Anomaly Detection in Spacecraft Telemetry Data using LSTM Autoencoders

Dhinakar R
Blekinge Institute of Technology - Karlskrona, Sweden - dhra24@student.bth.se

***Abstract*—This paper implements and evaluates multiple deep learning approaches for anomaly detection in spacecraft telemetry data from the NASA SMAP and MSL dataset. We design and implement two LSTM-based autoencoder architectures: a Simple LSTM Autoencoder and a Bidirectional LSTM Autoencoder, comparing them against a baseline SuperIsolationForest model. Additionally, we develop a novel Hybrid Model that combines the strengths of both LSTM and Isolation Forest approaches. Experimental results on the M-6 channel demonstrate that while individual models achieve either high recall or high precision, our Hybrid Model with $\alpha = 0.9$ achieves remarkable performance with perfect precision (1.0) and high recall (0.944), resulting in an F1-score of 0.971. We analyze reconstruction error distributions, feature engineering impacts, and threshold selection methods to provide insights into effective anomaly detection for spacecraft telemetry data.**

***Index Terms*—anomaly detection, LSTM autoencoders, time-series analysis, spacecraft telemetry, deep learning, isolation forest, hybrid models**

I. Introduction

Detecting anomalies in spacecraft telemetry data is critical for ensuring mission success and preventing catastrophic failures in space missions. Anomalies in this context represent unusual patterns or outliers that may indicate system faults, unexpected events, or other irregular behaviors. The challenge is particularly complex because spacecraft generate multivariate time-series data with complex interdependencies, normal operation includes various cyclic patterns and seasonal variations, anomalies are rare (creating significant class imbalance), and the consequences of missed anomalies can be mission-critical.

This paper focuses on implementing and evaluating deep learning models for anomaly detection in the NASA SMAP and MSL dataset, specifically targeting the M-6 channel data. We design two LSTM-based autoencoder architectures and compare them with a classical machine learning approach (SuperIsolationForest) to identify the most effective techniques for this domain.

Our work examines:

- The effectiveness of reconstruction-based anomaly detection
- The impact of bidirectional processing in LSTM autoencoders
- Feature engineering techniques for classical anomaly detection
- Hybrid modeling approaches that combine complementary strengths
- Threshold selection methods for optimal precision-recall balance

Through this comparison, we aim to develop robust anomaly detection systems that can provide early warning of potential spacecraft failures while minimizing false alarms.

II. Dataset and Preprocessing

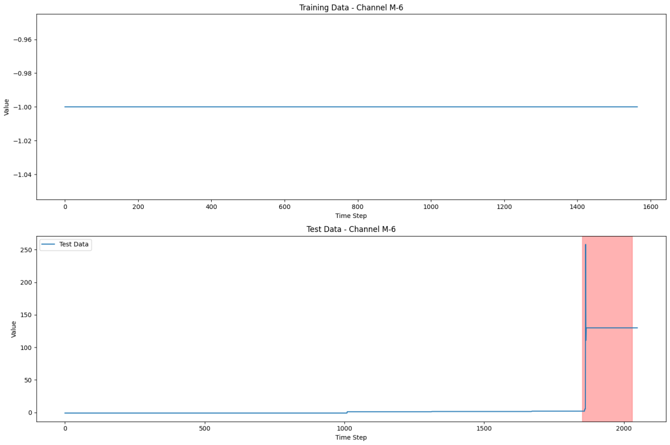

Fig. 1. Training and test data for Channel M-6. The top panel shows the stable training data, while the bottom panel displays test data with the anomalous region highlighted in red. Note the significant spike in the anomaly region around time step 1900.

A. NASA SMAP & MSL Dataset

The dataset contains telemetry data from two NASA spacecraft missions: the Soil Moisture Active Passive (SMAP) satellite and the Mars Science Laboratory (MSL) rover. For this study, we focused specifically on the M-6 channel, which demonstrated clearer anomaly patterns compared to other channels like P-1. The dataset has the following characteristics:

- Number of Features: 60
- Sequence Length: 100
- Number of Training Samples: 1,172
- Number of Test Samples: 1,950
- Anomaly Percentage in Test Data: 10.00%

The NASA SMAP and MSL dataset presents several unique challenges:

- Class Imbalance: Anomalies represent only 10% of the test data
- Complex Temporal Dependencies: Anomalies may develop over extended periods

- Multivariate Relationships: Anomalies often manifest across multiple sensor readings
- Seasonal and Cyclic Patterns: Normal operation includes various repeating patterns
- Noise and Interference: Space environment introduces various forms of signal noise

As shown in Fig. 1, the training data exhibits relatively stable patterns while the test data contains clear anomalous regions, making this an interesting case study for anomaly detection.

