A Comparative Analysis of Anomaly Detection Techniques for Battery Telemetry Data in Low Earth Orbit Remote Sensing Satellites

Ahmed Adam[1], Ahmed Mokhtar[2], Ahmed Mattar[3],Mohamed Abdrahman[4],and Sherif Helmy[5]

[1] *Electric Power and Energy Department, Military Technical College,Cairo, Egypt*
**Corresponding author: ahmed_mahgoub83@hotmail.com*

[2] *Aircraft Electric Equipment Department, Military Technical College, Cairo, Egypt.*
ahmed_refaie@ieee.org

[3] *Computers and Artificial Intelligence Department, Military Technical College, Cairo, Egypt.*
a.matter@ieee.org

[4] *Electrical Power and Machines Engineering Department, Faculty of Engineering, Ain Shams university, Cairo,Egypt.*
mohamed_a_rahaman@eng.asu.edu.eg

[5] *Electric Power and Energy Department, Military Technical College,Cairo, Egypt*
Sherif.helmy@mtc.edu.eg

ABSTRACT

The article presents a comprehensive assessment of telemetry data of batteries used in low-earth orbit satellites. The study further performs an analysis of the performance of using different anomaly detection techniques, including Statistical (Z-Score), Machine Learning (One class support vector machine OCSVM, Isolation Forest), Deep Learning (Autoencoder), and Hybrid Approaches (Autoencoder and neural network and Autoencoder and Z-score). This study introduces and evaluates a hybrid anomaly detection framework combining deep learning-based feature compression (Autoencoder) with various downstream classifiers. The models are validated on real satellite telemetry data and benchmarked using medical electrocardiogram ECG datasets for generalizability. In addition, the study continues to analyze the system by detecting the faulty sensor that was responsible for the detected anomalies, which can help the operators to get a more accurate analysis of the system.

1. INTRODUCTION

Any satellite mission needs a continuous power source throughout its lifetime under all conditions and modes of operation with different mission types. Satellites that have solar panels as their primary power source need batteries to supply them during eclipse periods. The power supply system (PSS) is considered the soul of the spacecraft(Peng, Fan, Xiao, & Tang, 2014; Tennberg & Ekeroot, 2021). The demand for reliable and efficient power sources in low-earth orbit (LEO) satellites is an essential objective (Mokhtar et al., 2024; Mostacciuolo et al., 2019; B. Lee & Wang, 2010), focusing on nickel-hydrogen batteries that have been used for decades in space due to their qualified performance. However, it suffers from size, weight, lower energy density, and operational complexity (Thaller & Zimmerman, 2003; Lewin, 1999; Balan & Müller, 2015; Purushothaman, Binu, Philip, Pillai, & Ilangovan, 2011; Lou, 2021). Also, in this mission case study, it suffers from sensor anomalies, notably pressure differences (ΔP), which impact the battery's performance. Thus, it will have a great consequence on the mission's lifetime. The battery has some important sensors that influence the assessment of its performance; the most notable are current (C), capacity (Cap.), voltage (V), temperature (T), and four pressure sensors (P1, P2, P3, and P4) put on the controlled cells.

https://doi.org/10.36001/IJPHM.2025.v16i2.4273

The readings of these cells control the algorithms of charging and discharging (because the average of pressure sensors is the main factor of activation or deactivation of the algorithm). The main problem faced in this case was the increasing difference in pressure readings (ΔP) Eq.(1) for pressure sensors that made failure messages and had a direct effect on the behavior of the mentioned algorithms.

$$\Delta P^j = \text{(max value - min value) of pressure sensors} \quad (1)$$

Where j is the number of the observation.

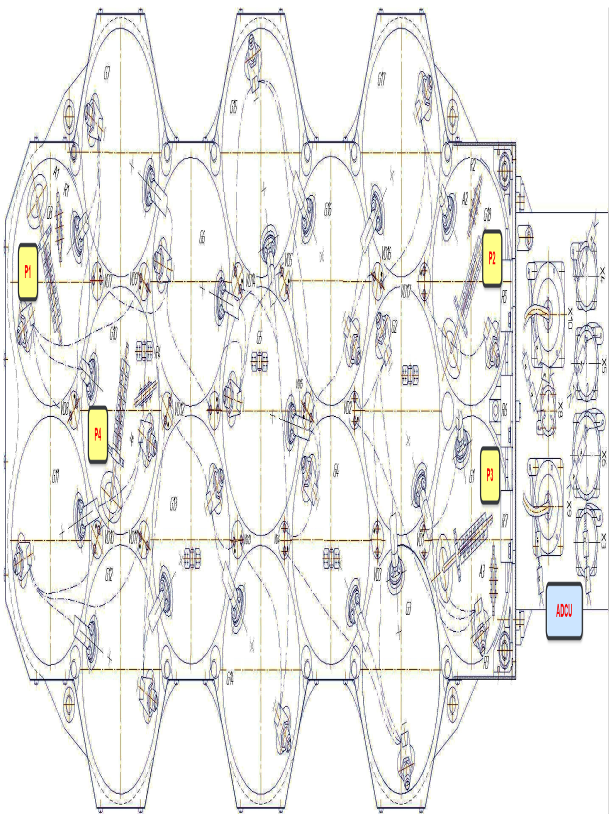

Figure 1. Battery layout

This case study involves a remote sensing satellite in low Earth orbit, equipped with two nickel-hydrogen battery modules, each containing 17 active cells. The schematic representation of the nickel-hydrogen battery module used in the satellite system, showing the placement of pressure sensors (P1–P4) and active cells, is shown in Figure 1. These batteries supply power during eclipse periods or misalignment during imaging and maneuvers. There are several sensors reading; as mentioned previously, these readings are collected by the analog/digital conversion unit (ADCU) and sent through the power control unit (PCU) to the ground station as telemetry every communication session. The main objective is to detect the anomaly of the battery's telemetry received.

Anomaly detection is the process of identifying unusual behavior, data points, occurrences, or observations that raise concerns because they deviate from the rest of the data points or observations (Wang, Gong, Zhang, & Han, 2022; Mane et al., 2022; Taha & Hadi, 2019; Chandola, Banerjee, & Kumar, 2009; Kalinichenko, Shanin, & Taraban, 2014; Neloy & Turgeon, 2024; Raj & Sharma, 2024; Ahmed, Mahmood, & Islam, 2016). In space applications, this issue increases its importance due to its complexity and the need for a reliable system. Also, anomaly detection of the telemetry of satellites is the main key that helps in health monitoring to avoid any possible malfunction in spacecraft (Maleki Sadr, Zhu, & Hu, 2022; Bernal-Mencia, Doerksen, & Yap, 2021; He, Cheng, & Guo, 2022; Mutholib, Rahim, Gunawan, & Ahmarofi, 2024).

There are several anomaly detection techniques used depending on the type of problem, the nature of the anomalies, the number of samples available or volume of dataset to be used, the complexity of the model accepted, and finally, the accuracy needed based on the significance of the case (Osmani, Haddad, Lemenand, Castanier, & Ramadan, 2020; Chandola et al., 2009; Kalinichenko et al., 2014; Singh, Singh, Alam, & Singh, 2024; Ahmed et al., 2016). The research focuses on anomaly detection of telemetry sensor readings of nickel-hydrogen batteries used in satellites by learning the different models that were created in the training stage to detect anomalies that have a deviation from the normal one (Torabi, Mirtaheri, & Greco, 2023; Bernal-Mencia et al., 2021).

Rule-based classification, or threshold-based classification, is reducing the optimal use of the correlation among the different features or sensors. These rules are often simplistic and fail to generalize well to new, unseen data, especially if the data has complicated relationships. It would be impractical to manually create rules for all edge situations and complicated circumstances. It becomes tough to manage all of the different combinations and circumstances with if statements. The complexity of rule writing grows considerably as the number of variables rises. Capturing all possible combinations manually is not feasible in complex scenarios.

The usage of AI systems expands with the increasing complexity of the system. It demonstrated high performance in anomaly detection across various applications (Bernal-Mencia et al., 2021; Lo, Flaus, & Adrot, 2019; Al Miaari & Ali, 2023; Prakash, Venkatasubbu, & Konidena, 2023; Ma et al., 2019; Chien & Morris, 2014; Fang, Shi, Dong, Fan, & Ren, 2017). They can learn the relations among different sensors; they are better able to interpolate between complex patterns in the data and capable of detecting the abnormal behavior of the whole system. Despite the sensors working in neither fault range individually, the system might be considered faulty. This is particularly evident in the setting of high-dimensional features.

The main contribution of this paper:

- The use of real-life satellite mission data in fault detection.
- Achieving higher performance of anomaly detection by using an autoencoder as a feature engineering method that has a great influence on the enhancement of the results.
- Proving that the implementation of a hybrid model, by merging two conceptual anomaly detection methodologies, one deep learning and one statistical (autoencoder-Z-Score), fulfills better results when dealing with high-dimensional datasets that can overcome the main problem of traditional methods.
- The use of AI-based methodologies can achieve more accurate fault detection over rule-based classification.
- The use of AI-based methodologies eased the detection of the actual sensors that can be the main cause of the anomalies.

2. RELATED WORK

There are several articles and research studies that have shown the importance of and made comparisons between different methods of anomaly detection (Chliah, Battou, Laoufi, et al., 2023; Mane et al., 2022; Jung et al., 2024; Gao & Lu, 2021; Dobos et al., 2023; Chandola et al., 2009; Raj & Sharma, 2024; Elattar, Elminir, & Riad, 2016; D. Liu et al., 2021; B. Lee & Wang, 2010).

This section focuses on some common methods that can be used for anomaly detection in this research, due to their properties, performance, and compatibility with the objective of the task:

2.1. Statistical Approaches

The Z-score (or standard score) is a widely used statistical approach for detecting outliers. It is simple and has great privilege in computational times, but it has some drawbacks, such as the assumption of normal distribution, being sensitive to outliers, fixed thresholding, and not being ideal for multivariate data (Chikodili, Abdulmalik, Abisoye, & Bashir, 2020; Sardar, Pavithra, Sanjay, & Gogoi, 2022; Jung et al., 2024; Chandola et al., 2009; Marathe, 2020; Grabaskas & Si, 2017; Tanriverdiyev, 2024; Singh et al., 2024).

