layer 2)	80 units; return sequences = False; L2 regularization = 0.00005
LeakyReLU (Layer 2)	Alpha = 0.2
Dropout (Layer 2)	0.3

Table 4.1: CGAN Generator Network

4.2 Discriminator

The discriminator is used to differentiate the real and fake telemetry data based on the data input and the specified condition. The network takes both real and synthetic data as

input, along with their corresponding conditional labels. Further, the discriminator network includes two LSTM layers to extract temporal patterns and identify sequential patterns in the data. Leaky relu is used to allow the gradients to flow for negative inputs and improve the flow of information across the network. In addition, dropout layers are added to avoid overfitting, and batch normalisation is applied to ensure the stability of the data flow. For further network improvement, L2 regularisation is applied to maintain stability during the training. The output layer produces a single probability (0 to 1) for the input data, using a sigmoid activation.

Parameter setting for the discriminator network:

Layer	**Details**
LSTM (Layer 1)	80 units; return sequences = True; L2 regularization = 0.00005
LeakyReLU (Layer 1)	Alpha = 0.2
Dropout (Layer 1)	0.3
LSTM (Layer 2)	80 units; return sequences = False; L2 regularization = 0.00005
LeakyReLU (Layer 2)	Alpha = 0.2
Dropout (Layer 2)	0.3

Table 4.2: CGAN Discriminator Network

4.3 CGAN Model

The generator and discriminator are combined together to allow the entire training process. During this process, the generator tries to trick the discriminator, whereas the discriminator tries to correctly classify the inputs. To maintain balance in the training process, the discriminator is frozen during the generator's updates, which allows the generator to learn the data pattern without any interference. The binary cross-entropy loss function is used for the discriminator as applicable for the binary classification tasks. In addition, the mean squared error (MSE) loss function is utilised for the generator to reduce variations in data patterns, thereby enhancing its ability to trick the discriminator. Also, Adam optimisers are used for the generator and discriminator, each with its own learning rates to balance the training

process.

Parameter setting for the CGAN model:

Parameter	**Details**
Generator Optimizer	Learning rate = 0.0005; β_1: 0.5
Discriminator Optimizer	Learning rate = 0.0001; β_1: 0.5

Table 4.3: CGAN Model

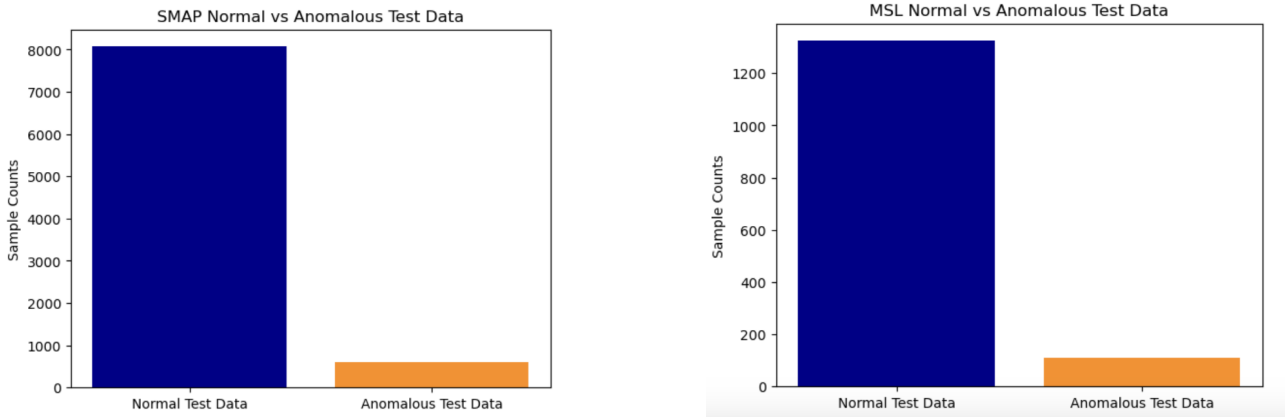
In general, this CGAN setup allows the generator to learn to generate synthetic data based on specified conditions (normal or anomalous), while the discriminator determines the reliability of these samples.

CHAPTER 5

Methodology

5.1 Test Set Counts

As shown in fig. 5.1, the test dataset is highly imbalanced for both SMAP and MSL, which are used for all the evaluation processes in the following phases.



(a) SMAP Test Set

(b) MSL Test Set

Figure 5.1: Test Set Counts

5.2 Integration of Real and CGAN Anomalies

5.2.1 CGAN Training

As previously discussed in chapter 4, the anomaly detection process leverages the CGAN framework to generate synthetic anomalies. The CGAN framework is trained using unlabelled real training data, which is assumed to contain only normal telemetry data. At the start of the training process, random noise and conditional labels are concatenated to create the generator input. The discriminator then undergoes training on batches of normal and synthetic data, with the loss determining how effectively it differentiates between real and fake data.

Parameter setting for the CGAN training:

Parameters	**Details**
Epochs	100
Batch Size	32
Half Batch Size	16
Noise	Mean: 0.1 ; Std Dev: 0.8

Table 5.1: SMAP and MSL CGAN Training Configuration

Subsequently, the generator is updated based on its ability to trick the discriminator, with loss indicating how successful it is at creating real sequences. The training process ensures that the generator and discriminator losses remain stable over time. The model is trained for several epochs to capture the complexity of the data, and the batch size is optimised for efficient training.

SMAP CGAN Training Analysis

Fig. 5.2 represents the generator and discriminator losses of the SMAP-CGAN framework, trained across 100 epochs. The generator loss gradually decreases over time and learns the data pattern, generating synthetic anomalies that deviate from the normal data distribution pattern based on the applied conditions. As the generator's output improves, the discrimina-

tor loss increases with time and becomes less effective in differentiating synthetic anomalies from the normal data patterns. This shows a balanced process where the generator learns and improves its output over time to generate realistic synthetic anomalies. Thus, this relationship between the loss functions shows an appropriate balance in the conditional GAN training process. The approximate time taken to complete the training process is 59 seconds, which shows the model's efficiency to learn data patterns in a short duration.