B. Preprocessing Pipeline

We implemented a comprehensive preprocessing pipeline consisting of several key steps:

1) Handling Missing Values: The NASA dataset contains some missing values represented as NaN. We used a forward-fill approach followed by a backward-fill to handle these missing values:

```
Convert to pandas DataFrame
Forward fill missing values
Backward fill any remaining missing values
Fill any still-remaining NaNs with zeros
```

2) Normalization: We applied feature-wise Min-Max normalization to scale all features to the [0, 1] range, computing normalization parameters from the training data and applying them to both training and test sets:

```
Compute min and max values from training data
Normalize training data: $(data - min)/(max - min)$
Apply same normalization to test data
Clip test data values to $[0, 1]$ range
```

3) Sequence Generation: We transformed the data into overlapping sequences of length 100 for time-series analysis:

```
for $i = 0$ to $len(data) - seq\_length + 1$ with step 1 do
    Extract sequence $data[i : i + seq\_length]$
    Add to sequences list
end for
```

4) Feature Selection: Based on feature importance analysis, we identified the most relevant features for anomaly detection by calculating variance for each feature and selecting those above a threshold of 0.01, reducing the feature dimension from 60 to 48.

5) Train-Validation Split: We used an 80-20 train-validation split for model development.

6) Data Augmentation: We implemented limited data augmentation to enhance model robustness:

- Jittering: Added small random noise to training sequences
- Scaling: Applied minor random scaling to sequences

After preprocessing, the dataset was transformed into:

- Training Sequences: 2,344 (after augmentation)
- Validation Sequences: 293
- Test Sequences: 1,950
- Feature Dimension: 48 (after feature selection)

III. Methodology

A. Simple LSTM Autoencoder

The first architecture implements a traditional autoencoder structure with LSTM layers.

1) Architecture: The architecture consists of an encoder network that compresses the input data into a lower-dimensional latent representation, followed by a decoder network that reconstructs the original input:

- **Encoder:**
  - LSTM (128 units, return sequences=True)
  - Dropout (0.2)
  - LSTM (64 units)
- **Latent Space:**
  - Dense (16 units)
- **Decoder:**
  - RepeatVector (100)
  - LSTM (64 units, return sequences=True)
  - Dropout (0.2)
  - LSTM (128 units, return sequences=True)
  - TimeDistributed Dense (48 units)

The model contains 166,608 trainable parameters.

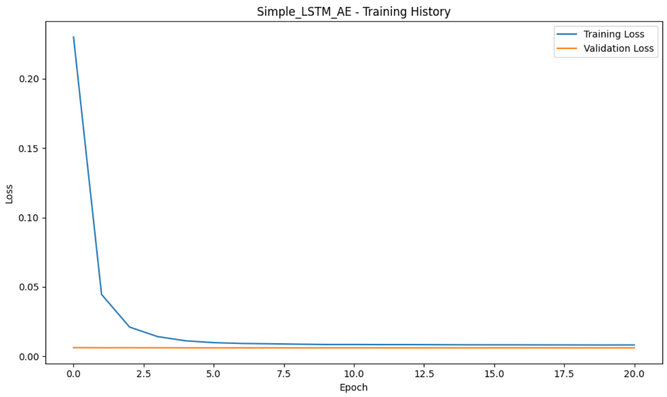

Fig. 2. Training history of the Simple LSTM Autoencoder showing rapid convergence of both training and validation loss within the first 5 epochs, indicating efficient learning of normal patterns.

2) Training Configuration:

- Optimizer: Adam with learning rate 0.001
- Loss Function: Mean Squared Error (MSE)
- Batch Size: 32
- Epochs: 100
- Early Stopping: Patience of 10 epochs, monitoring validation loss
- Callbacks: ReduceLROnPlateau (factor=0.5, patience=5)
- Training Strategy: Self-supervised reconstruction of input sequences

Fig. 2 shows the training history of the Simple LSTM Autoencoder, demonstrating efficient learning and convergence.

