TECHNICAL TASK EXPLANATION

1. Introduction:

The aim of this task was to analyze helicopter accelerometer data to detect abnormal behaviour by learning from normal data sequences. This involved preprocessing accelerometer data, structuring it for time-series analysis, and developing a machine learning model that distinguishes normal from potentially abnormal sequences. This detection system could enhance the early identification of irregularities, contributing to preventive maintenance and operational safety.

2. Data Analysis and Preprocessing:

Initially, I explored dataset_1 (healthy data) and dataset_2 (anomalies) in HDF5 format, containing one-minute sequences at 1024Hz, each recorded with various angles and locations on test helicopters. Each sequence has 61,440 time points (1024 Hz * 60 seconds). Since absolute accelerometer values were scaled by a factor, I avoided further normalization to preserve data distribution patterns.

Key Analysis steps included:

- Labeling: dataset_1 (normal) labeled as 0, and dataset_2 (abnormal) as 1.
- Combining Datasets: Merging both datasets into a single dataframe for unified analysis and later model training.
- Shuffling: Dataframe is shuffled to have random rows of “normal” and “abnormal” samples.
- Descriptive Statistics: Calculate mean, median, standard deviation, minimum, and maximum for each axis (X, Y, Z) across sequences. This provides insights into the typical range of values and variability.
- Distribution Analysis: Plotted histograms for each axis and visualized the data distribution. This can reveal any skewness or irregularities in data distributions, which might indicate biases or patterns. The features are normally distributed, as shown in Figure 1.
- Autocorrelation: As in Plot Figure 2, it shows the autocorrelation of five samples over various lags. Each sequence starts with high self-correlation and then

exhibits different decay and oscillation patterns, indicating varying periodicities. This can help in identifying periodic anomalies in the data.

- Extracted the labels and features separately for further analysis.
- Checking NaN values: The features doesn’t have any NaN or Null values.
- Label analysis: The dataset has class imbalance by having 1677 “normal” (0) and 594 “abnormal” (1) labels.

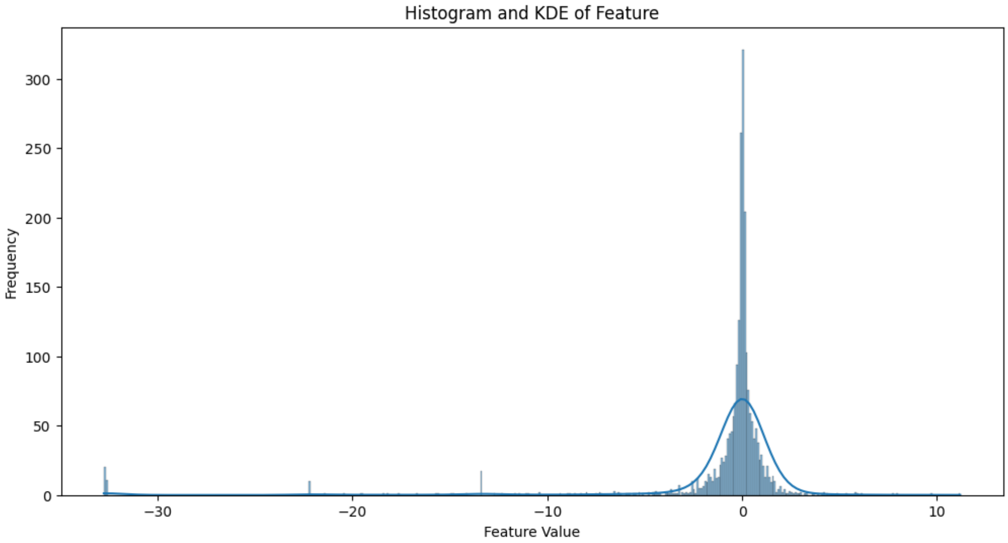

Figure 1: Histogram plot for the features

Key preprocessing steps included:

- Data Splitting: The dataset is split into three sets such as training set, validation set, and testing set before further preprocessing steps to avoid any data leakage.
- Class Imbalance: To address the class imbalance in the dataset, a sampling technique like SMOTE is implemented on the label for class balance. However, it is performed only on the training set to keep the validation set imbalanced, which resembles the real-world environment.
- Normalization/Standardization: The resampled training set is fit with the StandardScalar() to standardize (normalize) the values in the range [0.1]. Likewise, for the validation and testing set.