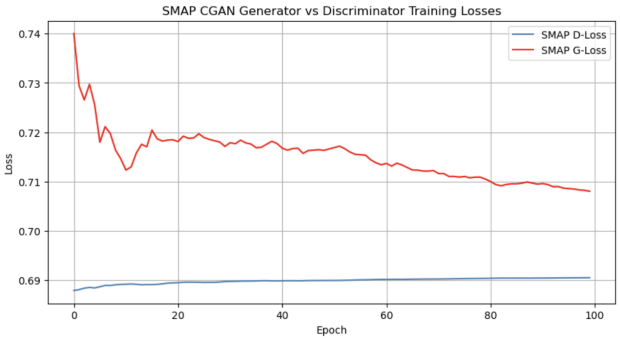


Figure 5.2: SMAP CGAN Losses

Fig. 5.3 shows the comparison of generated synthetic anomalies and real test set anomalies by considering a few samples. The sample plots indicate that the synthetic anomalies effectively capture key features of the real anomalies but are not able to fully replicate a few temporal patterns. The DTW distance of 15.11 suggests similarity, achieving the shortest distance but with slight deviation between the real and synthetic anomalous patterns. This process can be further improved by fine-tuning the parameters and framework.

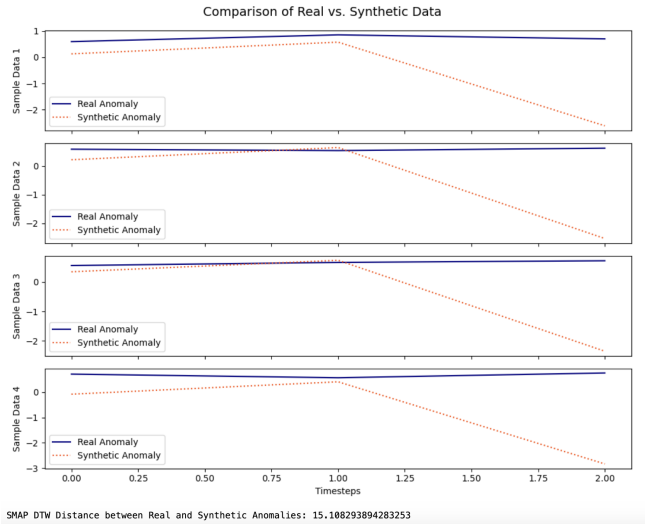


Figure 5.3: SMAP Comparison Real Vs Synthetic Anomalies

Overall, the analysis shows the ability of the SMAP CGAN framework to generate synthetic anomalies by leveraging normal unlabelled telemetry data. The stable downward pattern in generator loss, along with the similarities between synthetic and real anomaly patterns indicated by visualisation, highlights the strength of the model to mimic key features of anomalous patterns. Further, these synthetic anomalies are combined with real telemetry data to enhance the training dataset, providing additional anomalous data in subsequent steps. However, further refinement may be considered to stabilise the CGAN training process and to achieve more accurate synthetic anomalies, aligning closely with real anomaly patterns.

MSL CGAN Training Analysis

The plot in fig. 5.4 shows the generator loss, which gradually stabilises over time, indicating that the model is learning to generate synthetic anomalies that deviate from the normal data pattern. Similarly, the discriminator loss remains consistent and maintains its ability to differentiate between real and synthetic anomalous patterns. However, the observed loss functions indicate training instability, which is a well-known issue while training GANs. This requires further improvement in the training process to improve the synthetic data quality generated by CGAN. The approximate time taken to complete the training process is 60 seconds.

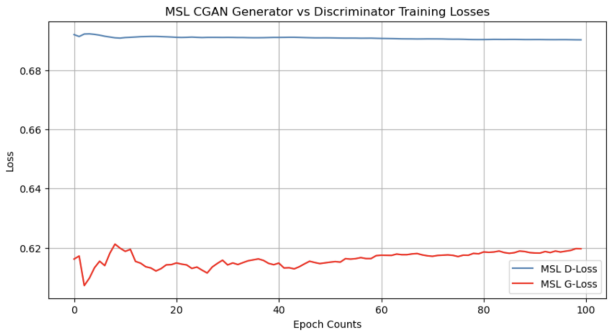


Figure 5.4: MSL CGAN Losses

As shown in fig. 5.5, the sample data shows that the synthetic anomalies show deviation from the real anomalous data pattern. Furthermore, the DTW distance of 66.23 shows a moderate difference between the synthetic and real anomalies.

Based on the analysis, the MSL CGAN framework generates synthetic anomalies to some extent, and the model has the ability to replicate some key features of anomalies but with

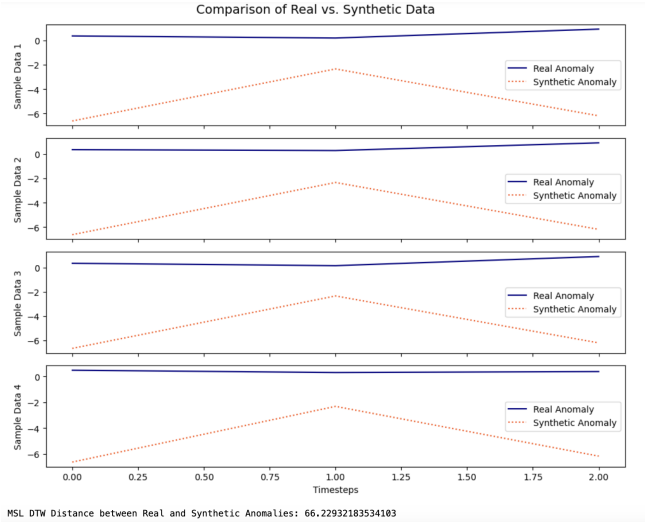
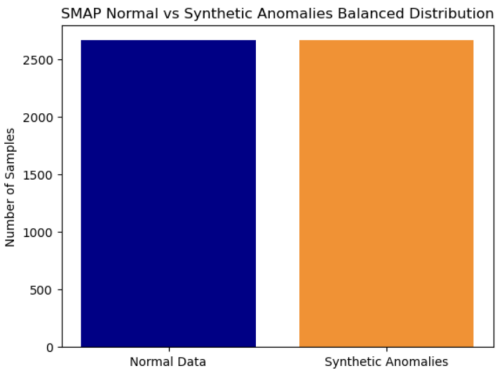
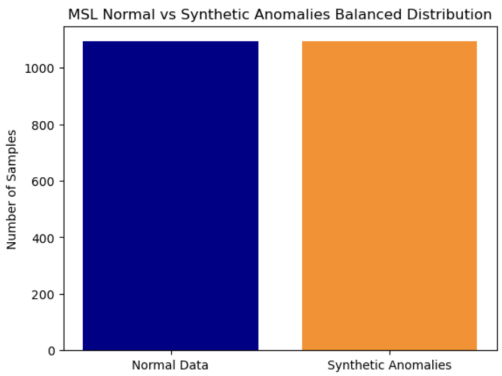