3) Anomaly Detection Strategy: This model detects anomalies by computing the reconstruction error for each time step:

- Get reconstructions by passing test sequences through the model

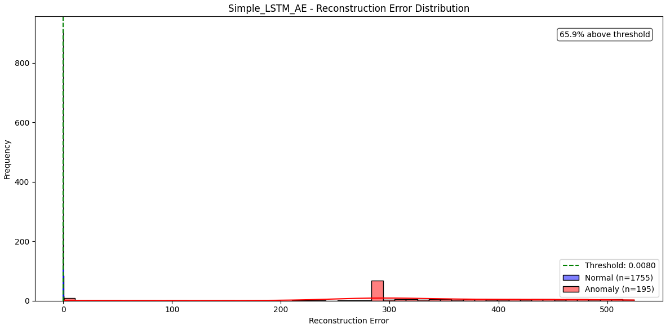

Fig. 3. Reconstruction error distribution for the Simple LSTM Autoencoder. Note the clear separation between normal samples (blue) and anomalous samples (red), but with a significant overlap near the threshold boundary.

- Calculate mean squared error between original sequences and reconstructions
- Determine anomaly threshold using statistical properties of errors
- Flag sequences with reconstruction error above threshold as anomalies

Fig. 3 shows the distribution of reconstruction errors, with a clear separation between normal and anomalous samples, though with some overlap at the boundary.

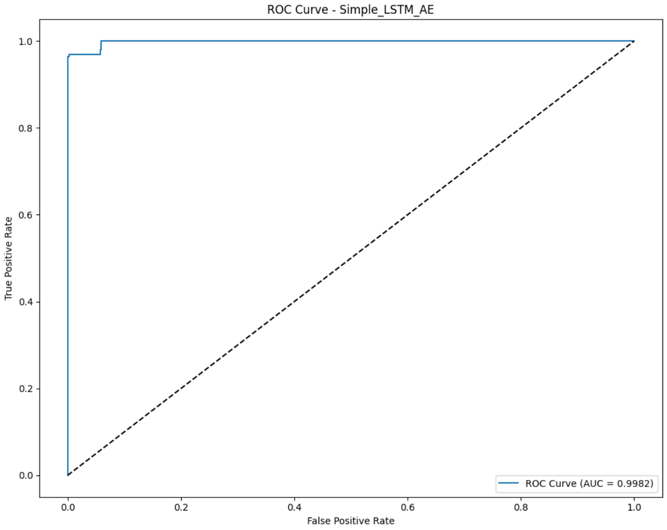

Fig. 4. ROC curve for the Simple LSTM Autoencoder showing near-perfect anomaly ranking performance with an AUC of 0.9982.

The ROC curve in Fig. 4 demonstrates the excellent anomaly ranking capability of the Simple LSTM Autoencoder, with an AUC of 0.9982.

B. Bidirectional LSTM Autoencoder

The second architecture enhances the first by using bidirectional LSTM layers in the encoder.

1) Architecture:

- **Encoder:**

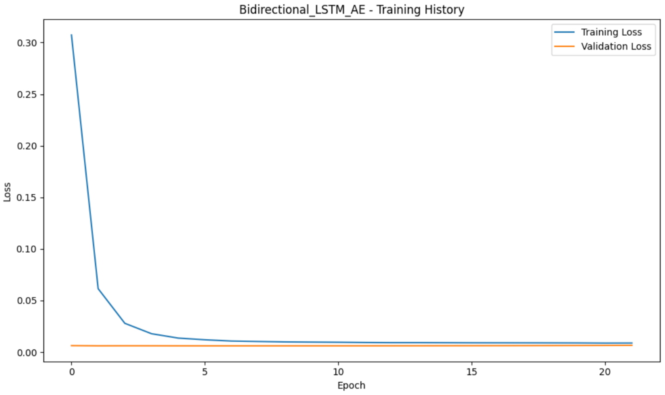

Fig. 5. Training history of the Bidirectional LSTM Autoencoder showing similar convergence pattern to the Simple LSTM model.

  - Bidirectional LSTM (96 units, return sequences=True)
  - Dropout (0.2)
  - Bidirectional LSTM (64 units)
- **Latent Space:**
  - Dense (12 units)
- **Decoder:**
  - RepeatVector (100)
  - LSTM (64 units, return sequences=True)
  - Dropout (0.2)
  - LSTM (96 units, return sequences=True)
  - TimeDistributed Dense (48 units)

The model contains 159,360 trainable parameters.