2.2. Machine Learning Approaches

Machine learning (ML) algorithms have high performance in detection and classification (Jan, Lee, & Koo, 2021; Mane et al., 2022; Gonzalez-Jimenez, del Olmo, Poza, Garramiola, & Sarasola, 2021; Ulmer, Zgraggen, & Huber, 2023; Prakash et al., 2023; Huč, Šalej, & Trebar, 2021).

The OCSVM is among the most popular methods because it learns from a class of data and identifies deviations as anomalies. It can handle datasets with many input features. Its performance is heavily dependent on the selection of kernel function and parameters (nu) (contamination) and (γ) (gamma) (Tennberg & Ekeroot, 2021; Taha & Hadi, 2019; Dobos et al., 2023; Erfani, Rajasegarar, Karunasekera, & Leckie, 2016).

Isolation Forest is also a good choice for anomaly detection due to its efficiency, scalability to large datasets, hhigh dimensionality,fast computational times, and robustness. To achieve the best results, the system needs proper tuning and assessment. Especially parameters like the number of trees (n. estimators) and the vital contamination factor (Kea, Han, & Kim, 2023; F. T. Liu, Ting, & Zhou, 2008; He et al., 2022; Dobos et al., 2023; D. Liu et al., 2021).

2.3. Deep Learning Approaches

Deep learning is a subset of machine learning that has been widely used for satellite anomaly detection problems. The main advantages of applying deep learning algorithms are the ability to analyze vast amounts of data generated by sensors and control inputs to find out the abnormality of the system. (Abed, Gitaffa, & Issa, 2021; Iqbal, Maniak, Doctor, & Karyotis, 2019; Wang et al., 2022; Mnyanghwalo, Kundaeli, Kalinga, & Hamisi, 2020; Mane et al., 2022; Huč et al., 2021; Singh et al., 2024; Fang et al., 2017).

One of the most ideal deep learning models is the autoencoder. It is an artificial neural network that gets two functions. An encoding function transforms input data, and a decoding function recreates input data from the encoded representation. The autoencoder finds an easy-to-use representation (encoding) for a set of information or data (Jeong et al., 2023; Kea et al., 2023; Tennberg & Ekeroot, 2021; Gao & Lu, 2021; Dobos et al., 2023; Ball, Anderson, & Chan, 2017; Neloy & Turgeon, 2024; Mosin, Staron, Tarakanov, & Durisic, 2022; Irsoy & Alpaydın, 2017; Maggipinto, Masiero, Beghi, & Susto, 2018; Sakurada & Yairi, 2014).

When constructing an autoencoder model, one should keep some things in mind. The most important hyperparameter to tune the autoencoder is the code or bottleneck size. It determines how much data needs to be compressed. When tuning autoencoders, it is important to consider the number of layers. The model has more complexity with a higher depth, but a lower depth is faster (Torabi et al., 2023; Irsoy & Alpaydın, 2017).

In recent years, several advanced deep learning architectures have been proposed for multivariate time-series anomaly detection, such as unsupervised anomaly detection USAD on multivariate time series (Audibert, Michiardi, Guyard, Marti, & Zuluaga, 2020; Chen et al., 2021), the hierarchical one-class model (Shen, Li, & Kwok, 2020; W. Zhang, Wu, Zhao, Deng, & Yang, 2022), multivariate anomaly detection for time series data with GAN (Li et al., 2019),Transformer-based multivariate time series (Shimillas, Malialis, Fokianos, & Polycarpou, 2025; Kang & Kang, 2024), and the cross-former anomaly detection model (Y. Zhang & Yan, 2023).These models demonstrate effective performance on benchmark datasets; they can detect anomalies accurately in lab-tested or open-source environments. They need a lot of processing power to operate effectively. This issue can arise in satellite systems, particularly at ground stations that require fast processing of large volumes of telemetry data or in on-board spacecraft computers, which are designed with limited hardware resources. It is hard to interpret why a decision was made. This lack of transparency and high complexity makes it difficult to be trusted in critical systems such as space applications.

Beyond technical models, several works have explored broader frameworks for industrial AI and system health management. Recent research emphasizes the significance of machine learning in diagnostics and prognostics utilizing publicly available challenge datasets, such as those reviewed by (Su & Lee, 2023). Their research demonstrates the potential of standardized anomaly detection benchmarks to improve the cross-sectoral applicability and industrial system dependability. Similar to this, the unified framework put forward in (J. Lee & Su, 2025) highlights modular architecture, interpretability, and system integration while outlining fundamental ideas for AI implementation in complex situations. By integrating hybrid AI models with actual mission telemetry, this study complies with these frameworks and shows how these ideas might be applied to the field of space systems monitoring.

3. METHODOLOGIES

The sequence of work was as follows: first, to represent and understand the system operation, preprocess the dataset, explore the data distribution of each feature, and find the relations between different features. Then select the applicable model that will achieve the best accuracy that can be obtained by evaluating the confusion matrix of each model created (Bernal-Mencia et al., 2021; Gonzalez-Jimenez et al., 2021; Dobos et al., 2023; Dhamodharan, 2022; Tanriverdiyev, 2024; Elattar et al., 2016). This overall sequence can be summarized in Figure 2. To verify the model, another dataset from another field was introduced to these models to check their effectiveness. Finally, fault detection for each anomaly observation to detect which sensor was the main cause of this anomaly; this workflow is represented in Figure . 3

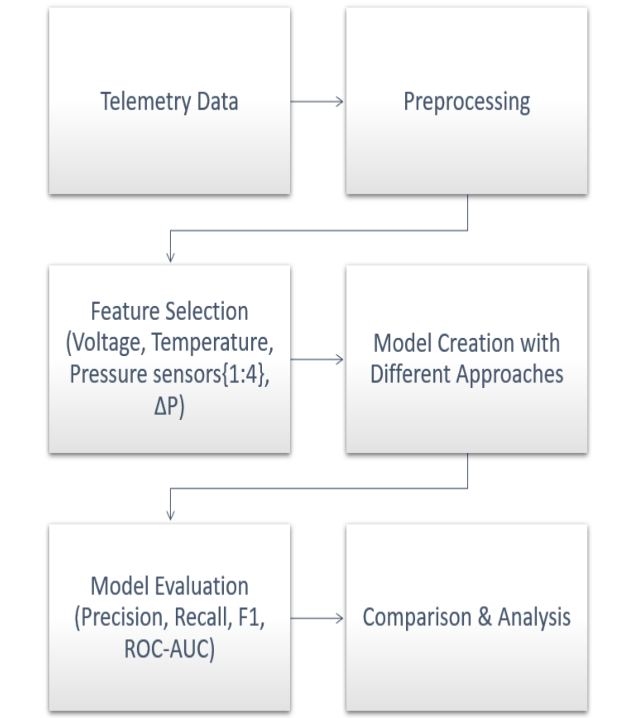

Figure 2. Overall Sequence

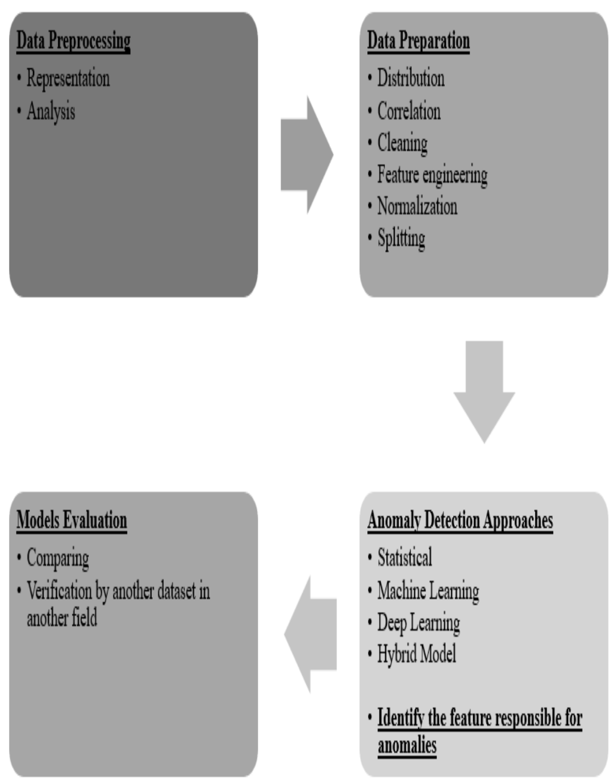

Figure 3. Anomaly Detection Workflow

3.1. Dataset

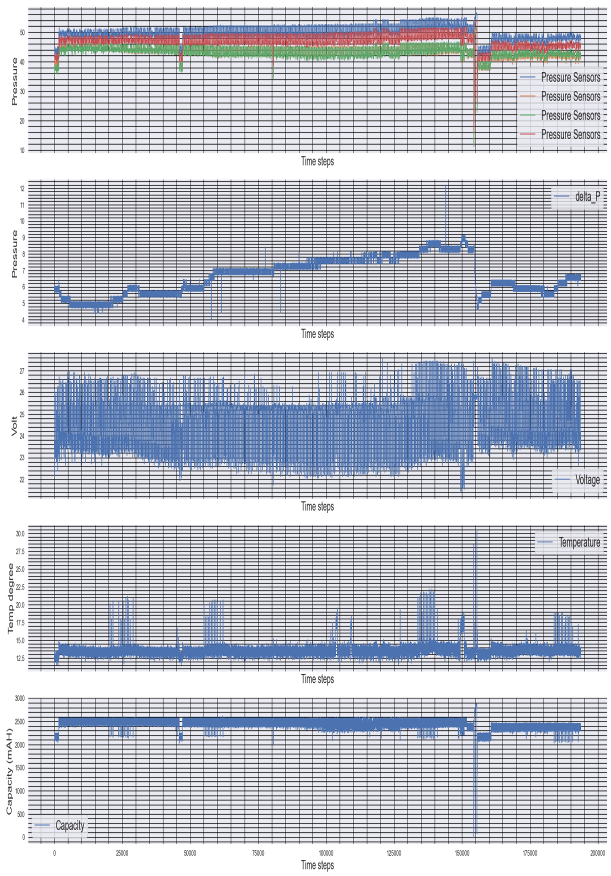

Figure 4. Data Representation

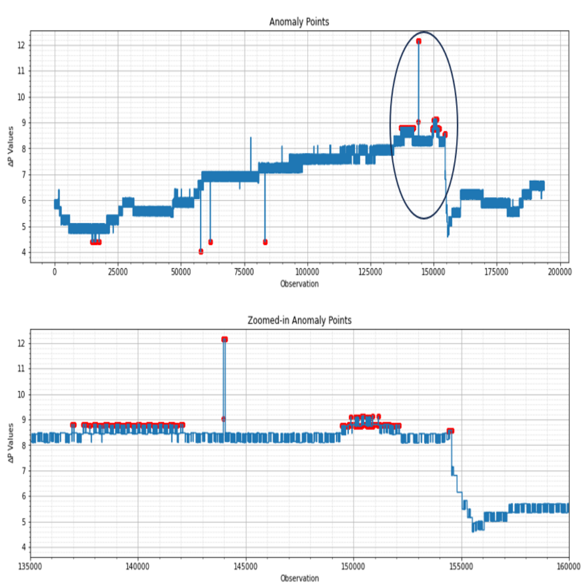

Figure 5. ΔP Signal with Annotated Anomalies

Before creating an anomaly detection model, the data should be represented and analyzed for more understanding and interpretation of the data (Mnyanghwalo et al., 2020; Chliah et al., 2023; Mane et al., 2022; He et al., 2022; Gao & Lu, 2021).