Figure 5.5: MSL Comparison Real Vs Synthetic Anomalies

minor deviations. In the following step, the observed synthetic anomalies are integrated into the real training dataset. This integration is considered to validate the overall diversity of the training dataset, which may improve the anomaly detection process.



(a) SMAP Balanced Training Set



(b) MSL Balanced Training Set

Figure 5.6: CGAN Integration Based Balanced Training Set

As shown in Fig. 5.6, the observed synthetic anomalies are combined with the real training dataset for both SMAP and MSL to form a more balanced training set. This process adopts a hybrid approach with synthetic anomalies added to the real data in order to enhance the model's performance in identifying anomalies efficiently.

5.3 Majority Voting Technique

The ensemble learning approach integrates the predictions of multiple classifiers to produce a stronger classifier, including techniques such as boosting, bagging, stacking, and majority voting classifiers, thereby enhancing overall performance Xue and Zhu [25]. The same approach is considered for this paper to identify anomalies in telemetry data, and the models used for this purpose are the Isolation Forest (IF), One-Class Support Vector Machine (OCSVM), and Long Short-Term Memory autoencoder (LSTM AE). These three models classify the data point as normal or anomalous, and then their predictions are combined using a majority voting technique. The output is obtained by a majority vote from the three models, resulting in more reliable results than the individual models.

5.3.1 Model Training and Evaluation Configuration

LSTM Autoencoder

The LSTM AE framework learns the normal patterns in the telemetry data and differentiates any deviation using the reconstruction error (MSE). In this study, the structural concept of LSTM AE is considered from the analysis performed by Brownlee [4]. The telemetry data is converted into a suitable format for LSTM input, where the dimension is taken as one timestep along with three input features. The model is configured based on an encoder-decoder concept, where the encoder has used bidirectional LSTM layers to identify temporal patterns in the input data sequence and then used dropout layers to avoid overfitting. Also, the relu activation function is used in all the layers, and a repeat vector layer is added to match the output with the input data sequence length.

The decoder layer is added to reconstruct the input sequences, and a time-distributed dense layer is used to map the output to the same dimensions as the input feature structure. Additionally, one more layer is added where the model is optimised using the Adam optimiser with a low learning rate for consistent weight updates. Further, the Mean Squared Error (MSE) is considered as the loss function to evaluate reconstruction accuracy, allowing the model to detect abnormal patterns by identifying variation in reconstruction error.

Layer	**Details**
LSTM (Bidirectional)	64 units; return sequences= True
LSTM (Second)	64 units; return sequences = False
LSTM (Decoder)	64 units; return sequences = False
Repeat Vector	Timesteps = 1
Dropout Rate	0.4 (Each)
Activation	ReLU (Each)
Learning Rate	SMAP: 0.0001; MSL: 0.0005
Epochs	50
Batch Size	64

Table 5.2: LSTM Autoencoder Layers

During the training phase, the model learns to reconstruct the input data, and afterwards reconstruction errors are calculated as the MSE between the input sequences and the predicted output sequences for the training set. Based on these training errors, a high threshold is set at the 85th percentile for SMAP and the 80th percentile for MSL. By setting this threshold based on training MSE errors, a test sample is classified as an anomaly if the MSE exceeds this threshold. After training, the LSTM AE is applied to the test datasets, using the same reconstruction-based approach and the already set threshold to detect anomalies. Based on this approach, the data points with reconstruction errors for the test set that exceed the defined threshold are marked as anomalous.

Isolation Forest

The Isolation Forest (IF) model is trained using the telemetry data to learn normal patterns and identify abnormal patterns. The model isolates points that don't match the training patterns. Further, the model evaluates each data point from the test set by estimating how isolated it is from the normal data points. Again, the test point is considered to be anomalous if it is far away from normal clusters. The output value -1 represents anomalies, and 1 represents normal data points. Further, the outputs are converted to 0 for anomalies and 1 for normal data. Generally, the boundary in an isolation forest is set based on the depth of isolation, with the contamination parameter determining the expected number of anomalies.

In this study, for both SMAP and MSL, the contamination parameter is set to 0.01, which is the proportion of anomalies in the dataset. Again, the max samples parameter is set to 0.25 and used as a smaller subset to train each tree faster without impacting the performance.

One-Class SVM

The One-Class SVM learns the boundary for normal data patterns by training the model on telemetry data and marking any deviation beyond this boundary as anomalous. The model is set with a radial basis function (RBF) kernel, a gamma value of 0.33, and regularisation parameters (nu) of 0.1 for SMAP and 0.2 for MSL. Similar to Isolation Forest, the model predicts anomalies as -1 and normal data as 1, which are then converted to binary labels of 0 and 1.

Overall, the predictions from the three models are integrated to produce the final anomaly labels. The labelled test dataset with known anomalies is used to determine the effectiveness of the model to identify anomalies. The key metrics, like precision, recall, and F1-score, are determined for this purpose. Again, for visualisation, confusion matrices and ROC curves are plotted to get an overview of the classification outcomes.