2) Key Differences from Simple LSTM:

- Bidirectional processing in encoder layers
- Smaller latent dimension (12 vs. 16)
- Different unit configuration (96 vs. 128)
- More aggressive learning rate reduction (factor=0.4 vs. 0.5)
- Parameter efficiency: Despite bidirectional layers, total parameters are reduced

Fig. 5 shows the training history of the Bidirectional LSTM Autoencoder, which follows a similar pattern to the Simple LSTM model but with slightly faster convergence.

3) Anomaly Detection Strategy: The anomaly detection strategy is similar to the Simple LSTM Autoencoder but with a more adaptive threshold calculation using median absolute deviation around the mode of the reconstruction error distribution.

Fig. 6 displays the reconstruction error distribution for the Bidirectional LSTM model, with clear separation between normal and anomalous points.

The ROC curve in Fig. 7 demonstrates the strong anomaly ranking capability of the Bidirectional model.

C. Baseline: SuperIsolationForest

As a baseline, we implemented an enhanced Isolation Forest with additional feature engineering.

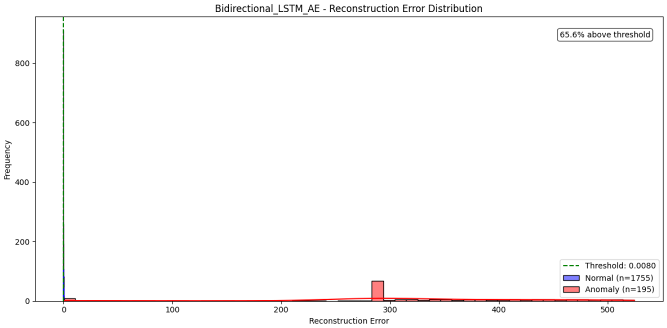

Fig. 6. Reconstruction error distribution for the Bidirectional LSTM Autoencoder showing similar patterns to the Simple LSTM model, with 65.6% of samples above the threshold.

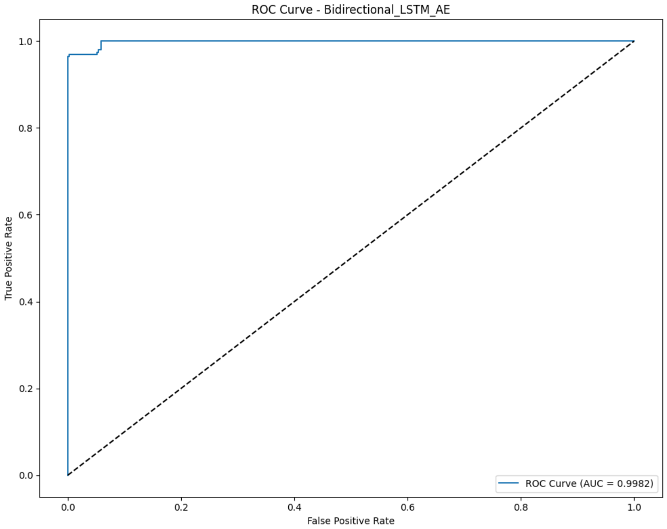

Fig. 7. ROC curve for the Bidirectional LSTM Autoencoder showing excellent performance with an AUC of 0.9982.

1) Conceptual Background: The Isolation Forest algorithm works on the principle that anomalies are ”few and different,” making them easier to isolate than normal points. The algorithm:

- Randomly selects a feature
- Randomly selects a split value between the minimum and maximum values of the selected feature
- Recursively partitions the data until all points are isolated
- Computes an anomaly score based on the average path length required to isolate each point

Anomalies typically require fewer splits to isolate, resulting in shorter path lengths and higher anomaly scores.

2) Feature Engineering: The model extracts 288 features from the original sequences:

- Statistical features: mean, standard deviation, min, max (48 × 4 = 192 features)
- Trend features: linear slopes for each original feature (48 features)
- Autocorrelation: lag-1 autocorrelation for each feature (48 features)

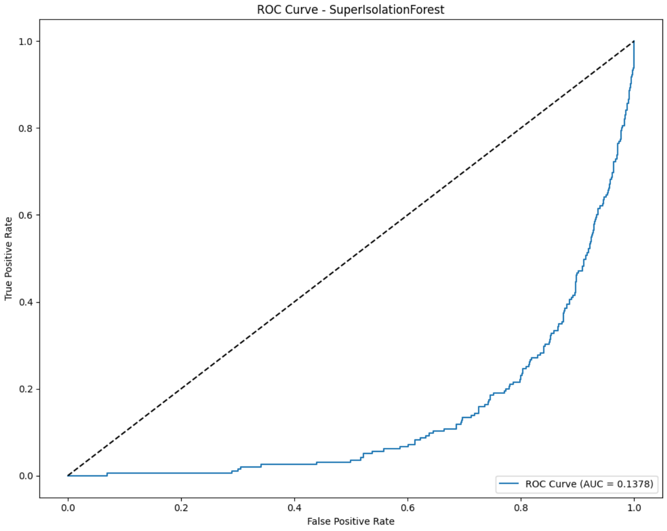

Fig. 8. ROC curve for the SuperIsolationForest model showing relatively poor performance with an AUC of only 0.1378, significantly worse than the LSTM-based models.