The representation shown in the Figure. 4 has the readings of the 4 pressure sensors, ΔP of these sensors, voltage reading, temperature, and capacity, respectively.

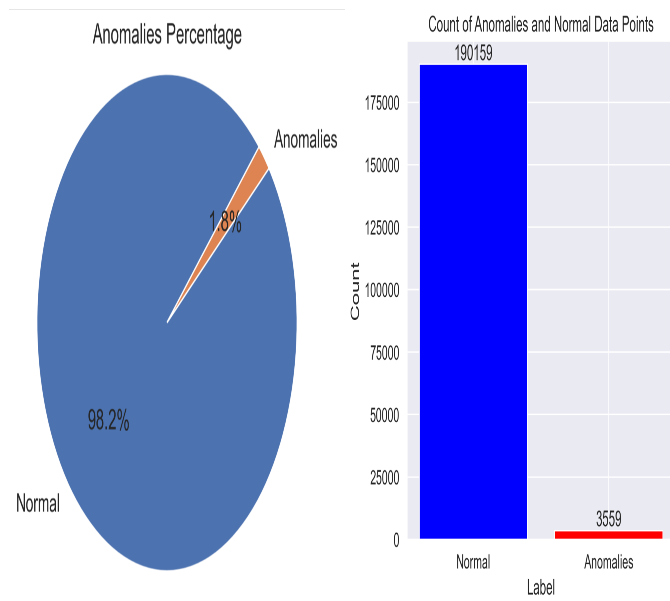

Figure 6. Anomaly points counter

According to domain knowledge, the anomalies can be represented on the (ΔP) graph as shown in Figure 5, and for more clarification, the graph can be zoomed in on a specific section to show more details along the whole period. Also, the number of anomalies in the dataset can be stated as just 1.8% of the total number of observations. i.e., there were 3559 anomaly points and 190159 normal ones. So, the minority class in the dataset was represented as rare. Here, there are 193,718 observation data points for satellite battery during one year of operation in space, including all modes of operation (shooting, downloading images, downloading telemetry, orbit correction, and recovery test). Table 1 summarizes the dataset representation and analysis.

Table 1. Dataset Summary

Dataset	No. of Points
Total no. of observations	193,718
Total Normal points	190,159
Total Anomaly points	3,559
Training points	135,602
Testing points	58,116

3.2. Data Preparation

To complete the data preprocessing, explore the data to understand its distributions along the number of observation periods as shown in Figure 7.

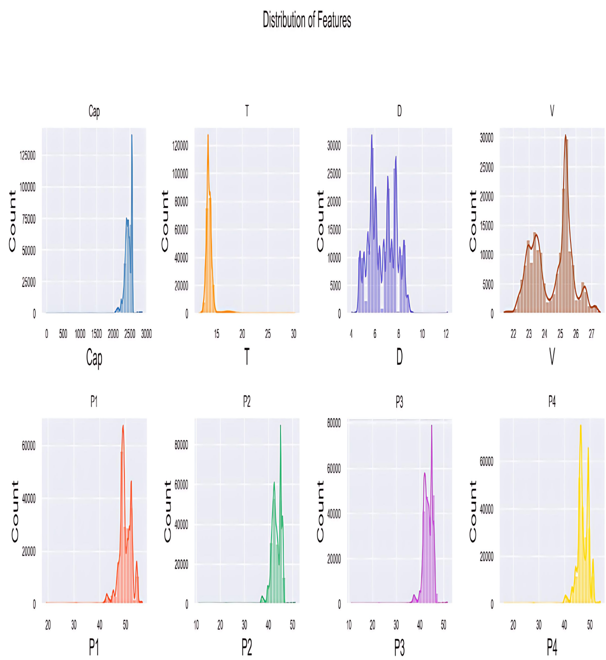

Figure 7. Data Distribution, where Cap = Capacity, T = Temperature,D= ΔP V= Voltage, and P1:P4 = 4 pressure sensors

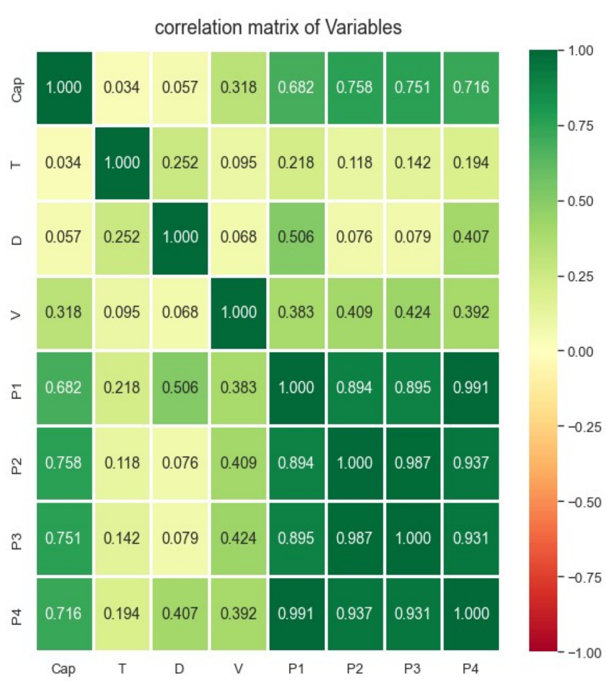

Figure 8. Feature Correlation Matrix

Use correlation matrices to determine relations between features, revealing multicollinearity or variable independence (Yang et al., 2022; Peng et al., 2014; Kim, Lee, Bhang, Choi, & Ahn, 2020). Pearson correlation heatmap among telemetry features, revealing multicollinearity and inter-sensor dependencies shown in Figure 8, there was a remarkable correlation between capacity and pressure readings and a noticeable correlation between P1-P4 and P2-P3 together that could be understood because the locations of these sensors are beside each other, as shown in Figure 1 and due to adhesion to the ADCU, which emits heat to the cell containing pressure sensors (P2, P3). That is reflected in the difference between P1-P4 and P2-P3.

It's necessary to change or adjust the data before using it in the model and correct any data quality concerns, such as missing numbers, duplication, inconsistencies, or noise. Cleaning ensures that the data is appropriate for analysis, removing the outliers, which could be due to missing telemetry during communication sessions or corrupted data (Bernal-Mencia et al., 2021; Kea et al., 2023; Mutholib et al., 2024).

Then, identify the most important model features. Feature engineering improves model behavior and computation time. In this case, voltage, temperature, four pressure sensors,and ΔP were selected to be inserted into the models after making the assessment of the data and the problem (Gonzalez-Jimenez et al., 2021; Erfani et al., 2016; Ball et al., 2017; Ahmed et al., 2016; Maggipinto et al., 2018).

Capacity readings were excluded because they were determined by the calculation of pressure sensors that were noticed in the great relation and impact obtained from the correlation shown above. i.e., using readings from sensors directly will be more reliable and accurate than readings that are calculated. Also, the current of the battery, because its value is mainly affected by the condition of the presence of the satellite in sunlight or shadow, may have a bad influence on the behavior of the model due to different values of current between negative or positive readings due to charge or discharge status.

There were 7 selected features, as mentioned above, that should be normalized/standardized, bringing them into a comparable range. Min-Max Scaling Eq.(2) would be a good choice because the scale and units of telemetry readings like temperature, voltage, and pressure vary. Also, it is very appropriate to different anomaly detection techniques. Making them standard ensures that no single sensor changes the model too much. (Kea et al., 2023; Tennberg & Ekeroot, 2021; Mutholib et al., 2024; Ali, 2022).

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \quad (2)$$

The dataset had been split to be training data of 70%, and testing data of 30% to avoid overfitting and assess how well your model works with new data (Al Miaari & Ali, 2023; Tennberg & Ekeroot, 2021; Gonzalez-Jimenez et al., 2021; Huč et al., 2021; Mutholib et al., 2024).The presence of anomaly samples in training data could influence the performance of the models, such as in real-world cases, where the anomaly may be in the training data samples (Ulmer et al., 2023; Erfani et al., 2016; Putina & Rossi, 2020). A 70/30 train-test split was applied consistently across all models to ensure a controlled and fair comparison. Although stratified or time-aware validation could provide more accuracy, the primary objective of this study was to benchmark various anomaly detection techniques under uniform conditions. The dataset includes telemetry spanning diverse operational phases, which helps decrease the risk of overfitting due to temporal leakage. Moreover, the class imbalance (1.8% anomalies) was preserved in the split to reflect realistic conditions, aligning with actual anomaly prevalence in mission telemetry.

3.3. Model Implementation

As aforementioned in Section 2, the different anomaly detection methods and their pros and cons. Next, the sequence of work will be to create a model by each of the following techniques and then compare the results and evaluate its performance.

