

**MANIPAL SCHOOL OF INFORMATION SCIENCES**

**(A Constituent unit of MAHE, Manipal)**

**Anomaly detection in Time series data using Transformers**

|  |  |  |
| --- | --- | --- |
| **Reg. Number** | **Name** | **Branch** |
| **241057022** | **Samarth Prathap M** | **AI-ML** |
| **241057015** | **Sanjana Y Alva** | **AI-ML** |

## **Under the guidance of**

## **Mr. Sudarsan N S Acharya**

**Assistant Professor**

**Manipal School of Information Sciences,**

**MAHE, MANIPAL**

### 15/11/2024

****

# Abstract

Anomaly detection in time series data is critical for ensuring the efficiency, safety, and reliability of industrial systems, particularly in sectors such as oil and gas, where real-time sensor data is continuously monitored. Traditional methods for anomaly detection often struggle to handle the complexity and volume of high-dimensional sensor data. This project explores the use of a Transformer-based Autoencoder for unsupervised anomaly detection in time series data. By combining the strengths of Transformer models—specifically their self-attention mechanism for capturing long-range dependencies—and Autoencoders—used for learning efficient data representations and detecting deviations based on reconstruction errors—the model can effectively identify both sudden faults and gradual shifts in the data. The Transformer’s ability to process large, sequential data without losing contextual information, coupled with the Autoencoder’s reconstruction error-based anomaly detection, makes this approach highly suitable for industrial applications where labelled data is scarce. The model was trained on sensor data from a hydraulic test rig, and its performance was evaluated through reconstruction error metrics, successfully detecting anomalies and providing a powerful tool for preventive maintenance and fault detection. This work demonstrates the potential of Transformer-based Autoencoders for enhancing real-time monitoring and anomaly detection in complex industrial environments.

# Acknowledgement

We would like to express our heartfelt gratitude to all those who have supported and contributed to the successful completion of this project.

First and foremost, we would like to extend our deepest thanks to **Mr. Sudarsan N S Acharya**, Assistant Professor at Manipal School of Information Sciences, MAHE, Manipal, for his exceptional guidance, unwavering support, and constant encouragement throughout the course of this project. His invaluable insights, constructive feedback, and profound expertise have been instrumental in shaping the direction of this work. We are truly fortunate to have had the opportunity to work under his mentorship.

We would also like to express our sincere appreciation to **Manipal School of Information Sciences** for providing the necessary resources and tools, which were crucial to the successful completion of this project. Their support ensured that we had everything needed to carry out this research effectively.

Finally, we would like to convey our deepest gratitude to our families and friends for their unfailing love, patience, and encouragement. Their belief in us, even during challenging moments, motivated us to continue striving for excellence and to complete this project with dedication and perseverance.

Samarth Prathap M

Sanjana Y Alva

# Table of Contents

[**Abstract 1**](#_Toc182509969)

[**Acknowledgement 2**](#_Toc182509970)

[**Table of Contents 3**](#_Toc182509971)

[**List of Figures 4**](#_Toc182509972)

[**Chapter 1: Introduction 5**](#_Toc182509973)

[**Chapter 2: Objectives 7**](#_Toc182509974)

[**Chapter 3: Literature Survey 8**](#_Toc182509975)

[**Chapter 4: Specifications 9**](#_Toc182509976)

[**Chapter 5: Methodology 11**](#_Toc182509977)

[**Chapter 6: Work Done 13**](#_Toc182509978)

[**Chapter 7: Results 18**](#_Toc182509979)

[**Chapter 8: Conclusion 19**](#_Toc182509980)

[**Chapter 9: Future Work 20**](#_Toc182509981)

[**Chapter 10: References 21**](#_Toc182509982)

# List of Figures

[Figure 5.1: Flowchart to Implement Transformer-based Autoencoder for anomaly detection 11](#_Toc182509868)

[Figure 6.1: Transformer-based Autoencoder Architecture 13](#_Toc182509869)

[Figure 6.2: Positional Encoding 14](#_Toc182509870)

[Figure 6.3: Encoder and Decoder Architecture 15](#_Toc182509871)

[Figure 6.4: Flowchart for Anomaly Detection 17](#_Toc182509872)