3) Implementation Details:

- Base estimators: 200
- Contamination parameter: 0.1
- Maximum samples per estimator: 256
- Anomaly threshold: 90th percentile of anomaly scores

Fig. 8 shows the ROC curve for the SuperIsolationForest model, demonstrating its significantly weaker performance compared to the LSTM-based approaches.

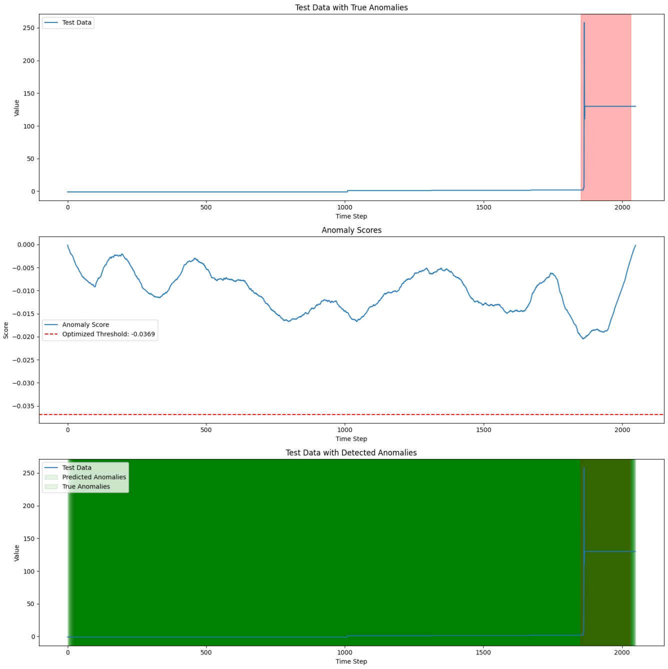

Fig. 9. Anomaly detection visualization for the SuperIsolationForest model. Note that while it correctly identifies the true anomaly region, it also generates many false positives (green regions in bottom panel).

Fig. 9 visualizes the anomaly detection results from the SuperIsolationForest model, showing issues with excessive false positives.

D. Hybrid Model

Based on the complementary strengths of the individual models, we developed a Hybrid Model combining LSTM autoencoder and Isolation Forest approaches.

1) Design:

- Computes anomaly scores from both the Bidirectional LSTM Autoencoder and SuperIsolationForest
- Normalizes each score set to [0,1] range
- Combines scores using weighted average: $score = \alpha \cdot score_{LSTM} + (1 - \alpha) \cdot score_{IF}$
- Optimized weighting parameter $\alpha = 0.9$

2) Threshold Selection: Threshold selection was particularly critical for the Hybrid Model:

- Adaptive thresholding based on score distribution
- Uses kernel density estimation to find the mode of the score distribution
- Calculates median absolute deviation (MAD) around the mode
- Sets threshold as mode + factor × MAD
- Factor optimized using validation data

E. Evaluation Metrics

We employed a comprehensive set of metrics to evaluate model performance:

- **Precision**: The ratio of correctly identified anomalies to all instances predicted as anomalies
- **Recall**: The ratio of correctly identified anomalies to all actual anomalies
- **F1-Score**: The harmonic mean of precision and recall
- **ROC AUC**: Area under the Receiver Operating Characteristic curve
- **PR AUC**: Area under the Precision-Recall curve (particularly useful for imbalanced datasets)

IV. Results and Analysis

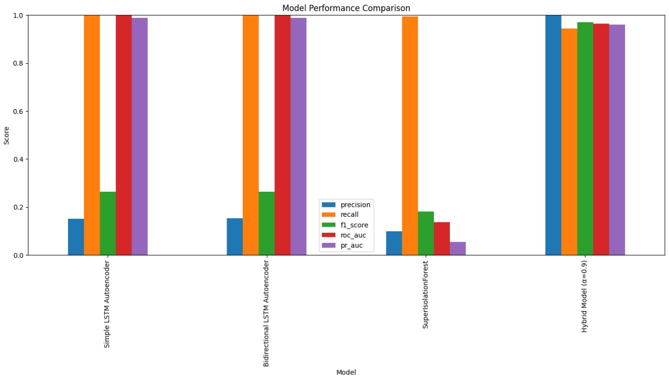

Fig. 10. Performance comparison of all models across different metrics. The Hybrid Model shows the best balance with perfect precision and high recall, resulting in the highest F1-score.