3.3.1. Statistical Approaches

Z-Score It is one of the statistical methods that depends on getting the mean and standard deviation under the condition of normally distributed datasets. The Z-score is defined as in Eq.(3) (Wang et al., 2022; Gonzalez-Jimenez et al., 2021; Chandola et al., 2009; Raj & Sharma, 2024; Tanriverdiyev, 2024):

$$Z = \frac{X - \mu}{\sigma} \tag{3}$$

where X is the data point, μ is the mean of the dataset, and σ is the standard deviation of the dataset.

It should be used with a fixed threshold that cannot be appropriate with all datasets. Also, it cannot deal with multivariate data. So, in this case, the model was selected ΔP as the main feature. Pick the threshold of values (0.5, 1, 1.5, and 2) according to the real anomalies due to domain knowledge obtained from analyzing the dataset (Putina & Rossi, 2020).

3.3.2. Machine Learning Approaches

OCSVM model The selection of hyperparameters of the model, such as the contamination factor, was selected using random search; the value of 0.001 was found as the best value. RFB as the kernel function in mapping the inputs is most commonly used due to its robustness and flexibility when dealing with nonlinear data. Also, use gamma with scale because it is more adaptive to feature distributions and their variance and works in most cases. The main hyperparameter of the model can be summarized in Table 2.

Table 2. OCSVM Hyperparameters

Parameter	Value
Kernel Function	Radial Basis Function (RBF)
Contamination Parameter	0.001
Gamma	Scale

Isolation Forest model In this model, grid search had been used to tune its main parameters of number of estimators and contamination factor, that gave values of 50 and 0.001, respectively, as shown in Table 3.

Table 3. Isolation Forest Hyperparameters

Parameter	Value
Contamination Factor	0.001
Number of Estimators	50
Max Sample	Auto

3.3.3. Deep Learning Approaches

Deep learning models, especially autoencoders, need a large number of trainable normal data samples, and some factors should be kept in mind when creating the model; it has very critical hyperparameters to tune, and the depth and number of layers will affect the complexity and the time of the model.

Autoencoder model The model used consisted of three layers for each of the encoding and decoding of (128, 64, 32) neurons. The activation function was the Rectified Linear Unit (ReLU), which is most commonly used due to avoiding vanishing gradients. It adds non-linearity to the model so that neural networks learn complex patterns. The learning rate was 0.001 with the Adam optimizer (Al Miaari & Ali, 2023; Ball et al., 2017). It is a hyperparameter that determines how much the model's weights are altered with each iteration of training. It directly affects the convergence of the model, which eventually affects the quality and speed of learning and controls the step size for weight updates during training. Table 4 summarizes the autoencoder parameters.

Then, a new strategy of work had been performed to improve the results by using a hybrid model.

Table 4. Autoencoder Hyperparameters

Parameter	Value
Batch Size	16
Epochs	50
Activation Function	Rectified Linear Unit (ReLU)
Number of Neurons	(128, 64, 32)
Learning Rate	0.001
Optimizer	Adam
Loss Function	Mean Square Error (MSE)

3.3.4. Hybrid Approaches

Here, the autoencoder was used for feature engineering to help in dimensionality reduction that will compress the information into fewer features, which allows the model to focus on the most essential features and efficient data representation. It is better than principal component analysis (PCA) due to dealing with nonlinear data more efficiently and not distorting the core of the data (Jung et al., 2024; Sakurada & Yairi, 2014). By learning lower-dimensional features, the model can work more efficiently with reduced data, both in terms of computation and storage. Also, it can capture nonlinear relationships in the data and can be combined with other techniques (Tennberg & Ekeroot, 2021; Ball et al., 2017; Meng, Catchpoole, Skillicom, & Kennedy, 2017; Kunang, Nurmaini, Stiawan, Zarkasi, et al., 2018; Maggipinto et al., 2018). The workflow can be explained in Figure 9.

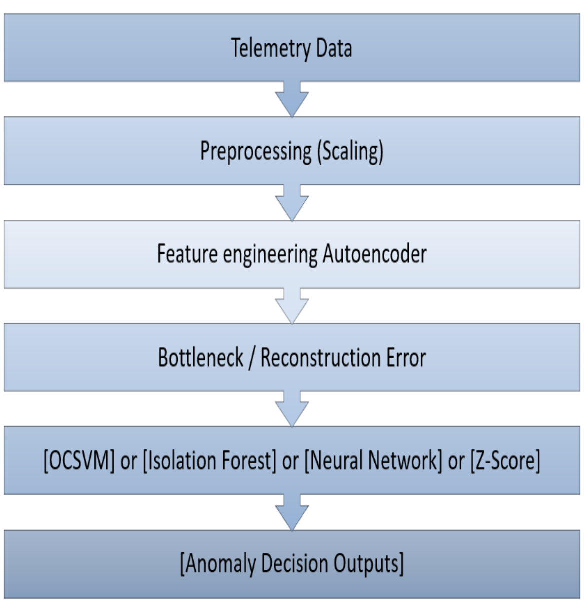

Figure 9. Hybrid Approach Pipeline

The output of the autoencoder stage was the input to:

- OCSVM
- Isolation Forest
- Neural Network
- Z-Score

3.4. Fault Detection

The final stage is to detect which feature or sensor was responsible for that anomaly; this proves that AI methodologies have a great strength of point of view because, at first glance, the primary statistical analysis considered that only ΔP was the main reason for the anomalies. Results showed that there were other sensors that had more deviation ΔP and had more influence for causing the anomalies. Creating an autoencoder means the model learns to compress and reconstruct the patterns from normal data. A large deviation in reconstruction error for a given feature suggests that the behavior of that feature is unexpected and does not match what was learned during training (Abed et al., 2021). The reconstruction error was used to detect which feature had the highest deviation after the anomaly detected in each observation (Jeong et al., 2023).Feature-wise reconstruction error across anomaly samples from the Autoencoder + Neural Network model, used for sensor fault attribution. This error is the difference between the actual value (using X_{test}) and the predicted/reconstructed value (X_{test_pred}) shown in Figure. 10.

Reconstruction errors by feature further help in narrowing down which element of the data is driving the anomaly. This is especially useful when working with multivariate datasets where anomalies might not be global but localized to specific features.

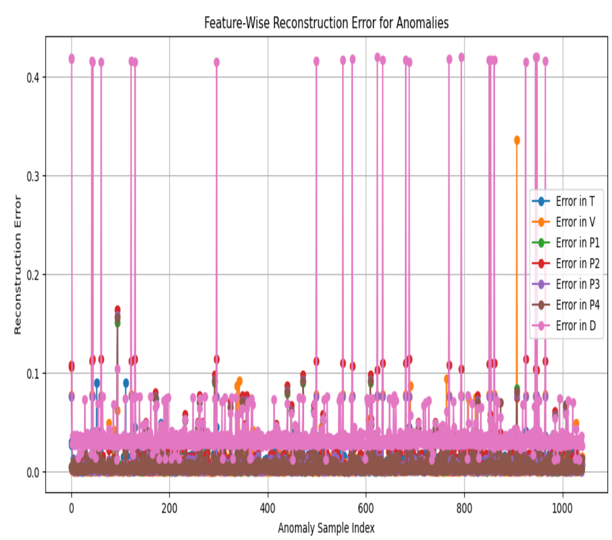

Figure 10. Per-Feature Reconstruction Error in Anomalous Observations

4. RESULTS

4.1. Comparison of performance

Create different models to achieve the best results according to evaluating each model by the confusion matrix to extract precision, recall, and F1-score. Also, AUC-ROC, which measures the model's ability to distinguish between normal and anomalous instances (Maleki Sadr et al., 2022; Abed et al., 2021; Torabi et al., 2023; Wang et al., 2022; Jan et al., 2021; Chliah et al., 2023; Kea et al., 2023; Tennberg & Ekeroot, 2021; He et al., 2022; Dobos et al., 2023; Huč et al., 2021).As shown in Figure. 11:

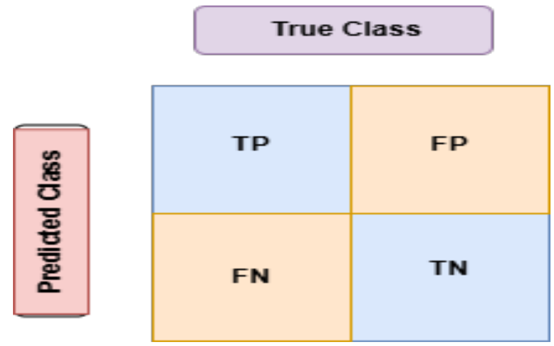

Figure 11. Confusion Matrix

Precision is the percentage of true positives out of false positives. Measures how well the model predicts the positive class, as in Eq.(4):

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

Recall is the percentage of true positives out of all actual positives. Determines the model's ability to obtain the positive class, as in Eq.(5):

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

F1 Score is the harmonic mean of precision and recall. It balances the two metrics, as in Eq.(6):

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{6}$$

FAR is defined as the proportion of normal samples incorrectly classified as anomalies.

$$\text{FAR} = \frac{FP}{FP + TN} \tag{7}$$

A receiver operating characteristic (ROC) is a graphical plot that helps assess the performance of a binary classifier.

Another crucial parameter is Time-to-Detect (TTD), which calls for an online or sequential detection system that was beyond the scope of this batch-based comparative study. TTD determination is less significant or consistent among approaches. In current approaches detection was performed on a per-sample basis without temporal accumulation or event-triggered windows; that makes TTD determination less significant or consistent among approaches.

Overall, the choice of metric depends on the specific goals and the consequences of the anomaly detection task. Practically, it is often beneficial to check multiple metrics to achieve a comprehensive understanding of the performance of the model.