These steps ensured the data was appropriately labeled and formatted, which is essential for effective model training.

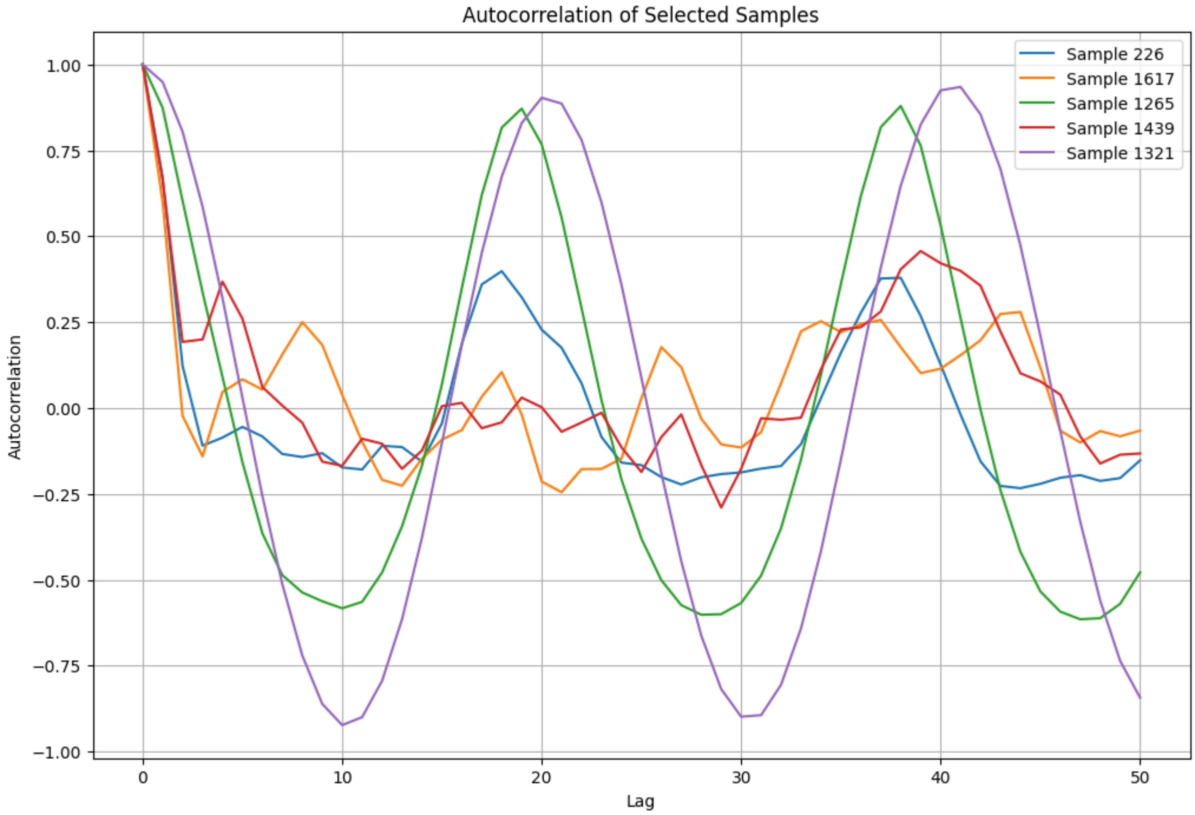

Figure 2: Autocorrelation of Samples

3. Feature Engineering:

For feature engineering, I applied the following techniques:

- Segmented Time-Series with Sliding Windows: To enhance granularity, each sequence was segmented into smaller overlapping windows, improving the model's ability to detect finer temporal abnormalities.
- Fourier Transform (FFT): I replaced raw time-domain data with frequency-domain representations. This helps the model identify periodic patterns and oscillations that could signify abnormal behaviors, leveraging both time and frequency domains.

Time-series data often has frequency components that vary over time, and the sliding window FFT approach captures this dynamic information. By applying a Hann window to each segment, spectral leakage is reduced, making the FFT results more accurate for signals where frequencies change over time. This method is particularly helpful for anomaly detection in time-series data, where abnormal events may introduce unique frequency patterns. By converting time-domain signals to frequency-domain data, it becomes easier to apply machine learning models to

These steps are useful for analyzing the frequency components of time-series data across multiple segments. Using a sliding window approach allows capturing how the frequency content of the signal changes over time.