Initially, the analysis is performed using only real telemetry data. Subsequently, the ensemble model is trained on a combined dataset consisting of real telemetry data and synthetic anomalies generated by the CGAN framework. Further, the efficiency of the model is determined by the same procedures as applied for the real telemetry data. This experiment is conducted for both the SMAP and MSL datasets. In addition, SHAP analysis is considered for all the models of the real and combined datasets separately to enhance explainability in the following sections.

5.3.2 Evaluation Metrics

Precision

The precision considers the percentage of correctly identified anomalies within overall predicted anomalies.

Mathematically, precision is given by:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Where,

- TP: True Positives
- FP: False Positives

Recall

The recall considers the percentage of true positive rates in our case, which is an anomaly, and evaluates the performance of the model to detect the anomaly correctly.

Mathematically, recall is given by:

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

Where,

- TP: True Positives
- FN: False Negatives

F1-Score

The F1 Score is achieved by balancing both recall and precision and suggests the overall performance of the model.

Mathematically, the F1 score is given by:

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The above metrics help to determine the efficiency of the model in identifying irregular patterns. Additionally, confusion matrices and ROC curves are taken into consideration for the visualisation of performance for both real and combined datasets.

CHAPTER 6

Results

6.1 SMAP Metrics and Visualisation

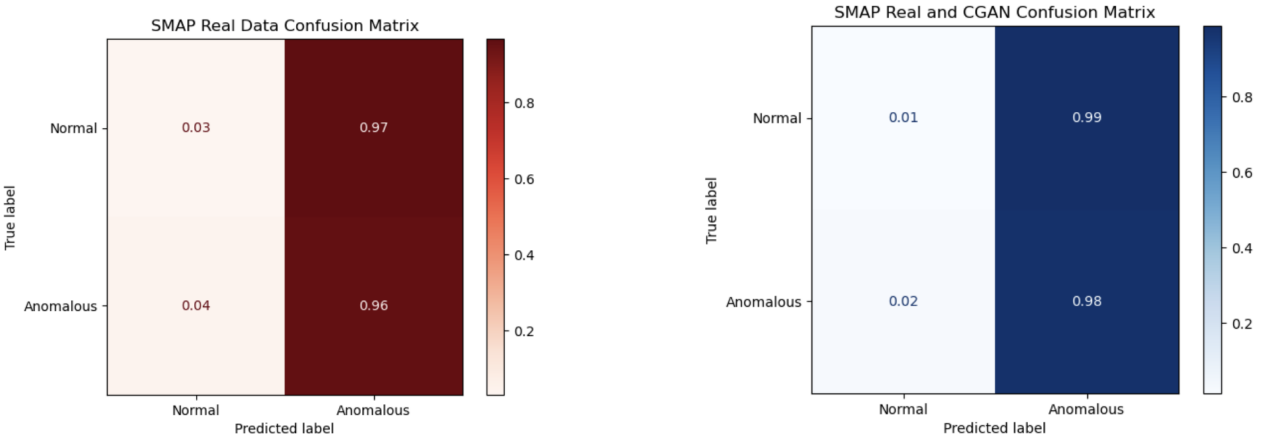
This section discusses the results and outputs of the SMAP real and combined datasets. For the real dataset, the majority voting technique achieves a high recall and a low F1 score with a low precision. In fig 6.1 (a), the confusion matrix for the majority voting technique shows a high true positive rate for detecting anomalous patterns effectively. However, the model struggles to classify normal patterns, resulting in a high false positive rate and low precision.

Models	**Precision**	**Recall**	**F1 Score**
Majority Voting	0.07	0.96	0.13
LSTM Autoencoder	0.121	0.327	0.176
Isolation Forest	0.069	0.980	0.129
One-Class SVM	0.063	0.787	0.116

Table 6.1: SMAP Real Data Metrics Across Models

The individual models are evaluated to provide additional information on their performance. As per the results, the LSTM autoencoder shows a limited ability to detect anomalies effectively. However, the Isolation Forest performs better than the LSTM autoencoder in recall

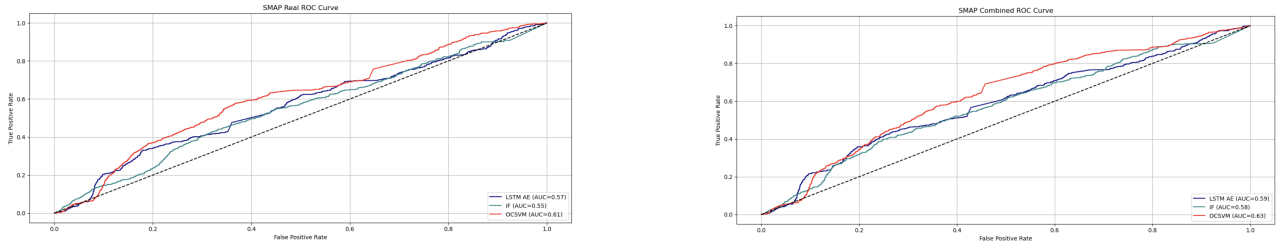
but again struggles with low precision and F1 score. Similarly, the One-Class SVM shows improved recall compared to the LSTM autoencoder while marginally underperforming the Isolation Forest in terms of recall. As shown in Fig. 6.2 (a), the ROC curve analysis is considered to validate these findings and suggest that the One-Class SVM achieves the highest AUC of 0.61, followed by the LSTM autoencoder and Isolation Forest with an AUC of 0.57 and 0.55 respectively.



(a) SMAP Real Confusion Matrix

(b) SMAP Combined Confusion Matrix

Figure 6.1: SMAP Confusion Matrices



(a) SMAP Real ROC

(b) SMAP Combined ROC

Figure 6.2: SMAP ROC

The integration of CGAN synthetic anomalies into the real training dataset shows a notable improvement in the recall and AUC of the models. For the combined dataset, the majority voting technique improves recall. However, both the precision and F1 score remain constant. The plot in fig 6.1 (b) for the majority voting technique further shows the improvement, with an increased true positive rate, and indicates that the synthetic anomalies enhance the model's performance in detecting anomalous patterns.