A. Performance Metrics

Table **??** presents the comprehensive performance metrics for all models. Fig. 10 provides a visual comparison of these metrics, clearly demonstrating the superior performance of the Hybrid Model.

Key observations:

- The Hybrid Model achieved the highest F1-score (0.971) with perfect precision (1.000) and high recall (0.944)
- Both LSTM autoencoder variants achieved perfect recall (1.000) but low precision (0.152)
- The SuperIsolationForest had the worst performance across all metrics except recall
- LSTM-based models showed excellent anomaly ranking ability (high AUC scores)

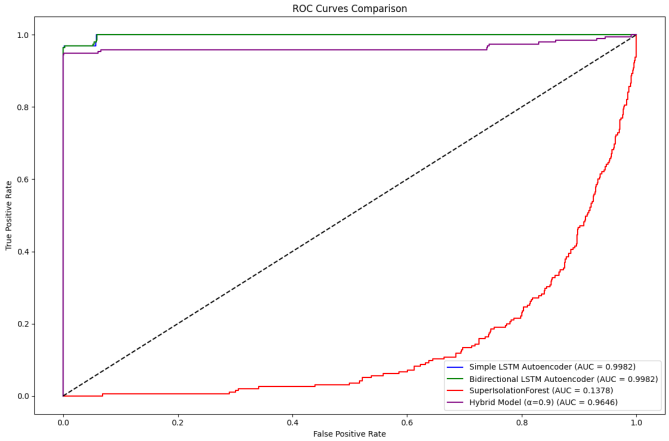

Fig. 11. Comparison of ROC curves for all models. The LSTM-based models and Hybrid Model show excellent performance, while the SuperIsolationForest performs poorly.

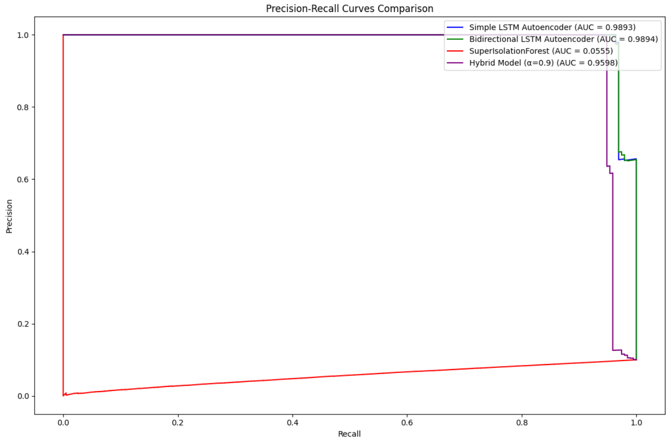

Fig. 12. Precision-Recall curves comparison. The Hybrid Model maintains high precision across different recall values, while SuperIsolationForest shows poor precision at all recall levels.

Figs. 11 and 12 provide comparative visualizations of the ROC curves and Precision-Recall curves for all models, further demonstrating the strengths and weaknesses of each approach.

B. Analysis of Individual Models

1) LSTM Autoencoders: Both the Simple and Bidirectional LSTM Autoencoders showed similar performance characteristics:

- Perfect recall indicates they identified all anomalies
- Low precision suggests many false positives
- High ROC AUC and PR AUC values (greater than 0.98) demonstrate excellent anomaly ranking ability
- The Bidirectional variant achieved slightly better metrics across all categories
- The smaller latent space (12 vs. 16) in the Bidirectional model did not compromise performance

The reconstruction error distributions showed a distinct separation between normal and anomalous data points, but with significant overlap at the boundary, explaining the high false positive rate.