[Figure 7.1: Model Performance Metrics for Different Classifiers 18](#_Toc182509873)

[Figure 7.2: Training Mean Squared Error 18](#_Toc182509874)

# Chapter 1: Introduction

In today’s industrial world, especially in important sectors like oil and gas, it is very important to continuously monitor machinery and processes to ensure efficiency, safety, and reliability. In various industries, particularly on oil rigs, sensor networks are being widely used to gather real-time data on key factors like temperature, pressure, and flow rates. However, as the amount of data increases, analysing it becomes more difficult. Detecting abnormalities in these data streams is crucial because they can indicate problems like equipment failures, safety risks, or inefficiencies.

Traditionally, detecting anomalies has depended a lot on labelled datasets. But in many real-world situations, getting such labelled data can be tough and costly. Moreover, traditional methods often find it hard to handle the complex and high-dimensional nature of time series data, especially the kind produced by industrial sensors on oil rigs. This project aims to explore how Transformer models, which are known for their success in natural language processing, can be used to detect anomalies in time series data.

Transformers are well-suited for this task because they can understand long-term patterns and relationships in sequences. The goal of this project is to develop a strong anomaly detection system that does not rely on labelled data, using unsupervised learning methods. This system will aim to spot even the smallest and most complex irregularities in sensor data, making it a crucial tool for preventive maintenance, fault detection, and improving operational efficiency.

This project intends to show how Transformer models can be useful in a new field, providing insights into how they can be adjusted for non-labelled time series data, and paving the way for their practical use in industrial settings, both in India and beyond, particularly in the oil and gas industry.

The Transformer model, introduced in the 2017 paper "Attention is All You Need," uses a self-attention mechanism to capture long-range dependencies in sequential data. Unlike traditional models like RNNs or LSTMs, Transformers allow for parallel processing of sequences, making them more computationally efficient for large datasets.

Key features include:

* Self-Attention: Captures both short- and long-term dependencies in the data.
* Position Encoding: Adds temporal information to input sequences, ensuring the model understands the order of data points.
* Scalability: Suitable for processing long time series data without losing performance.

In this project, the Transformer helps analyse sensor data from industrial equipment, learning normal patterns and detecting deviations.

An Autoencoder is a type of neural network used for unsupervised learning, consisting of an encoder that compresses input data into a lower-dimensional space and a decoder that attempts to reconstruct the original data. The key idea is that the model will struggle to reconstruct anomalous data, resulting in high reconstruction errors, which can be flagged as anomalies.

For anomaly detection:

* Normal Data: Low reconstruction error.
* Anomalous Data: High reconstruction error, indicating a deviation from normal patterns.

Autoencoders are ideal for time series anomaly detection because they learn the typical patterns in the data and can detect deviations without needing labelled data.

Combining the strengths of both models, the Transformer-based Autoencoder is well-suited for time series anomaly detection for the following reasons:

1. Capturing Temporal Patterns: The Transformer’s attention mechanism can capture long-range dependencies in time series data, which is critical for detecting anomalies that may span multiple time steps.
2. Efficient Anomaly Detection: The Autoencoder reconstructs the input data, and anomalies are flagged based on high reconstruction errors, allowing for effective anomaly detection without labelled data.
3. Scalability: This combination can handle large, high-dimensional sensor data while efficiently identifying subtle anomalies in real-time.

In this project, the Transformer-based Autoencoder is used to analyse sensor data from industrial equipment, detecting anomalies such as equipment malfunctions, safety risks, or inefficiencies.

# Chapter 2: Objectives

1. **Develop an Advanced Anomaly Detection System**:
   * Leverage transformers to accurately detect anomalies in multivariate time series data from hydraulic systems.
2. **Capture Complex Dependencies**:
   * Use transformer models to effectively learn temporal and feature dependencies, providing a robust approach to anomaly detection in industrial settings.
3. **Enhance Predictive Maintenance**:
   * Identify early signs of equipment failure to minimize downtime and maintenance costs.