The individual models are evaluated, where the LSTM autoencoder shows a marginal im-

Models	Precision	Recall	F1 Score
Majority Voting	0.07	0.98	0.13
LSTM Autoencoder	0.118	0.359	0.178
Isolation Forest	0.069	0.998	0.129
One-Class SVM	0.066	0.860	0.123

Table 6.2: SMAP Combined Data Metrics Across Models

provement with the metrics, highlighting the model's adaptability to the refined dataset. The Isolation Forest maintains a high recall but struggles with low precision and F1 score. On the other hand, the One-Class SVM shows better adaptability to synthetic anomalies with a high recall and a slightly improved F1 score. In fig 6.2 (b), the ROC curve analysis shows improvements across all models, where the One-Class SVM achieves the highest AUC of 0.63 and the LSTM Autoencoder at 0.59 with Isolation Forest at 0.58.

In general, these results show the potential of CGAN-generated anomalies in enhancing the model's performance to predict and identify anomalous patterns more efficiently, which can be seen from the recall, ROC curves, and AUC values. Also, a high recall is considered crucial in anomaly detection to ensure that no irregular patterns are missed, thereby maintaining the efficacy of the overall system.

6.2 MSL Metrics and Visualisation

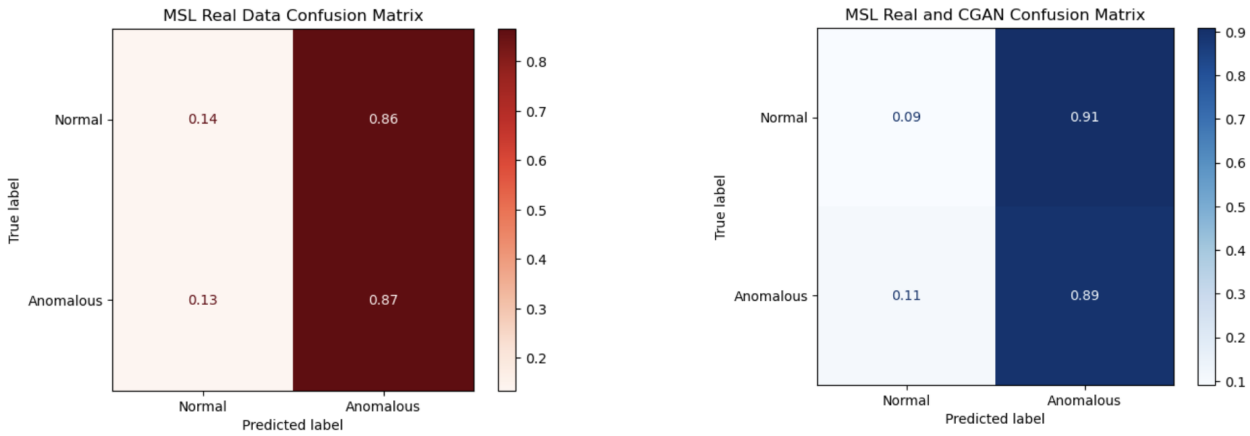
For the MSL real dataset, the majority voting technique achieves a recall of 0.87 and can identify anomalies to some extent. This is represented in the confusion matrix for the majority voting technique in fig. 6.3 (a). However, the low F1 score and precision show challenges in differentiating normal patterns, resulting in a high false positive rates.

Individual models are also analysed where the Isolation Forest shows better performance than the other models with a high recall and an improved F1 score. Similarly, the One-Class SVM gets moderate recall with an F1 score and shows moderate performance. The LSTM autoencoder indicates the training instability along with the inability to detect anomalies

Models	Precision	Recall	F1 Score
Majority Voting	0.08	0.87	0.14
LSTM Autoencoder	0.073	0.295	0.118
Isolation Forest	0.077	0.929	0.143
One-Class SVM	0.076	0.714	0.138

Table 6.3: MSL Real Data Metrics Across Models

properly.

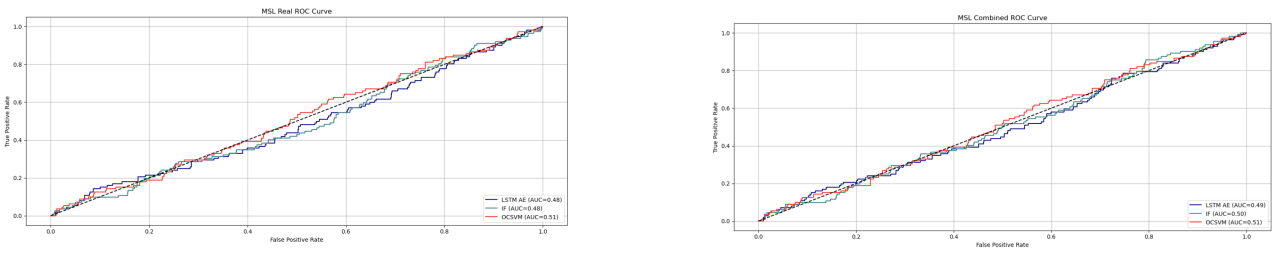


(a) MSL Real Confusion Matrix

(b) MSL Combined Confusion Matrix

Figure 6.3: MSL Confusion Matrices

The ROC curve in fig. 6.4(a) for the real dataset shows randomness in the models, with AUC scores of 0.48 for both LSTM and Isolation Forest. However, OCSVM performs slightly better with an AUC value of 0.51. Again, these findings validate the issues of identifying anomalies in the MSL telemetry dataset. After integrating the MSL real data with CGAN synthetic



(a) MSL Real ROC

(b) MSL Combined ROC

Figure 6.4: MSL ROC

anomalies, the performance metrics improved slightly. In Fig. 6.3 (b), the confusion matrix for the majority voting technique shows a better true positive rate of 89% and suggests the advantages of adding synthetic anomalies to the real training dataset. However, the false positive rate for normal data remains a known challenge.