4.1.1. Z-score results

The results showed a mean value (μ) of 6.6441 and a standard deviation (σ) of 1.1175. Figure 12 and Figure 13 showed that the selection of the threshold as mentioned above (0.5, 1, 1.5, 2) had a great influence on the results and the percentage of anomalies to be detected. It can be more clarified in Table 5

Table 5. Performance Metrics of Z-score with Different Thresholds

Threshold	No. of Anomalies	Precision	Recall	F1-score	ROC AUC
0.5	144,804	1.00	0.443	0.614	0.721
1.0	72,781	1.00	0.534	0.696	0.767
1.5	26,020	1.00	0.596	0.746	0.798
2.0	446	1.00	0.594	0.745	0.797

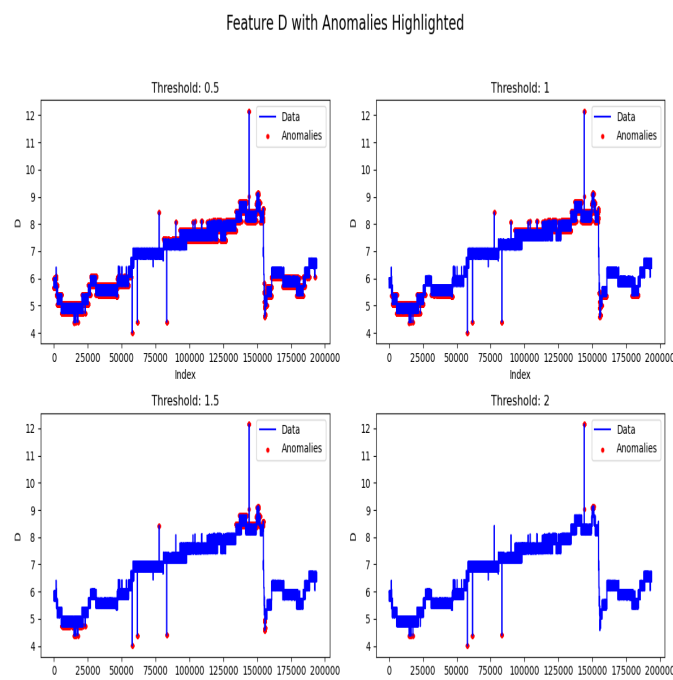

Figure 12. ΔP with Anomalies highlighted of Z-score model with different threshold

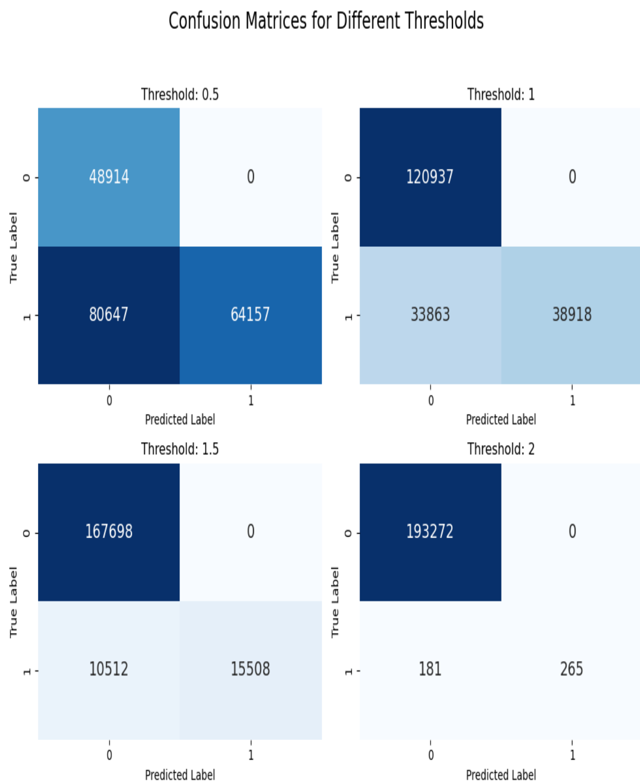

Figure 13. Results of Z-score model with different threshold

Also, the accuracy of that model was not persuasive for each threshold used. So, the Z-score is not compatible with the data distribution and not robust for multiple features.

4.1.2. OCSVM results

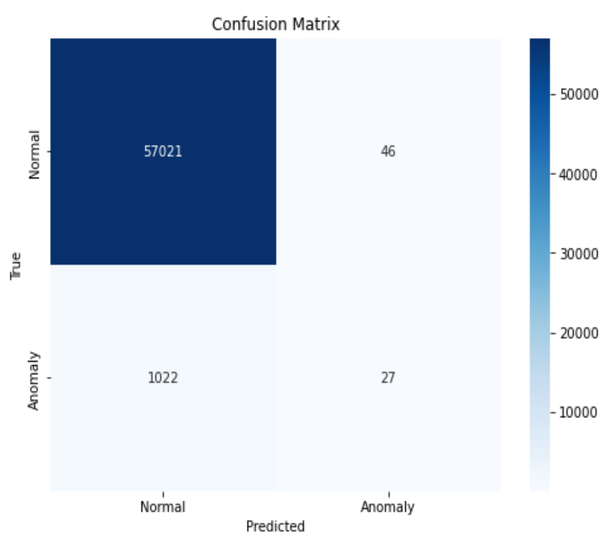

Figure 14. OCSVM Results

The results of using one-class SVM were as shown in Figure 14; the ROC is very low at 0.51, and the detection of anomalies is very poor detection = 27 from 1049 anomaly samples. But it can detect the normal with significant results. This is due to dealing with rare anomalies during training that can make the model struggle to distinguish the anomalies.

4.1.3. Isolation Forest results

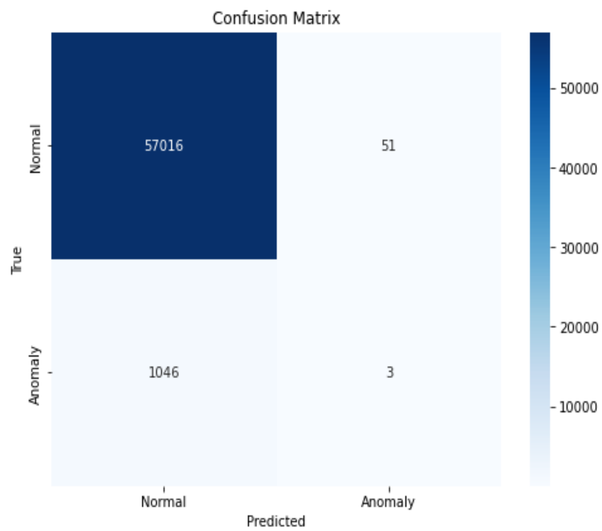

Figure 15. Isolation Forest Results

As shown in Figure 15, the ROC is very low = 0.5, and the detection of anomalies is very bad detection = 3 from 1049 anomaly samples. But it can detect the normal with notable results. Also, this is because it needs to work with datasets of well-separated anomalies.

4.1.4. Autoencoder results

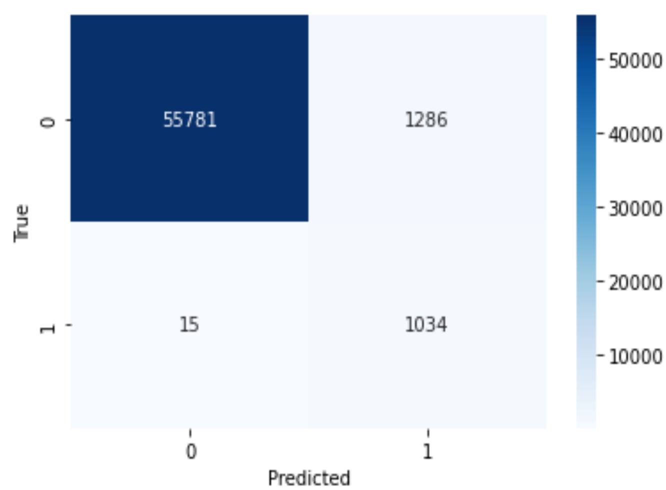

Figure 16. Autoencoder Results

The autoencoder model has noticed improvement in the results shown in Figure 16 compared to the previous models; with a high recall of 0.9857 and ROC AUC of 0.9815 but low precision of 0.4456 and a moderate F1 score of 0.613, it detected 1034 anomalies. So, this model can give hopeful results.

4.1.5. Autoencoder + OCSVM model

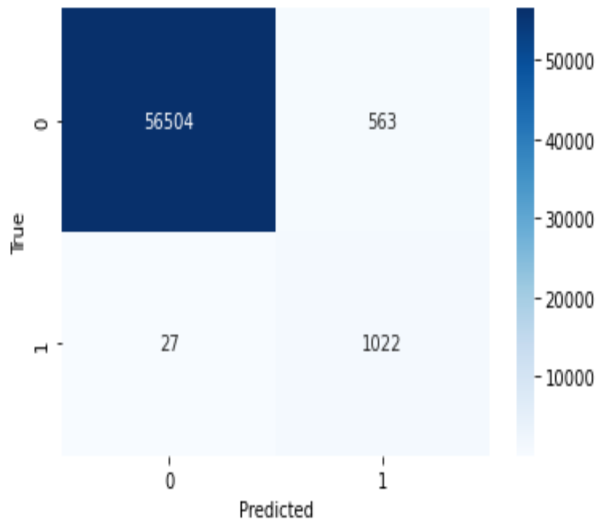

Figure 17. Autoencoder + OCSVM Results

The results in this model shown in Figure 17 were enhanced more using OCSVM only; it could predict 1022 correctly and could not detect only 27 anomalies from 1049, which gave an improvement in recall of 0.974, ROC AUC of 0.982, and F1-score of 0.776, but still insufficient precision of 0.644.

4.1.6. Autoencoder + Isolation Forest model

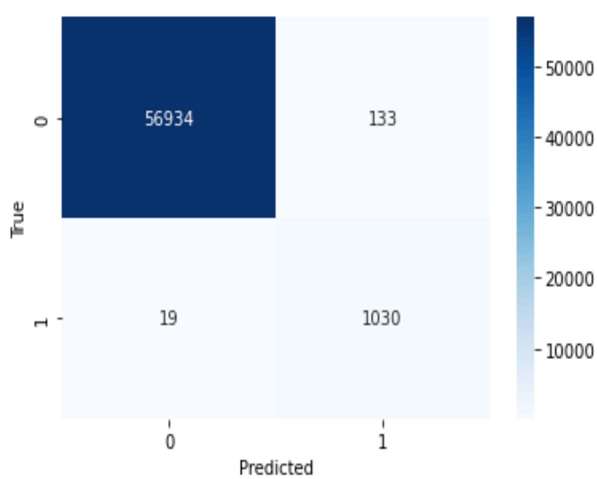

Figure 18. Autoencoder + Isolation Forest Results

The results in this model in Figure 18 have noticed more improvement using Isolation Forest only; it could predict 1030 correctly and could not detect only 19 anomalies from 1049, which gave an improvement in recall of 0.9818, ROC AUC of 0.989, and F1-score of 0.9312, and also for precision of 0.88.