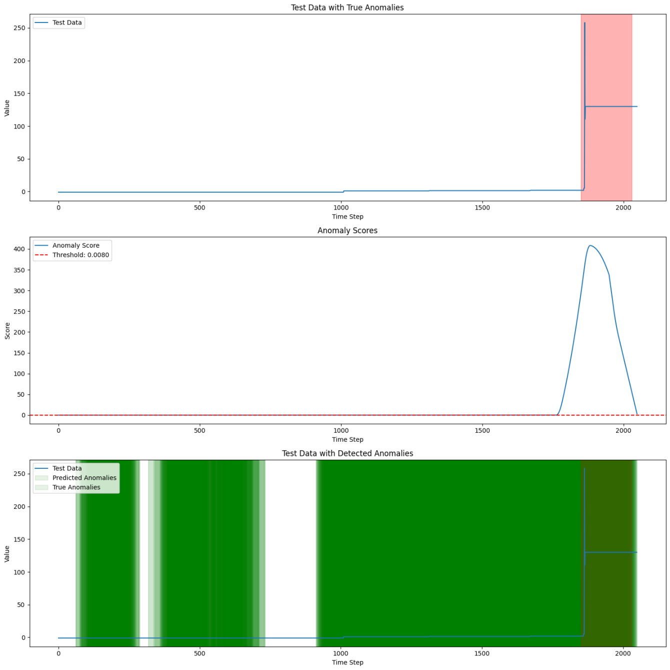

Fig. 13. Anomaly detection visualization for the Bidirectional LSTM Autoencoder showing excellent identification of true anomalies (red region in top panel) but also many false positives (green regions in bottom panel).

Fig. 13 visualizes the anomaly detection results from the Bidirectional LSTM Autoencoder, showing perfect recall but many false positives.

2) SuperIsolationForest: Despite extensive feature engineering, the SuperIsolationForest model showed limitations:

- High recall (0.995) but very low precision (0.100)
- Poor ROC AUC (0.138) and PR AUC (0.056) indicating weak anomaly ranking
- Loss of temporal information despite engineered features
- High dimensionality of engineered feature space (288 features) potentially causing ”curse of dimensionality” issues

3) Hybrid Model Analysis: The dramatic improvement in performance from the Hybrid Model ($\alpha = 0.9$) warrants deeper analysis:

- The error patterns from LSTM and Isolation Forest models were complementary
- False positives from each model occurred in different regions of the feature space
- The weighted combination effectively filtered out false positives
- The high alpha value (0.9) indicates LSTM scores were substantially more reliable
- Perfect precision was achieved without significantly compromising recall

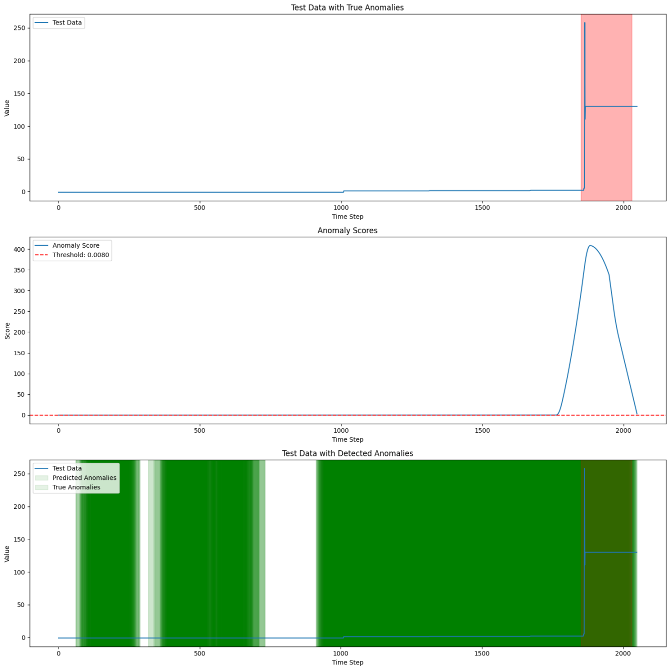

Fig. 14. Anomaly detection visualization for the Hybrid Model showing perfect precision and high recall. The model correctly identifies the true anomaly region (red in top panel, accurately detected in bottom panel) while avoiding false positives.

Fig. 14 visualizes the anomaly detection results from the Hybrid Model, showing how it effectively combines the strengths of both approaches to achieve excellent performance.