Models	**Precision**	**Recall**	**F1 Score**
Majority Voting	0.08	0.89	0.14
LSTM Autoencoder	0.074	0.357	0.123
Isolation Forest	0.076	0.955	0.141
One-Class SVM	0.076	0.714	0.138

Table 6.4: MSL Combined Data Metrics Across Models

Across all the individual models, the Isolation Forest performs most effectively with a high recall and shows its adaptability to the refined dataset. Similarly, One-Class SVM remains consistent in the metrics. The LSTM autoencoder also shows slight improvement in the performance but remains unstable during training while detecting anomalies. In fig 6.4 (b), the AUC scores for the LSTM autoencoder, Isolation Forest, and One Class SVM are 0.49, 0.50, and 0.51, respectively. The ROC curve for the combined dataset shows marginal improvement over the real dataset. However, the plot shows randomness and suggests that the CGAN synthetic anomalies have not completely addressed the issue and require further refinement.

In general, the above observation aligns with the earlier findings where the CGAN training for MSL showed signs of instability along with an average DTW distance and also suggests that the CGAN model is not completely capturing the complexity and variability in the MSL telemetry data. Furthermore, the issue with the MSL dataset and metrics has already been observed by Hundman et al. [11], where the dataset shows challenges to detect anomalies due to the complexity and variability of MSL operations.

Apart from this, similar experiments as performed above are carried out for both SMAP and MSL using different thresholds for LSTM AE and parameters for both OCSVM and IF. The LSTM Autoencoder model used thresholds of 80th, 90th, and 95th for SMAP and 85th, 90th, and 95th for MSL. Similarly, Isolation Forest used contamination rates of 0.05 and 0.1 for

model evaluation. One-Class SVM adjusted nu values to 0.05 and 0.2 for SMAP and 0.05 and 0.1 for MSL. However, no significant improvements are observed with the change in threshold and parameters. Therefore, after experimenting with these multiple thresholds and parameters for each model, the most appropriate results are presented in this paper along with the required plots and metrics.

6.3 Comparision of Real and Combined Metrics

The plots as shown in Fig. 6.5 achieve good recall and show the potential to identify anomalies without missing any certain instance. However, the false positive rates remain a challenge and require further improvement. Despite the challenges, the results show the importance of combining CGAN synthetic anomalies and their contribution to improving the anomaly detection system, which is earlier observed in the ROC curves. Again, the combined framework of the majority voting technique and CGAN shows potential as an initial step toward adopting a hybrid approach for anomaly detection.

(a) SMAP Real Vs Combined

(b) MSL Real Vs Combined

Figure 6.5: Final Metrics Comparision

CHAPTER 7

SHAP-XAI Based Feature Analysis

SHAP explanations are applied to all three models based on the ideas gathered from the work done by Lundberg [15] and Roshan and Zafar [20]. For the One-Class SVM, a subset of 100 samples is used from the training dataset, which is considered as the background dataset for the kernel explainer. Further, a subset of 50 test set samples is considered for SHAP analysis to reduce time and extra computing load. Similarly, the SHAP analysis is performed using reconstruction error for the LSTM Autoencoder, where a subset of 100 training data samples is taken as the background dataset, and analysis is done using 50 test data samples. The use of these subsets for kernel explainer in the case of one-class SVM and LSTM autoencoder is required due to the time-consuming issue of the SHAP approach as mentioned by Roshan and Zafar [20]. Again, the Isolation Forest is a tree-based model for which SHAP TreeExplainer is taken into consideration, and SHAP analysis is performed with the complete test dataset without using any subset. In general, the SHAP analysis is applied across the models to provide valuable information based on features like waveform, frequency, and magnitude and determine the key contributors to the anomaly detection process.

7.1 SMAP SHAP Analysis

SHAP analysis on the SMAP real dataset in fig 7.1 highlights the importance of the feature in all the models. Both the One-Class SVM and the Isolation Forest show the waveform as the most relevant feature and suggest that the fluctuations in the waveform are the key factors for identifying abnormal patterns in these models. Again, frequency and magnitude are the next relevant features for these models. The LSTM autoencoder determines frequency as the relevant feature, and next to it are magnitude and waveform. Hence, the analysis provides a clear understanding of the better results shown by the Isolation Forest and the One-Class SVM compared to the LSTM Autoencoder with respect to recall and AUC values.

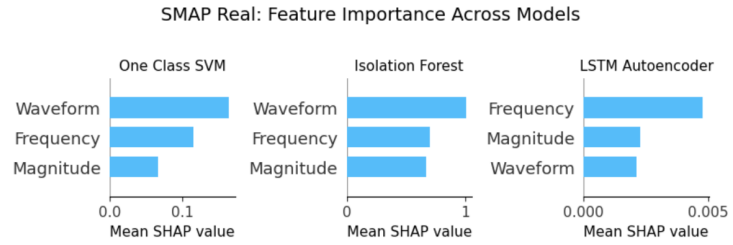


Figure 7.1: SMAP SHAP Analysis for Real

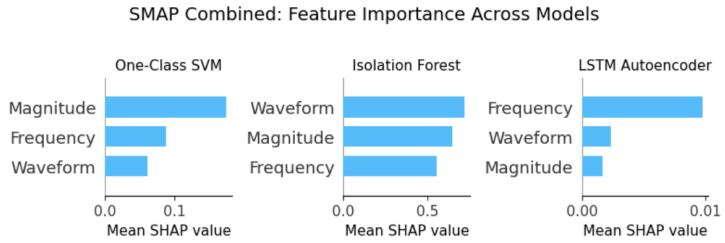


Figure 7.2: SMAP SHAP Analysis for Combined

In fig 7.2, the SHAP analysis shows the importance of explanations with each feature contributing to the anomaly detection for the combined dataset. The magnitude is the most relevant feature for One-Class SVM, and thereafter, frequency and waveform show the feature importance in detecting abnormal patterns. The change in feature positions shows the improved performance of the model after adding CGAN anomalies. Similarly, the waveform remains the relevant feature for the Isolation Forest, and even the other two features contribute to some extent to identifying the anomalous patterns. There is a change for the LSTM autoencoder with feature positions, where the model has adjusted to irregular patterns

mainly caused by frequency changes. Thus, the SHAP analysis for SMAP suggests that the CGAN synthetic anomalies serve to some extent and work as an initial step towards enhancing the anomaly detection process.