4.1.7. Autoencoder + Neural Network model

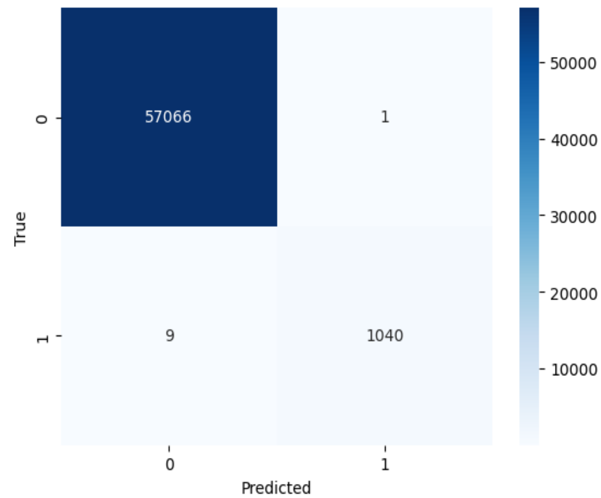

Figure 19. Autoencoder + Neural Network Results

The final anomaly decision in the Autoencoder + Neural Network model was determined by selecting a threshold of 0.5 for the sigmoid output of the final neural network layer. This threshold was selected based on widespread application in binary classification and due to achieving high precision and recall in this case. The results of this proposed model shown in Figure 19 were better than all previous models used. It could predict 1040 correctly and could not detect only 9 anomalies from 1049, with perfect accuracy in recall of 0.991, ROC AUC of 0.995, F1-score of 0.995, and very high precision of 0.999.

4.1.8. Autoencoder + Z-Score model

In this proposed model, there was a merge between two conceptual anomaly detection methodologies: one deep learning (autoencoder) and one statistical (Z-score). Autoencoder had been used as feature engineering to solve the problem of Z-score to deal with multidimensional features. The results of this model shown in Figure 20 could predict 1021 correctly and could not detect only 28 anomalies from 1049, with a recall of 0.973 and an ROC AUC of 0.9866, an F1 score of 0.983, and a precision of 0.994.

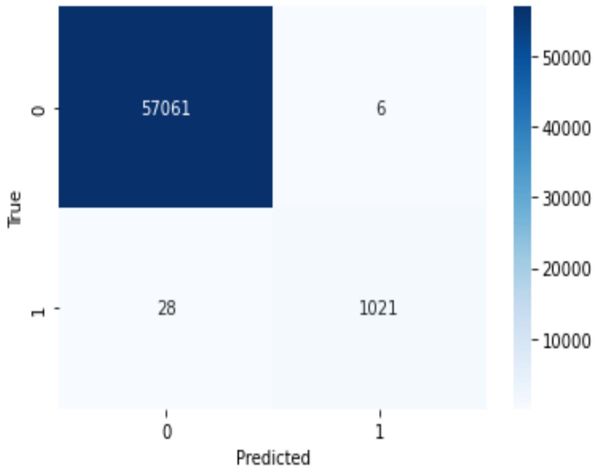

Figure 20. Autoencoder + Z-Score Results

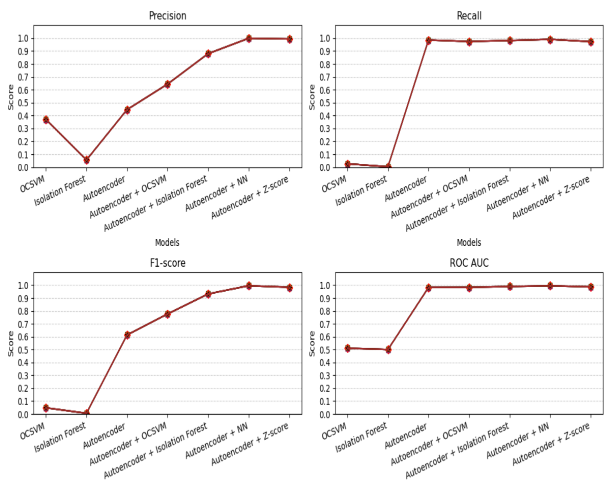

Figure 21. Comparative Performance Metrics of All Models

These results of different models are represented in Figure 21 and have been summarized in Table 6. Traditional accuracy is insufficient in this case due to an imbalanced dataset with (1.8% anomalies) that will reflect the prediction of all samples as normal, which gives an accuracy over 98%. Therefore, the accuracy was de-emphasized in the evaluation, instead focusing on more meaningful metrics such as F1-score, precision, recall, ROC AUC, and the False Alarm Rate (FAR). As shown in Table 6, traditional methods like OCSVM and Isolation Forest exhibited high FARs (0.63 and 0.94, respectively), limiting their practical use. In contrast, the proposed AE + Neural Network hybrid model achieved an FAR of 0.00096, indicating exceptional reliability in avoiding false anomaly flags during normal satellite operation.

According to these results, it was found that the proposed hybrid models, Autoencoder + NN and Autoencoder + Z-Score, have achieved the best results that could be obtained.

4.2. Cross-Dataset Evaluation on ECG Data

To check the effectiveness of these models, it is proposed to try them with other datasets (Taha & Hadi, 2019; Mokhtar et al., 2024). Next, these models were checked with different datasets in different fields of the medical section of the electrocardiogram (ECG). The electrocardiogram (ECG) dataset of 4998 patients was done with each patient having 140 data points, around 700,000 data points (Wulsin, Blanco, Mani, & Litt, 2010). As the same concept was performed previously, the dataset had also been split as training of 70 % and testing of 30 %. The ECG dataset was randomly split into 70% training and 30 % testing sets using Scikit-learn's train_test_split with a fixed random seed for reproducibility. Stratification was not used since the training set was later balanced via SMOTE oversampling to achieve class distribution parity before training the models. This can be summarized in Table 7.

Table 7. Summary of ECG Dataset

ECG Dataset	Value
Total number of observations (patients)	4,998
Data points per patient	140
Total number of data points	70,000
Training points	3,498
Testing points	1,500

The following results had been obtained:

4.2.1. Autoencoder + NN model

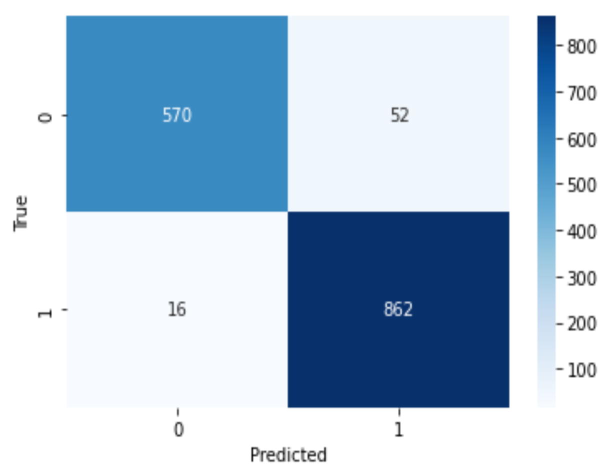

Figure 22. Autoencoder + NN Results for ECG dataset

Here the model also had excellent performance, as shown in Figure 22; it could predict 862 correctly and could not detect

Table 6. Performance Comparison of Different Models

Approach	Model	Precision	Recall	F1-score	ROC AUC	FAR
ML	OCSVM	0.3698	0.0257	0.0481	0.51	0.63
	Isolation Forest	0.0555	0.0028	0.0054	0.5	0.94
DL	Autoencoder	0.4456	0.9857	0.613	0.9815	0.554
Hybrid	Autoencoder + OCSVM	0.644	0.974	0.776	0.982	0.355
	Autoencoder + Isolation Forest	0.88	0.9818	0.9312	0.989	0.114
	Autoencoder + Neural Network	0.999	0.991	**0.995**	0.995	**0.00096**
	Autoencoder + Z-score	0.994	0.973	**0.983**	0.986	0.0058

only 16 anomalies from 878, which gave a recall of 0.9817 and an ROC AUC of 0.949, an F1-score of 0.962, and precision of 0.943.

4.2.2. Autoencoder + Z-score:

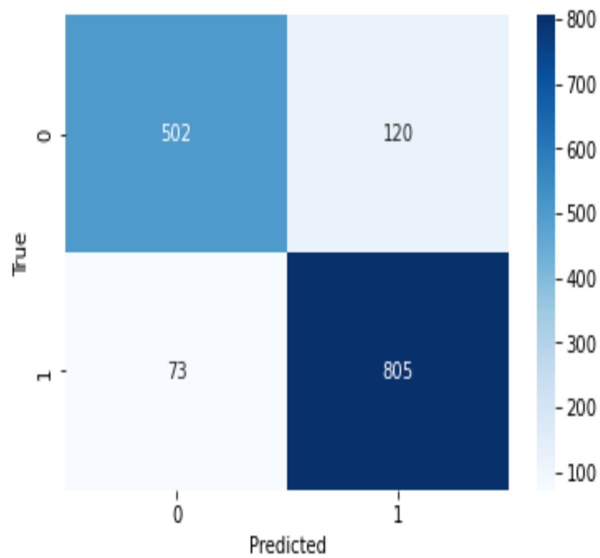

Figure 23. Autoencoder + Z-score Results for ECG dataset

The model also had high-quality results shown in Figure 23; it could predict 805 correctly and could not detect only 73 anomalies from 878, but here with a slight decrease in total accuracy that was reflected in a recall of 0.9168 and ROC AUC of 0.949, an F1-score of 0.892, and a precision of 0.87.

To conclude these results, it was found that the hybrid models had superior performance; it was noticed that there were impressive outcomes in the F1 score, which had provided a more balanced view of the model's performance by considering both precision and recall, which is useful when needing to strike a balance between detecting anomalies and minimizing false alarms (Dobos et al., 2023). It can be summarized in Table 8

Table 8. Performance of Different Models and Datasets

Model	Dataset	Precision	Recall	F1-score	FAR
Autoencoder + NN	Satellite	0.9990	0.9910	**0.9950**	0.00096
	ECG	0.9431	0.9817	0.9620	0.056
Autoencoder + Z-score	Satellite	0.9941	0.9733	0.9836	0.0058
	ECG	0.8702	0.9168	0.8929	0.129

4.3. Model Sensitivity and Analysis

Although the primary focus of this study was to compare different anomaly detection techniques using satellite battery telemetry, it is also important to consider several implementation aspects that influence real-world deployment:

First, the hyperparameter sensitivity: the effectiveness of the models was influenced by their internal configurations.