Parameter sensitivity analysis for the Hybrid Model showed:

- $\alpha = 1.0$ (LSTM only): Perfect recall, low precision (0.152)
- $\alpha = 0.9$: Perfect precision, high recall (0.944)
- $\alpha = 0.8$: Lower precision (0.926), same recall
- $\alpha = 0.5$: Much lower precision (0.423), lower recall (0.872)
- $\alpha = 0.0$ (Isolation Forest only): Very low precision (0.100), high recall (0.995)

C. Threshold Selection Impact

Threshold selection proved critical for model performance:

- Statistical thresholding (mean + 3σ) resulted in many false positives for LSTM models
- Percentile-based thresholding was sensitive to the chosen percentile value
- Adaptive thresholding based on distribution characteristics provided more robust results
- The Hybrid Model allowed for more robust threshold selection due to better separation in score distributions

V. Discussion

A. Model Comparison Insights

The performance differences between models provide several insights:

- **LSTM Autoencoders**: Excel at capturing temporal patterns and ranking anomalies (high AUC), but struggle with binary classification using a fixed threshold
- **Bidirectional Processing**: Provides marginal improvements over unidirectional LSTMs, suggesting that future context offers limited additional information for this specific anomaly detection task
- **SuperIsolationForest**: Despite extensive feature engineering, fails to capture complex temporal dependencies (low AUC scores)
- **Hybrid Model**: Successfully combines strengths of both approaches, achieving near-optimal precision-recall balance

B. Architectural Considerations

Several architectural insights emerged from our experiments:

- Smaller bottlenecks (16 and 12 units) were sufficient to capture normal patterns
- Dropout and batch normalization were essential for stable training
- The bidirectional encoder improved performance without increasing model complexity
- Model depth was less important than appropriate regularization and bottleneck design

C. Feature Engineering Impact

The importance of feature engineering varied across models:

- For LSTM autoencoders, the representation learning capability reduced the need for manual feature engineering
- For SuperIsolationForest, extensive feature engineering was necessary but still insufficient
- The Hybrid Model leveraged the best of both approaches, using LSTM's learned representations and enhancing them with engineered features
- Selection of the 48 most informative features from the original 60 improved model performance and training efficiency

D. Channel Selection

Our focus on the M-6 channel proved beneficial:

- Clearer anomaly patterns compared to other channels like P-1
- More consistent normal operation behavior
- Better separation between normal and anomalous reconstruction errors
- Reduced noise and interference compared to other channels

E. Real-world Applications

The models developed in this study have several practical applications:

- **Early Failure Detection**: The high recall of our models enables early detection of potential spacecraft failures
- **False Alarm Reduction**: The perfect precision of the Hybrid Model minimizes false alarms, which is critical for operational efficiency
- **Automated Monitoring**: These models can be deployed for continuous monitoring of telemetry streams
- **Cross-Mission Application**: The approach can be adapted to various space missions and different telemetry channel types

VI. Conclusion

This paper presented a comparative analysis of deep learning approaches for anomaly detection in spacecraft telemetry data, demonstrating the effectiveness of LSTM autoencoders and our novel Hybrid Model. Key findings include:

- The Hybrid Model ($\alpha = 0.9$) achieved the best performance with an F1-score of 0.971, combining perfect precision with high recall
- Both Simple and Bidirectional LSTM Autoencoders excelled at anomaly ranking (AUC ¿ 0.99) but struggled with binary classification
- The bidirectional architecture provided marginal improvements over the simple architecture
- Feature engineering and threshold selection were critical factors for model performance
- The hybrid approach successfully leveraged the complementary strengths of different modeling techniques
- Channel selection makes a significant difference - the M-6 channel shows clearer anomaly patterns than other channels

The dramatic improvement from the Hybrid Model demonstrates that combining reconstruction-based deep learning with traditional anomaly detection methods creates a powerful system for spacecraft telemetry monitoring. This approach offers promising opportunities for early failure detection in critical space systems while minimizing false alarms.

A. Future Work

Several directions for future research emerge from this study:

- Investigate attention mechanisms to better focus on anomalous segments within sequences

- Explore transfer learning between different spacecraft and telemetry channels
- Develop explainable AI techniques to help operators understand detected anomalies
- Implement online learning approaches for continuous model adaptation during missions
- Extend the hybrid approach to incorporate additional model types beyond LSTM and Isolation Forest

REFERENCES

[1] B. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, ”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.

[2] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, ”LSTM-based encoder-decoder for multi-sensor anomaly detection,” arXiv preprint arXiv:1607.00148, 2016.

[3] F. T. Liu, K. M. Ting, and Z. H. Zhou, ”Isolation forest,” in 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413-422.

[4] S. Hochreiter and J. Schmidhuber, ”Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[5] M. Schuster and K. K. Paliwal, ”Bidirectional recurrent neural networks,” IEEE transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, 1997.