# Chapter 3: Literature Survey

**Z. Chen et al., "Autoencoder-based Network Anomaly Detection" (2018)**

This paper explores the use of autoencoders for network anomaly detection, demonstrating how reconstruction error can identify anomalies in high-dimensional data, such as network traffic. The study shows that autoencoders, particularly convolutional autoencoders, perform well in detecting non-linear anomalies compared to traditional methods like Principal Component Analysis (PCA). By using dimensionality reduction and feature extraction, the model simplifies complex data while maintaining essential patterns for anomaly detection. This approach, tested on the NSL-KDD dataset, inspired the use of autoencoders in our project to detect anomalies in time series data through reconstruction error, benefiting from the autoencoder's ability to model complex patterns in sensor data.

**D. Liu and G. Liu, "A Transformer-Based Variational Autoencoder for Sentence Generation" (2019)**

Liu and Liu’s paper introduces a Transformer-based Variational Autoencoder (VAE) for sentence generation, combining self-attention mechanisms with a probabilistic encoding-decoding process. The self-attention mechanism enables the model to capture long-range dependencies within sequences, making it particularly effective for sequential data tasks like text generation. This approach inspired the incorporation of Transformers into our anomaly detection model, as their ability to model complex temporal dependencies in time series data is crucial for identifying both sudden and gradual anomalies. The paper’s combination of Transformers and autoencoders provides a strong foundation for applying similar techniques to time series anomaly detection.

Thus, the combination of Transformers and Autoencoders in this project is driven by their complementary strengths. Transformers are well-suited for capturing long-range dependencies and efficiently processing sequential data, making them ideal for identifying complex temporal patterns in time series. Autoencoders, on the other hand, are effective at learning compact representations of data and detecting anomalies through reconstruction error. By combining these models, we can leverage the Transformer’s ability to capture intricate temporal relationships and the Autoencoder’s capacity to identify deviations from normal patterns, resulting in a powerful approach for unsupervised anomaly detection in time series data.

# Chapter 4: Specifications

The project focuses on unsupervised anomaly detection in multivariate time-series data using a Transformer-based Autoencoder. Below are the key specifications:

**1. Input Data:**

* **Data Source:** Time-series data from industrial sensors, including pressure, temperature, flow rate, and vibration sensors, gathered from a hydraulic test rig.
* **Attributes:**
  + **Pressure Sensors:** PS1 to PS6 (100 Hz)
  + **Flow Sensors:** FS1, FS2 (10 Hz)
  + **Temperature Sensors:** TS1 to TS4 (1 Hz)
  + **Vibration Sensors:** VS1 (1 Hz)
  + **Motor Power (EPS1):** 100 Hz
  + **Cooling Efficiency (CE):** 1 Hz
  + **Cooling Power (CP):** 1 Hz
  + **Efficiency Factor (SE):** 1 Hz
* **Data Format:** The raw process sensor data is structured as matrices (tab-delimited), where rows represent cycles and columns represent the data points within a cycle. The data set contains 2205 instances and 43,680 attributes, with time series data at various sampling rates (1 Hz, 10 Hz, and 100 Hz).
* **Target Values:** The target condition values for each cycle, including cooler, valve, pump, and accumulator conditions, are annotated in a separate file ('profile.txt'). This includes:
  + **Cooler condition:** 3% (close to total failure) to 100% (full efficiency)
  + **Valve condition:** 100% (optimal) to 73% (close to total failure)
  + **Pump leakage:** 0 (no leakage) to 2 (severe leakage)
  + **Accumulator pressure:** 90 bar (close to total failure) to 130 bar (optimal pressure)
  + **Stable flag:** 0 (stable) to 1 (static conditions not reached)

**2. Model Architecture:**

* **Transformer-based Autoencoder:** The model comprises:
  + **Encoder:** Includes 1D convolutional layers, multi-head attention, and global average pooling to capture both local and global patterns in the time series data.
  + **Decoder:** Utilizes an LSTM layer followed by time-distributed dense layers for reconstructing the time series input.
  + **Positional Encoding:** Applied to the input data to preserve the temporal information necessary for detecting anomalies in sequential data.