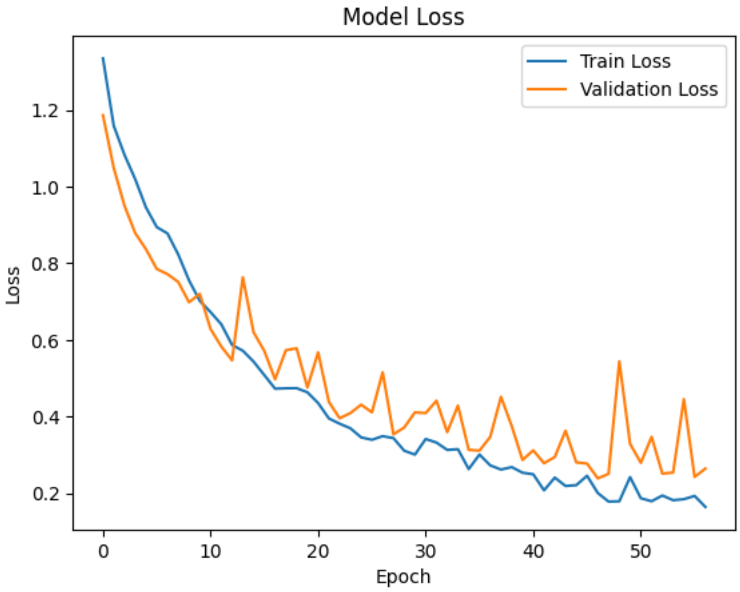

Figure 3: Loss curve on the Training set and Validation set

5. Model Evaluation:

For model evaluation, I used:

- **F1-Score and Accuracy**: To assess performance in detecting abnormal sequences accurately.
- **Loss Curve**: Visualized the learning loss curve on the training and validation set as displayed in Figure 3. This figure represents that the model is not overfitting, as the validation loss curve did not diverge.
- **Confusion Matrix**: Visualized the model's ability to differentiate between normal and abnormal sequences as shown in Figure 4. The result shows that there are 30 misclassifications.

These metrics ensured that the model was not only accurate but also robust in detecting subtle anomalies, and minimizing false negatives (missed abnormal sequences).

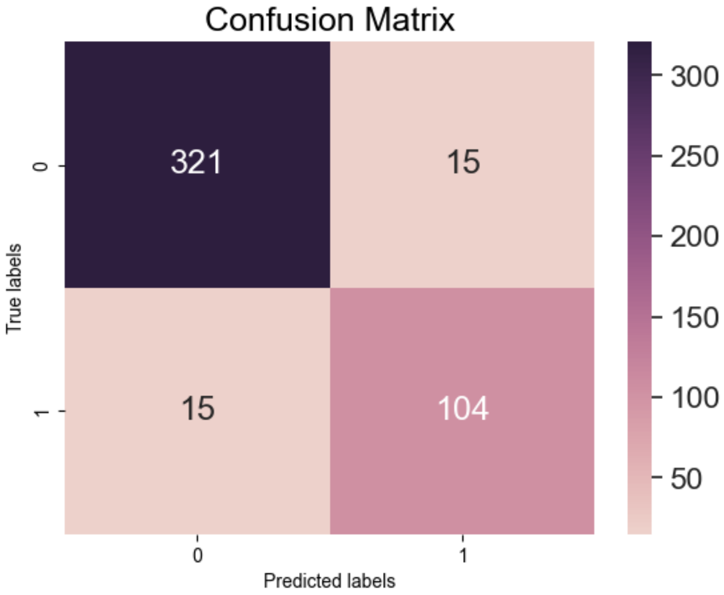

Figure 4: Confusion Matrix on Test set

6. Conclusion:

The results indicate that the CNN-LSTM model, coupled with FFT-transformed data, effectively distinguishes normal and abnormal accelerometer patterns with an F1 score of 0.87 and an accuracy of 0.94. Segmenting time series with a sliding window significantly enhanced detection performance.

Potential improvements include experimenting with more complex architectures or fine-tuning the sliding window size to capture varying levels of granularity. Moreover, exploring different sampling techniques or Generative models to address the class imbalance. Future steps can also involve exploring ensemble methods and other time-series algorithms like Transformer models.