- For the Autoencoder, reducing the bottleneck size below 16 led to loss of meaningful representations, while increasing model depth beyond three layers showed minimal gains and increased training complexity.
- In the Neural Network, more than three hidden layers or too many neurons caused overfitting, especially without dropout. The model was sensitive to learning rate; values above 0.01 led to unstable training, while those lower than 0.0005 resulted in slow convergence.The number of epochs was controlled using early stopping; going significantly beyond 50 epochs often did not yield better generalization.
- For OCSVM, the contamination rate (set to 0.001) was a key factor; increasing it beyond 0.005 degraded precision considerably.
- In Isolation Forest, the number of estimators (trees) and contamination were critical. Fewer than 50 trees produced unstable results, while increasing contamination above 0.001 led to high false alarm rates. The model was also sensitive to data scaling, and performance degraded if features were not normalized.

For noise robustness in anomaly detection models refers to the ability to maintain accurate performance despite sensor noise, telemetry artifacts, or minor fluctuations in input signals. Models that are sensitive to noise may produce false positives or miss subtle anomalies in noisy environments. Uncertainty Quantification helps the models to be sure of the results of predictions, which is very critical and important for applications including autonomous onboard decision-making.

In this case, the work was mainly interested in offline mode, in that there is a preprocessing stage that cleans the noise and the corrupted telemetry, and also uncertainty can be checked by monitoring and analysis on the ground. However, to further assess resilience under degraded conditions and advanced future workflow to operate as an onboard autonomous anomaly detection model. The work will be extended by injecting synthetic noise, evaluating model stability under controlled perturbations, and integrating uncertainty-aware approaches, such as Monte Carlo dropout or confidence bounds on reconstruction errors, to quantify prediction certainty.

4.4. Feature Attribution and Fault Localization

The trained Autoencoder was used to look at the feature-wise reconstruction error for each anomalous test sample to figure out which sensors were responsible for the anomalies. For each data point predicted as anomalous by the AE + NN model, we computed the absolute reconstruction error per feature:

$$e_{i,j} = |x_{i,j} - \hat{x}_{i,j}| \tag{8}$$

where $x_{i,j}$ is the original input feature value and $\hat{x}_{i,j}$ is the reconstructed feature value for feature j in sample i. The feature (sensor) with the maximum reconstruction error in each sample was considered the most likely contributor to that anomaly. Table 9 shows the feature-wise fault attribution summary, which is based on the total number of these events across all identified anomalies.

Table 9. Feature-wise attribution of anomalies based on maximum per-feature reconstruction error among anomalous samples

Feature Name	Number of Faults
D	**985**
P2	25
P4	16
V	11
T	4

The results of the model of fault detection were in Figure 24, which shows the count of anomalies attributed to each telemetry feature based on reconstruction error analysis. Thus, this model can detect the main feature or sensor that was the main cause or reason for the fault, not as the rule-based model had detected. This will help the operators to detect the real cause of the anomaly that will have a direct impact on getting an accurate analysis for the system.

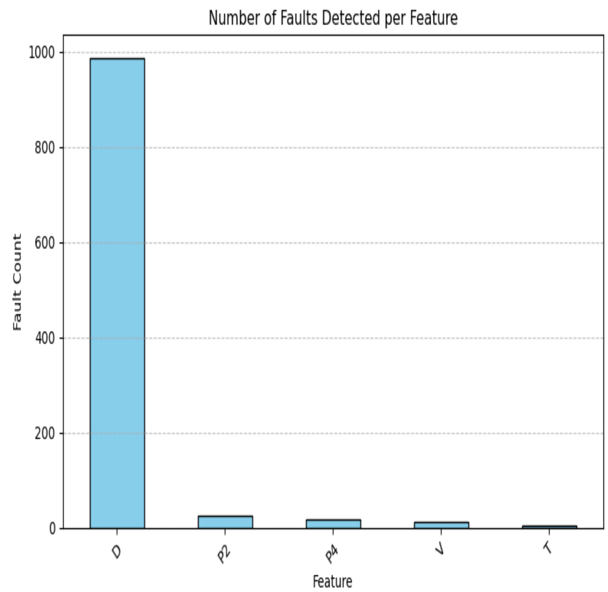

Figure 24. Fault Attribution Histogram by Feature

5. CONCLUSION

The results showed that the statistical approach (Z-Score) provided insufficient results due to dealing with high-dimensional data and has the main problem of picking the adequate threshold. The ML standalone models also performed with inappropriate accuracy. For the autoencoder, there was a noticeable enhancement in the results. Using hybrid approaches, the results were improved more than the standalone models due to the benefit realization from using the autoencoder as a feature engineering. Merging two conceptual anomaly detection methodologies, one deep learning and one statistical (autoencoder-Z-Score), carries out better results. The best output was obtained from (Autoencoder-NN). The verification of the models was fulfilled by using them with another dataset.

The proposed methodology contributes to the broader industrial AI vision by illustrating the alignment of hybrid anomaly detection frameworks, based on autoencoder representation and domain-specific context, with the foundational diagnostic principles outlined in recent literature.

AI systems achieve much better results, make the best use of the correlation between different features, and capture all possible combinations between different relations, which can be the main cause of the unexpected behavior of the system, by interpolating the complex pattern in the data. In contrast to threshold-based classification, which is very complex to manage the combination of all conditions and complicated circumstances. The complexity of rule-writing increases with increasing the number of variables. Finally, proof of concept: the expanding usage of AI methodologies in fault detection and analysis of satellite battery telemetry.

REFERENCES

Abed, A. M., Gitaffa, S. A., & Issa, A. H. (2021). Robust geophone string sensors fault detection and isolation using pattern recognition techniques based on raspberry pi4. *Mater. Today Proc S*.

Ahmed, M., Mahmood, A. N., & Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, *55*, 278–288.

Ali, P. J. M. (2022). Investigating the impact of min-max data normalization on the regression performance of k-nearest neighbor with different similarity measurements. *ARO-The Scientific Journal of Koya University*, *10*(1), 85–91.

Al Miaari, A., & Ali, H. M. (2023). Batteries temperature prediction and thermal management using machine learning: An overview. *Energy Reports*, *10*, 2277–2305.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., & Zuluaga, M. A. (2020). Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining* (pp. 3395–3404).

Balan, A. K., & Müller, K. (2015). Operational experience with nickel-hydrogen and lithium-ion batteries. In *Space2015-aiaa space and astronautics forum and exposition.*

Ball, J. E., Anderson, D. T., & Chan, C. S. (2017). Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of applied remote sensing*, *11*(4), 042609–042609.

Bernal-Mencia, P., Doerksen, K., & Yap, C. (2021). Machine learning for early satellite anomaly detection. In *Proceedings of the 35th annual small satellite conference.*

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1–58.

Chen, X., Deng, L., Huang, F., Zhang, C., Zhang, Z., Zhao, Y., & Zheng, K. (2021). Daemon: Unsupervised anomaly detection and interpretation for multivariate time series. In *2021 ieee 37th international conference on data engineering (icde)* (pp. 2225–2230).

Chien, S., & Morris, R. (2014). Space applications of artificial intelligence. *Ai Magazine*, *35*(4), 3–6.

Chikodili, N. B., Abdulmalik, M. D., Abisoye, O. A., & Bashir, S. A. (2020). 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* (pp. 259–271).

Chliah, H., Battou, A., Laoufi, A., et al. (2023). Hybrid machine learning-based approach for anomaly detection using apache spark. *International Journal of Advanced Computer Science and Applications*, *14*(4).

Dhamodharan, B. (2022). Beyond traditional methods: A novel approach to anomaly detection and classification using ai techniques. *Transactions on Latest Trends in Artificial Intelligence*, *3*(3).

Dobos, D., Nguyen, T. T., Dang, T., Wilson, A., Corbett, H., McCall, J., & Stockton, P. (2023). A comparative study of anomaly detection methods for gross error detection problems. *Computers & Chemical Engineering*, *175*, 108263.

Elattar, H. M., Elminir, H. K., & Riad, A. M. (2016). Prognostics: a literature review. *Complex & Intelligent Systems*, *2*(2), 125–154.

Erfani, S. M., Rajasegarar, S., Karunasekera, S., & Leckie, C. (2016). High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, *58*, 121–134.

Fang, H., Shi, H., Dong, Y., Fan, H., & Ren, S. (2017). Spacecraft power system fault diagnosis based on dnn. In *2017 prognostics and system health management conference (phm-harbin)* (pp. 1–5).

Gao, T., & Lu, W. (2021). Machine learning toward advanced energy storage devices and systems. *Iscience*, *24*(1).

Gonzalez-Jimenez, D., del Olmo, J., Poza, J., Garramiola, F., & Sarasola, I. (2021). Machine learning-based fault detection and diagnosis of faulty power connections of induction machines. *Energies*, *14*(16), 4886.

Grabaskas, N., & Si, D. (2017). Anomaly detection from kepler satellite time-series data. In *Machine learning and data mining in pattern recognition: 13th international conference, mldm 2017, new york, ny, usa, july 15-20, 2017, proceedings 13* (pp. 220–232).

He, J., Cheng, Z., & Guo, B. (2022). Anomaly detection in satellite telemetry data using a sparse feature-based method. *Sensors*, *22*(17), 6358.

Huč, A., Šalej, J., & Trebar, M. (2021). Analysis of machine learning algorithms for anomaly detection on edge devices. *Sensors*, *21*(14), 4946.

Iqbal, R., Maniak, T., Doctor, F., & Karyotis, C. (2019). Fault detection and isolation in industrial processes using deep learning approaches. *IEEE Transactions on Industrial Informatics*, *15*, 3077–3084.

Irsoy, O., & Alpaydın, E. (2017). Unsupervised feature extraction with autoencoder trees. *Neurocomputing*, *258*, 63–73.

Jan, S. U., Lee, Y. D., & Koo, I. S. (2021). A distributed sensor-fault detection and diagnosis framework using machine learning. *Information Sciences*, *547*, 777–796.