**3. Training:**

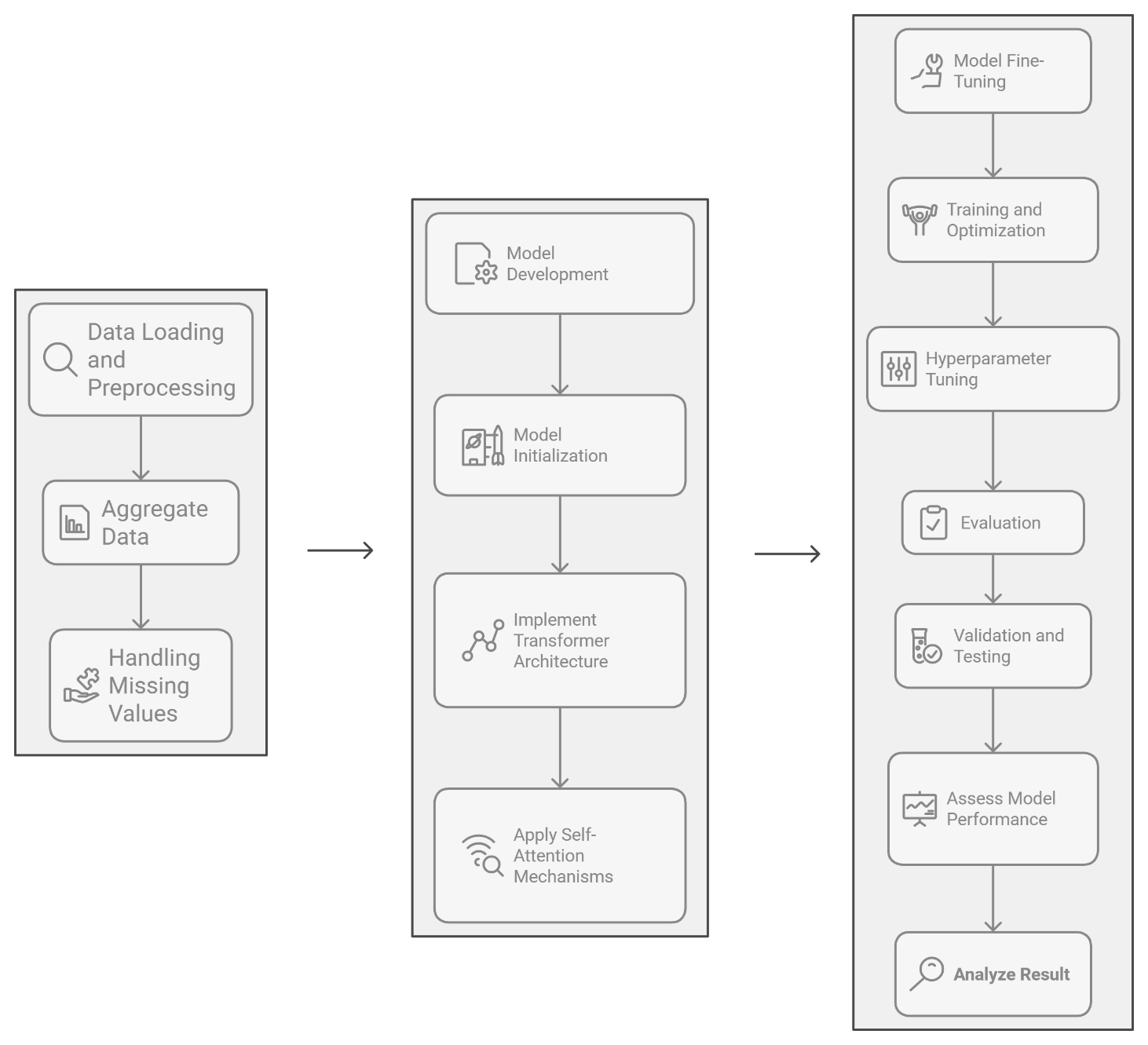
* **Unsupervised Learning:** The model is trained unsupervised, learning normal patterns from the input data without requiring labeled instances for anomalies.
* **Batch Processing:** Implemented to handle large datasets efficiently.
* **Optimizer:** Adam optimizer, with Mean Squared Error (MSE) used as the loss function to minimize reconstruction error during training.

**4. Evaluation:**

* **Reconstruction Error (MSE):** Reconstruction error is used to identify anomalies. A threshold for the error is set, and anomalies are flagged when the error exceeds this threshold.