7.2 MSL SHAP Analysis

The SHAP analysis in fig 7.3 of the MSL real dataset shows that waveform is the most important feature in all three models. Waveform is a highly efficient feature contributing to anomaly detection for one-class SVM, followed by frequency and magnitude. Similarly, in Isolation Forest, waveform acts as the most relevant feature, along with magnitude and frequency next to it. For the LSTM autoencoder, waveform is the relevant feature, while frequency shows a moderate contribution and magnitude shows the least effect in detecting anomalies.

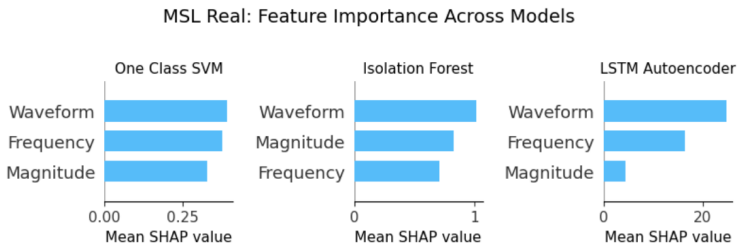


Figure 7.3: MSL SHAP Analysis for Real

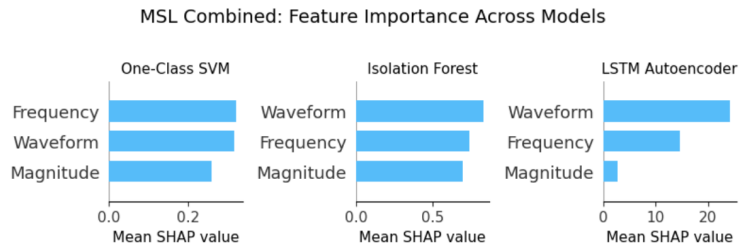


Figure 7.4: MSL SHAP Analysis for Combined

In the combined dataset, fig 7.4, One-Class SVM considers frequency as the most relevant feature, followed closely by waveform and magnitude. Again, Isolation Forest shows almost equal impact from all the features, where waveform remains significant and frequency along with magnitude shows a slightly better impact than the real dataset. For LSTM autoencoders,

the waveform continues to be the relevant feature, and magnitude has the least impact in the whole process. Thus, the results show slight variation in feature rankings after the addition of CGAN synthetic anomalies.

CHAPTER 8

Discussion and Future Scope

This chapter focuses on the challenges observed during the experiments and provides suggestions for future work to enhance the anomaly detection process in the following sections. The primary challenge observed is the high rate of false positive rates, which is quite common in anomaly detection tasks. In addition, the neural networks, both LSTM AE and CGAN, are highly complex due to the architectural layers and training time. Hence, the networks require further refinement and stability for better performance. Further, extraction of features plays a vital role in time series data analysis. According to Hundman et al. [11], refined feature engineering is the most critical part and needs to be explored further for an enhanced anomaly detection system.

8.1 False Positive Rates

As discussed in the previous chapter, this study achieves high recall, which is crucial in spacecraft telemetry to capture most of the anomalies. Usually, recall is given a high priority in the case of anomaly detection, as an undetected anomaly can have a severe impact on the whole system and may lead to system failure. However, the low precision shows a false positive rate, which may reduce the overall effectiveness of the system. Due to the nature of precision, the F1 score finds it challenging to achieve an appropriate balance. The same

issue has been observed in the existing studies, where Hundman et al. [11] mentioned the challenges of high false positive rates in their research, and Geiger et al. [10] mentioned that sliding-window techniques may contribute to false positive rates. To overcome these issues, both studies used an anomaly pruning approach to reduce false positive rates but impacted the recall. Furthermore, in this paper, a simple majority voting technique is used to combine predictions from each model to avoid false positive rates and enhance the model's robustness. However, the approach does not completely eliminate false positives and depends on the individual performance of models.

Future work can include analysis with a robust feature extraction process and using advanced frameworks like the weighted majority voting technique with different models. These techniques can be used to further reduce false positive rates along with the anomaly pruning approach without impacting recall.

8.2 CGAN Training Instability and Synthetic Anomalies

As observed in our study, the CGAN framework shows great potential for generating synthetic anomalies in the anomaly detection tasks. However, during the experiment, a major issue identified with the CGAN framework is training instability. Despite setting the random seed, multiple training executions produced varied results and impacted the metrics. Furthermore, it has been observed that the CGAN frameworks are time-consuming depending on increased epochs and batch size. In addition, the training instability in generative models, particularly in GAN frameworks, has been discussed in previous studies. According to Chen [6], GAN frameworks generally experience mode collapse and instability regardless of their advanced applications. Further, Kumar and Sharma [13] mentioned the training instability of GANs, which leads to mode collapse, where the generator produces a few varieties of outputs and lacks the ability to learn the entire data pattern.

Despite these challenges, the CGAN model, along with the majority voting technique, shows the ability to achieve high recall with a better AUC score by improving the CGAN training stability and making the model efficient for anomaly detection tasks. Future research can explore advanced techniques to stabilise CGAN training. This includes the use of feature

matching, as performed by Akcay, Atapour-Abarghouei, and Breckon [2] in GANomaly to reduce training instability of a different GAN framework. In addition, using advanced loss functions, such as Wasserstein loss, can be explored to avoid mode collapse and improve training stability. Further, saving and reloading the CGAN model after the stabilisation of loss functions is expected to stabilise varied results observed throughout the experiments. In addition, fine-tuning the architectural layers and hyperparameters can enhance the robustness of the CGAN framework. Additionally, the generated synthetic anomalies can be further compared against real anomalies using more additional statistical techniques. By replicating the accurate real-world anomalies, the training set can be refined and prepared for effective anomaly detection tasks.