Jeong, S.-K., Ji, D.-H., Oh, M., Park, H., Baeg, S., & Lee, J. (2023). A study on anomaly detection of unmanned marine systems using machine learning. *Measurement and Control*, *56*(3-4), 470–480.

Jung, S., Kim, M., Kim, E., Kim, B., Kim, J., Cho, K.-H., ... Kim, S. (2024). The early detection of faults for lithium-ion batteries in energy storage systems using independent component analysis with mahalanobis distance. *Energies*, *17*(2), 535.

Kalinichenko, L., Shanin, I., & Taraban, I. (2014). Methods for anomaly detection: A survey. In *Ceur workshop proceedings* (Vol. 1297, p. 2025).

Kang, H., & Kang, P. (2024). Transformer-based multivariate time series anomaly detection using inter-variable attention mechanism. *Knowledge-Based Systems*, *290*, 111507.

Kea, K., Han, Y., & Kim, T.-K. (2023). Enhancing anomaly detection in distributed power systems using autoencoder-based federated learning. *Plos one*, *18*(8), e0290337.

Kim, G. G., Lee, W., Bhang, B. G., Choi, J. H., & Ahn, H.-K. (2020). Fault detection for photovoltaic systems using multivariate analysis with electrical and environmental variables. *IEEE Journal of photovoltaics*, *11*(1), 202–212.

Kunang, Y. N., Nurmaini, S., Stiawan, D., Zarkasi, A., et al. (2018). Automatic features extraction using autoencoder in intrusion detection system. In *2018 international conference on electrical engineering and computer science (icecos)* (pp. 219–224).

Lee, B., & Wang, X. (2010). Fault detection and reconstruction for micro-satellite power subsystem based on pca. In *2010 3rd international symposium on systems and control in aeronautics and astronautics* (pp. 1169–1173).

Lee, J., & Su, H. (2025). Rethinking industrial artificial intelligence: a unified foundation framework. *arXiv preprint arXiv:2504.01797*.

Lewin, A. (1999). A maintenance-free battery charge control approach for nickel-hydrogen batteries. In *Proceedings of the 13th aiaa/usu conference on small satellites.*

Li, D., Chen, D., Jin, B., Shi, L., Goh, J., & Ng, S.-K. (2019). Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International conference on artificial neural networks* (pp. 703–716).

Liu, D., Zhen, H., Kong, D., Chen, X., Zhang, L., Yuan, M., & Wang, H. (2021). Sensors anomaly detection of industrial internet of things based on isolated forest algorithm and data compression. *Scientific Programming*, *2021*(1), 6699313.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413–422).

Lo, N. G., Flaus, J.-M., & Adrot, O. (2019). Review of machine learning approaches in fault diagnosis applied to iot systems. In *2019 international conference on control, automation and diagnosis (iccad)* (pp. 1–6).

Lou, Y. (2021). *Hybrid approaches of battery performance modeling and prognosis* (Unpublished doctoral dissertation). University of Michigan.

Ma, K. Y., Montoro, M., Shaw, H., Mihir, P., Woods, L., Williams, T., ... others (2019). Utilization of unsupervised anomalies detector as a tool for managing the tdrs constellation at gsfc. In *International astronautical congress (iac).*

Maggipinto, M., Masiero, C., Beghi, A., & Susto, G. A. (2018). A convolutional autoencoder approach for feature extraction in virtual metrology. *Procedia Manufacturing*, *17*, 126–133.

Maleki Sadr, M. A., Zhu, Y., & Hu, P. (2022). An anomaly detection method for satellites using monte carlo dropout. *arXiv e-prints*, arXiv–2211.

Mane, D. T., Sangve, S., Upadhye, G., Kandhare, S., Mohole, S., Sonar, S., & Tupare, S. (2022). Detection of anomaly using machine learning: a comprehensive survey. *Int. J. Emerg. Technol. Adv. Eng.*, *12*(11), 134–152.

Marathe, A. P. (2020). Lrz convolution: an algorithm for automatic anomaly detection in time-series data. In *Proceedings of the 32nd international conference on scientific and statistical database management* (pp. 1–12).

Meng, Q., Catchpoole, D., Skillicom, D., & Kennedy, P. J. (2017). Relational autoencoder for feature extraction. In *2017 international joint conference on neural networks (ijcnn)* (pp. 364–371).

Mnyanghwalo, D., Kundaeli, H., Kalinga, E., & Hamisi, N. (2020). Deep learning approaches for fault detection and classifications in the electrical secondary distribution network: Methods comparison and recurrent neural network accuracy comparison. *Cogent Engineering*, *7*(1), 1857500.

Mokhtar, A., Ibrahim, M., Hanafy, M. E., ElTohamy, F. H. A., Murtada, W. A., & Elhalwagy, Y. Z. (2024). A reliable spacecraft power supply subsystem based on discrete water cycle multi-objective optimization. *Franklin Open*, *7*, 100092.

Mosin, V., Staron, M., Tarakanov, Y., & Durisic, D. (2022). Comparing autoencoder-based approaches for anomaly detection in highway driving scenario images. *SN Applied Sciences*, *4*(12), 334.

Mostacciuolo, E., Vasca, F., Baccari, S., Iannelli, L., Sagnelli, S., Luisi, R., & Stanzione, V. (2019). Fault analysis to improve reliability of a leo satellite eps. *IFAC-PapersOnLine*, *52*(12), 200–205.

Mutholib, A., Rahim, N. A., Gunawan, T. S., & Ahmarofi,

A. A. (2024). Performance comparison of data preprocessing methods for trade-space exploration with ai model: Case study of satellite anomalies detection. In *2024 ieee 10th international conference on smart instrumentation, measurement and applications (icsima)* (pp. 271–275).

Neloy, A. A., & Turgeon, M. (2024). A comprehensive study of auto-encoders for anomaly detection: Efficiency and trade-offs. *Machine Learning with Applications*, 100572.

Osmani, K., Haddad, A., Lemenand, T., Castanier, B., & Ramadan, M. (2020). Material-based fault detection methods for pv systems. In *Key engineering materials* (Vol. 865, pp. 111–115).

Peng, J., Fan, L., Xiao, W., & Tang, J. (2014). Anomaly monitoring method for key components of satellite. *The Scientific World Journal*, *2014*(1), 104052.

Prakash, S., Venkatasubbu, S., & Konidena, B. K. (2023). From burden to advantage: Leveraging ai/ml for regulatory reporting in us banking. *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)*, *1*(1), 167–176.

Purushothaman, K., Binu, B., Philip, J., Pillai, S., & Ilangovan, S. (2011). Acoustic emission monitoring and evaluation of nickel hydrogen battery cell cases. In *Proceedings of the national seminar & exhibition on non-destructive evaluation (nde).*

Putina, A., & Rossi, D. (2020). Online anomaly detection leveraging stream-based clustering and real-time telemetry. *IEEE Transactions on Network and Service Management*, *18*(1), 839–854.

Raj, A., & Sharma, S. (2024). A comprehensive study on anomaly detection methods using traditional and machine learning approaches. *International Journal of High School Research*, *6*(11).

Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the mlsda 2014 2nd workshop on machine learning for sensory data analysis* (pp. 4–11).

Sardar, B. K., Pavithra, S., Sanjay, H., & Gogoi, P. (2022). Determining stock market anomalies by using optimized z-score technique on clusters obtained from k-means. In *Recent advances in artificial intelligence and data engineering: Select proceedings of aide 2020* (pp. 401–414).

Shen, L., Li, Z., & Kwok, J. (2020). Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in neural information processing systems*, *33*, 13016–13026.

Shimillas, C., Malialis, K., Fokianos, K., & Polycarpou, M. M. (2025). Transformer-based multivariate time series anomaly localization. In *2025 ieee symposium on computational intelligence on engineering/cyber physical systems (cies)* (pp. 1–8).

Singh, A., Singh, S. P., Alam, M. N., & Singh, G. (2024). Deep learning for anomaly detection in iot systems: Techniques applications and future directions. *International Journal For Multidisciplinary Research*.

Su, H., & Lee, J. (2023). Machine learning approaches for diagnostics and prognostics of industrial systems using open source data from phm data challenges: a review. *arXiv preprint arXiv:2312.16810*.

Taha, A., & Hadi, A. S. (2019). Anomaly detection methods for categorical data: A review. *ACM Computing Surveys (CSUR)*, *52*(2), 1–35.

Tanriverdiyev, E. (2024). Analysis of computer network statistics for identifying stability-disrupting information flows in military local networks. *Studia Bezpieczeństwa Narodowego*, *34*(4), 77–89.

Tennberg, M., & Ekeroot, L. (2021). *Anomaly detection on satellite time-series.*

Thaller, L. H., & Zimmerman, A. H. (2003). *Overview of the design, development, and application of nickel-hydrogen batteries* (Technical Report). National Aeronautics and Space Administration (NASA).

Torabi, H., Mirtaheri, S. L., & Greco, S. (2023). Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*, *6*(1), 1.

Ulmer, M., Zgraggen, J., & Huber, L. G. (2023). A generic machine learning framework for fully-unsupervised anomaly detection with contaminated data. *arXiv preprint arXiv:2308.13352*.

Wang, Y., Gong, J., Zhang, J., & Han, X. (2022). A deep learning anomaly detection framework for satellite telemetry with fake anomalies. *International Journal of Aerospace Engineering*, *2022*(1), 1676933.

Wulsin, D., Blanco, J., Mani, R., & Litt, B. (2010). Semi-supervised anomaly detection for eeg waveforms using deep belief nets. In *2010 ninth international conference on machine learning and applications* (pp. 436–441).

Yang, S., Hu, S., Zhao, J., Cui, H., Wang, Y., Zhao, S., ... Dong, Z. (2022). Prediction on discharging properties of nickel–manganese materials for high-performance sodium-ion batteries via machine learning methods. *Energy Technology*, *10*(11), 2200733.

Zhang, W., Wu, Q. J., Zhao, W. W., Deng, H., & Yang, Y. (2022). Hierarchical one-class model with subnetwork for representation learning and outlier detection. *IEEE Transactions on Cybernetics*, *53*(10), 6303–6316.

Zhang, Y., & Yan, J. (2023). Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations.*