# Chapter 5: Methodology

The project follows a structured workflow to implement the Transformer-based Autoencoder for anomaly detection.



## Figure 5.: Flowchart to Implement Transformer-based Autoencoder for anomaly detection

The main steps are as follows:

1. **Data Collection and Preprocessing**:
   * Collect raw sensor data, aggregate 5-minute interval data to a 30-minute time frame, and perform any necessary data cleaning.
   * Handle missing values using interpolation to ensure the data is continuous and complete.
2. **Model Architecture Design**:
   * Define the Transformer-based Autoencoder architecture with attention mechanisms, convolutional layers, and LSTM for reconstruction.
   * Implement positional encoding to provide temporal information to the model.
3. **Model Training**:
   * Train the model using unsupervised learning, where the model learns the normal patterns of the time-series data.
   * Use batch processing to efficiently handle large datasets and prevent memory overload.
4. **Anomaly Detection**:

* After training, use the model to reconstruct the test data and calculate the reconstruction error.
* Identify anomalies based on the threshold of reconstruction error.

1. **Evaluation**:

* Evaluate the model by calculating the reconstruction error for both normal and anomalous test data.
* Analyse the model's performance in detecting anomalies based on error distribution and visual inspection.

# Chapter 6: Work Done

**6.1. Data Preparation**

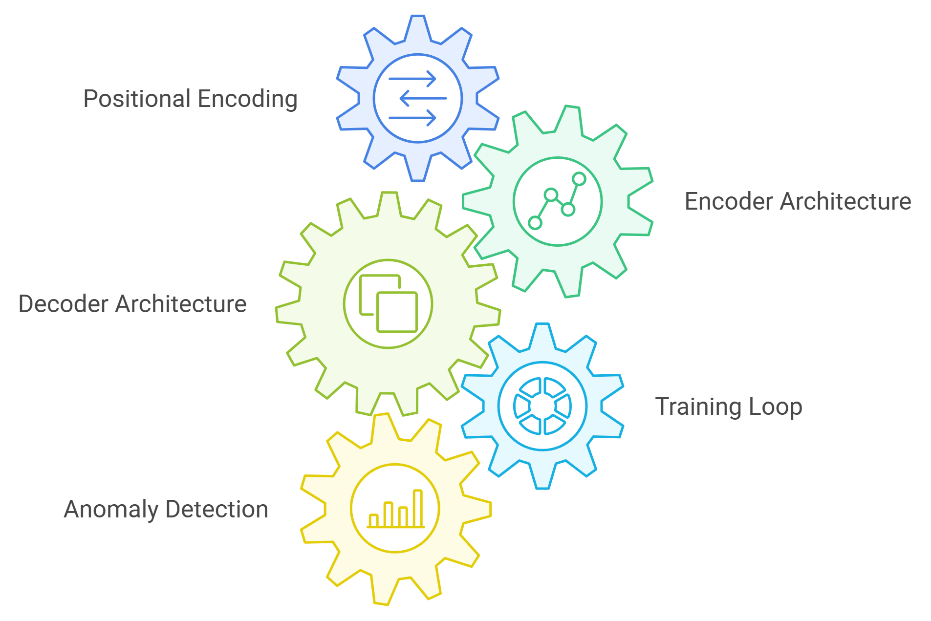
The first step in the project was to process and prepare the raw time series data collected from sensors. The key tasks involved in data preparation were:

* **Aggregation of Data**: The raw sensor data, which was originally collected at 5-minute intervals, was aggregated into 30-minute intervals to reduce the granularity and improve the model's ability to detect trends and patterns over longer periods.
* **Handling Missing Data**: Due to the presence of missing data points in the sensor readings, linear interpolation methods were used to fill in missing values. This ensured that the dataset remained complete, which is important for training an effective model.

**6.2. Model Development**

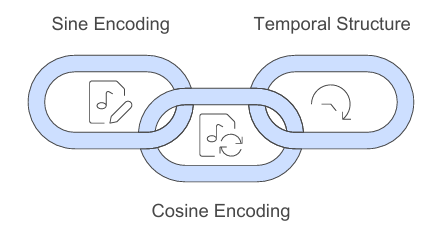
The next step was to implement the Transformer-based Autoencoder architecture, which was designed to learn patterns in the time series data and detect anomalies. The tasks performed during model development included:

* **Transformer-based Autoencoder Architecture**:
  + The model was implemented using TensorFlow and Keras. This architecture includes both an encoder and a decoder, with the encoder learning to extract features and the decoder reconstructing the original input sequence.