CHAPTER 9

Conclusions

Overall, this paper is designed as an initial step towards developing a hybrid approach for anomaly detection, where the labelled data are insufficient in real-world scenarios. The analysis and findings show the potential of CGAN in improving the anomaly detection techniques in the space domain by adding synthetic anomalies in the training process. Despite the training challenges in neural networks and false positive rates, the results manage to show improvement in recall and ROC curve across experiments, which are necessary for anomaly detection systems. However, the current analysis highlights the need for further research in CGAN to get more accurate anomalies. In addition, there's a scope for further refinement in the feature engineering process and ensemble techniques in order to achieve a robust anomaly detection system. Apart from this, SHAP analysis shows the significance of the explanations for each feature in the anomaly detection process. All three features, waveform, magnitude, and frequency, contribute to the detection process and provide the meaningful information for drawing conclusions about certain anomaly detection decisions. Hence, the approach shows promise and serves as a foundation for broader applications in spacecraft missions and other critical sectors like finance, healthcare, and energy to enhance the anomaly detection system.

Bibliography

[1] Edmund Fosu Agyemang. "Anomaly detection using unsupervised machine learning algorithms: A simulation study". In: *Scientific African* 26 (2024), e02386.

[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. "Ganomaly: Semi-supervised anomaly detection via adversarial training". In: *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer. 2019, pp. 622–637.

[3] Jason Brownlee. *How to Develop a Conditional GAN (cGAN) From Scratch*. en-US. July 2019. URL: https://www.machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/.

[4] Jason Brownlee. *How to Develop LSTM Autoencoders for Anomaly Detection in Python*. en-US. Aug. 2020. URL: https://machinelearningmastery.com/lstm-autoencoders/.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[6] Haiyang Chen. "Challenges and corresponding solutions of generative adversarial networks (GANs): a survey study". In: *Journal of Physics: Conference Series*. Vol. 1827. 1. IOP Publishing. 2021, p. 012066.

[7] Nwodo Benita Chikodili et al. "Outlier detection in multivariate time series data using a fusion of K-medoid, standardized euclidean distance and Z-score". In: *International Conference on Information and Communication Technology and Applications*. Springer. 2020, pp. 259–271.

[8] Mohamed Ahmed Abo El-Enen et al. "Fraud Detection in Medical Insurance Claims Using Majority Voting of Multiple Unsupervised Algorithms". In: *Procedia Computer Science* 244 (2024), pp. 9–22.

[9] Gonzalo Farias et al. “Explainable Anomaly Detection in Spacecraft Telemetry”. In: (2024).

[10] Alexander Geiger et al. “Tadgan: Time series anomaly detection using generative adversarial networks”. In: *2020 ieee international conference on big data (big data)*. IEEE. 2020, pp. 33–43.

[11] Kyle Hundman et al. “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding”. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 387–395.

[12] Lattawit Kulanuwat et al. “Anomaly detection using a sliding window technique and data imputation with machine learning for hydrological time series”. In: *Water* 13.13 (2021), p. 1862.

[13] Subodh Kumar and Manisha Sharma. “A Comprehensive Review of Generative AI - From its Origins to Today and Beyond”. In: *Preprint* (2024). DOI: 10.13140/RG.2.2.19420.81281.

[14] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen. “A survey on explainable anomaly detection”. In: *ACM Transactions on Knowledge Discovery from Data* 18.1 (2023), pp. 1–54.

[15] Scott Lundberg. *Welcome to the SHAP Documentation*. 2018. URL: https://shap.readthedocs.io/en/latest/.

[16] Fernando Mateo et al. “Dynamic classifier auditing by unsupervised anomaly detection methods: an application in packaging industry predictive maintenance”. In: *arXiv preprint arXiv:2405.11960* (2024).

[17] Sowmya Ramesh Kumar. “Anomaly Detection Techniques in Time Series Forecasting: Identifying Outliers”. en. In: *International Journal of Science and Research (IJSR)* 9.11 (Nov. 2020), pp. 1707–1709. ISSN: 23197064. DOI: 10.21275/SR24213014030. URL: https://www.ijsr.net/getabstract.php?paperid=SR24213014030.

[18] Rohit Raturi et al. “A novel approach for anomaly detection in time-series data using generative adversarial networks”. In: *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. IEEE. 2023, pp. 1352–1357.

[19] Param Raval. *Moving Average Time Series Model*. en-US. Oct. 2024. URL: https://www.projectpro.io/article/moving-average-time-series-model/716/.

[20] Khushnaseeb Roshan and Aasim Zafar. "Using kernel shap xai method to optimize the network anomaly detection model". In: *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE. 2022, pp. 74–80.

[21] Thomas Schlegl et al. "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks". In: *Medical image analysis* 54 (2019), pp. 30–44.

[22] Laya Rafiee Sevyeri and Thomas Fevens. "Ad-cgan: Contrastive generative adversarial network for anomaly detection". In: *International Conference on Image Analysis and Processing*. Springer. 2022, pp. 322–334.

[23] Yue Song et al. "Telemetry data-based spacecraft anomaly detection using generative adversarial networks". In: *2020 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD)*. IEEE. 2020, pp. 297–301.

[24] Belén Vega-Márquez et al. "Creation of synthetic data with conditional generative adversarial networks". In: *14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019) Seville, Spain, May 13–15, 2019, Proceedings 14*. Springer. 2020, pp. 231–240.

[25] Liang Xue and Tianqing Zhu. "Hybrid resampling and weighted majority voting for multi-class anomaly detection on imbalanced malware and network traffic data". In: *Engineering Applications of Artificial Intelligence* 128 (2024), p. 107568.