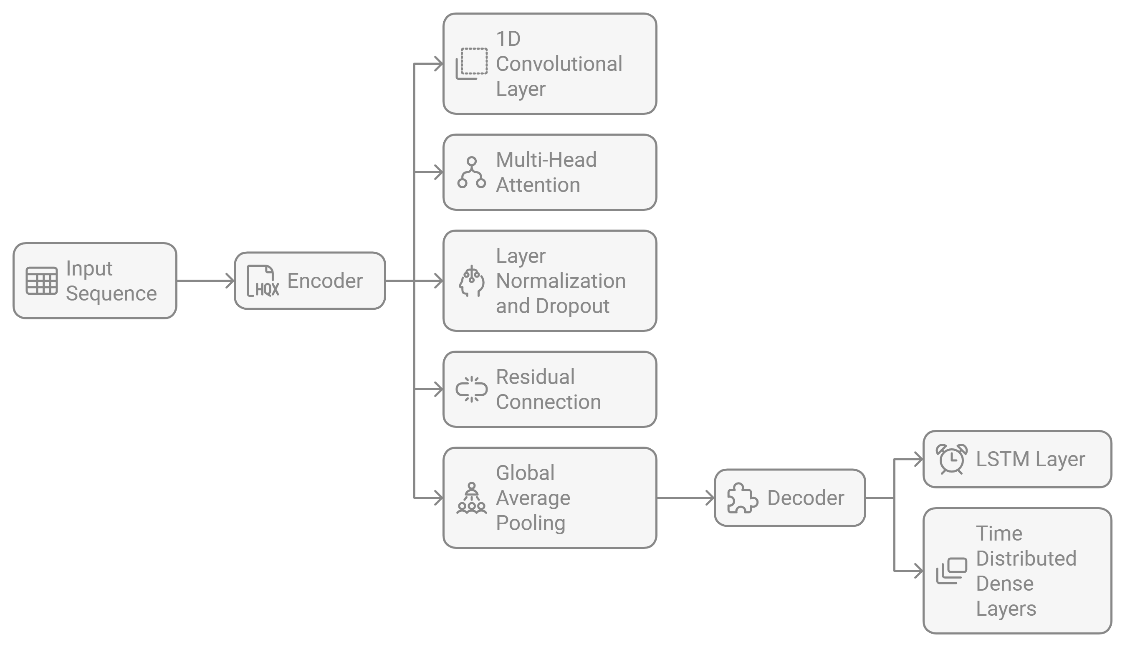
## Figure 6.1: Transformer-based Autoencoder Architecture

* **Positional Encoding**:
  + A critical component of the model is positional encoding, which is used to inject time-dependent information into the input data. This allows the Transformer model to understand the temporal relationships between data points in the sequence. The encoding was based on a sine-cosine function, allowing the model to capture temporal dependencies in the time-series data.



## Figure 6.: Positional Encoding

* **Encoder (Understanding Sequence Data):** The encoder is responsible for learning relationships within the input sequence and transforming it into a feature representation. The encoder architecture includes the following components:
* **1D Convolutional Layer:** This layer is applied to the input time series data to capture local patterns and dependencies within the sequence. It helps the model identify important features at different time steps.
* **Multi-Head Attention:** Multi-head attention enables the model to capture relationships between different parts of the input sequence simultaneously. It helps the model focus on various segments of the data and learn the global dependencies in the time series.
* **Layer Normalization and Dropout:** Layer normalization is applied to stabilize and normalize activations across the batch, helping to improve training speed and stability. Dropout is used as a regularization technique to reduce overfitting during training.
* **Residual Connection:** The residual connection ensures that important information is preserved throughout the encoding process. It helps avoid issues like vanishing gradients and allows the model to retain more information from the original input.
* **Global Average Pooling:** This layer aggregates the features extracted by the encoder, reducing the dimensionality of the data and preparing it for the decoder.



## Figure 6.: Encoder and Decoder Architecture

* **Decoder:** Reconstructing the Sequence The decoder is responsible for reconstructing the original time series data from the encoded features. It aims to minimize the reconstruction error, which is critical for anomaly detection. The decoder's structure includes the following:
* **LSTM Layer:** The output from the encoder is passed through an LSTM layer. The LSTM layer helps model long-term dependencies and reconstruct the time series sequence by processing sequential data.
* **Time Distributed Dense Layers:** These layers are applied independently to each time step in the sequence to produce the final output, which is the reconstructed version of the original time series input.

**6.3. Training and Optimization**

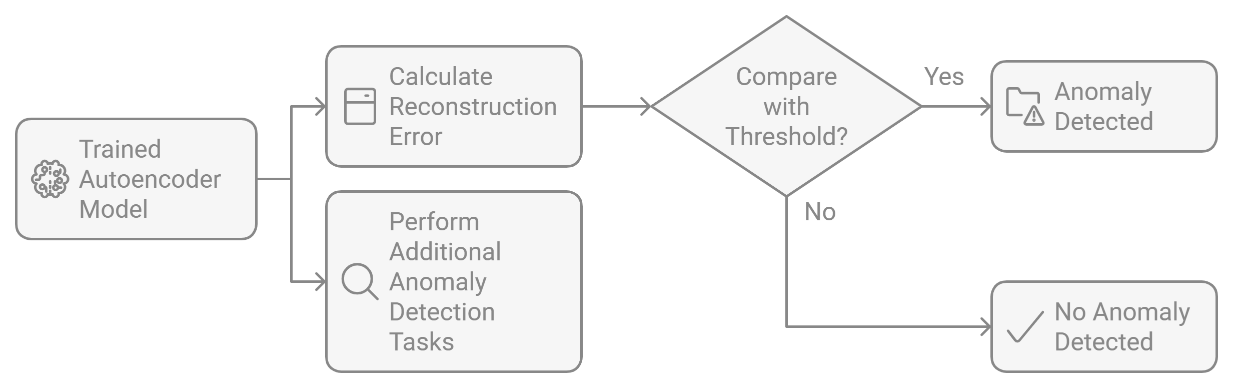
Once the model architecture was developed, the next task was to train and optimize the model. The following steps were undertaken to ensure efficient training and model optimization:

* **Efficient Memory Usage with Batch Generator**:
  + A batch generator was created to handle large datasets and ensure memory-efficient training. This function dynamically yields smaller batches of data during training, which prevents memory overloads when working with large time-series datasets.
* **Model Training**:
  + The model was trained for a total of 10 epochs. During training, the model learned to reconstruct time series sequences from the encoder's feature representations. The reconstruction error was used as the metric to evaluate the model's performance in identifying anomalies.
* **Optimizer and Hyperparameter Tuning**:
  + The Adam optimizer was used for training the model due to its efficiency in optimizing deep learning models. To further improve model performance, grid search was employed for hyperparameter tuning. This helped in finding the optimal settings for the model, such as learning rate and other relevant parameters.

**6.4. Anomaly Detection**

Once the model was trained, it was tested on unseen data to detect anomalies in time series sequences. The following tasks were conducted for anomaly detection:

* **Reconstruction Error Calculation**:
  + The trained model was used to compute the reconstruction error for each data point in the test dataset. The reconstruction error represents the difference between the original data and the reconstructed data. Normal data points are expected to have low reconstruction errors, while anomalies will have higher reconstruction errors.
* **Anomaly Identification**:
  + Anomalies were identified by setting a **threshold** for the reconstruction error. If the reconstruction error exceeded this threshold, the data point was flagged as an anomaly. This step is critical in differentiating normal behaviour from potentially faulty or anomalous events in the data.



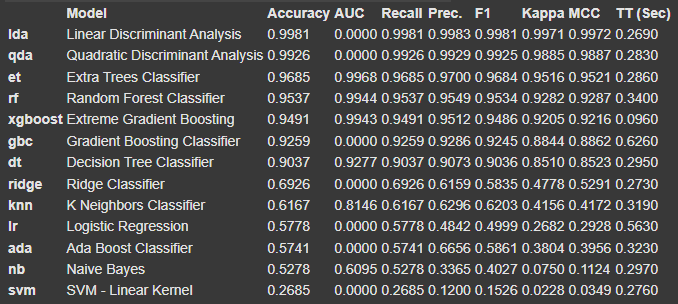
## Figure 6.: Flowchart for Anomaly Detection

**6.5. Evaluation**

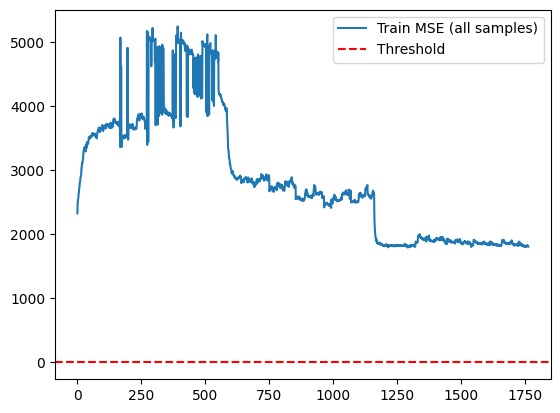
After the model successfully detected anomalies, it was evaluated to assess its effectiveness in identifying both sudden faults and gradual shifts in the data. The following steps were taken for evaluation:

* **Performance Evaluation**:
  + The model’s performance was evaluated based on its ability to detect anomalies accurately. The reconstruction error was used as the primary metric for evaluation, with higher errors indicating anomalies.
* **Visual Inspection of Anomalies**:
  + To verify the model’s anomaly detection capabilities, a visual inspection of the detected anomalies was carried out. This involved comparing the anomalies detected by the model with actual fault conditions in the data (where available). The visual inspection confirmed that the model was able to detect both sudden faults (e.g., equipment failures) and gradual shifts (e.g., sensor drift or gradual deterioration) effectively.

# Chapter 7: Results



## Figure 7.1: Model Performance Metrics for Different Classifiers



## Figure 7.2: Training Mean Squared Error

# Chapter 8: Conclusion

The Transformer-based Autoencoder model developed during this project has shown promising results in detecting anomalies in time series data. The use of positional encoding for capturing temporal dependencies, combined with a robust training and evaluation pipeline, enabled the model to effectively identify irregularities in sensor data without the need for labelled data. The model's ability to detect both sudden and gradual anomalies in real-time sensor data makes it an ideal candidate for applications in industries like oil and gas, where continuous monitoring of equipment and systems is crucial for safety and efficiency.

# Chapter 9: Future Work

There are several areas for improvement and extension of the current work:

1. **Data Augmentation**:

* Incorporating synthetic anomaly data through data augmentation techniques could help improve the model’s robustness, especially for rare anomalies.

1. **Model Enhancements**:

* Experimenting with different model architectures, such as adding more layers to the encoder-decoder structure or using Bidirectional LSTM layers in the decoder, could improve the model’s performance.

1. **Real-time Anomaly Detection**:

* Deploying the trained model in a real-time environment to monitor ongoing sensor data from industrial systems, with continuous updates to model predictions.

1. **Evaluation with Other Datasets**:

* Testing the model on other types of time-series data from different industrial applications could demonstrate its versatility.

# Chapter 10: References

[1] Z. Chen, C. K. Yeo, B. S. Lee and C. T. Lau, "Autoencoder-based network anomaly detection," 2018 Wireless Telecommunications Symposium (WTS), Phoenix, AZ, USA, 2018, pp. 1-5, doi: 10.1109/WTS.2018.8363930. keywords: {Anomaly detection;Dimensionality reduction;Principal component analysis;Training;Deconvolution;Two dimensional displays;Correlation;Network Anomaly Detection;Autoencoder;Convolutional Autoencoder;Dimensionality Reduction;Reconstruction Error;NSL-KDD Dataset},

[2] D. Liu and G. Liu, "A Transformer-Based Variational Autoencoder for Sentence Generation," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-7, doi: 10.1109/IJCNN.2019.8852155. keywords: {Decoding;Training;Task analysis;Neural networks;Gaussian distribution;Computer architecture;Natural languages;variational autoencoder;text generation;self-attention;transformer},