A Project Report

On

ANOMALY DETECTION IN
WBANS (WIRELESS BODY AREA
NETWORK)

BY

HARDIK AGRAWAL 2022A7PS0045H

Under the supervision of

Prof. Chittaranjan Hota

SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS OF CS F366:

LAB PROJECT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI

HYDERABAD CAMPUS

(APRIL 2025)

Birla Institute of Technology and Science-Pilani,

Hyderabad Campus

Certificate

This is to certify that the project report entitled

"ANOMALY DETECTION IN WBANS (WIRELESS BODY AREA NETWORK)"

submitted by Mr. HARDIK AGRAWAL(ID No. 2022A7PS0045H) in fulfillment of the requirements of the courses CS F366, Study Project Course, embodies the work done by him under my supervision and guidance.

Date: 24/4/2025 **(Chittaranjan Hota)**

ABSTRACT

Wireless Body Area Networks (WBANs) are wireless networks that consist of microscopic sensors that are either placed on a subject's body or attached to their clothes. These low-power devices track the physiological readings from a subject and relay them to a server over the Internet. Since WBANs find various uses, from remote medical monitoring to early disease detection in the medical field, their readings must be accurate. Anomalies can occur in the data being transmitted for various reasons, such as sensor faults or malicious attacks.

Also It is important to understand that all data points flagged as anomaly might not be anomalous point as in some cases context of what activities subject is performing during taking data readings might satisfy the abnormal behavior of certain physiological parameters. To mitigate such cases, we must classify anomaly as Point and Contextual anomaly. The anomaly detection developed here consists of using reconstruction error from CNN Autoencoders to flag Point and Contextual Anomaly which then passed to LSTM+SELF ATTENTION Architecture to distinguish between

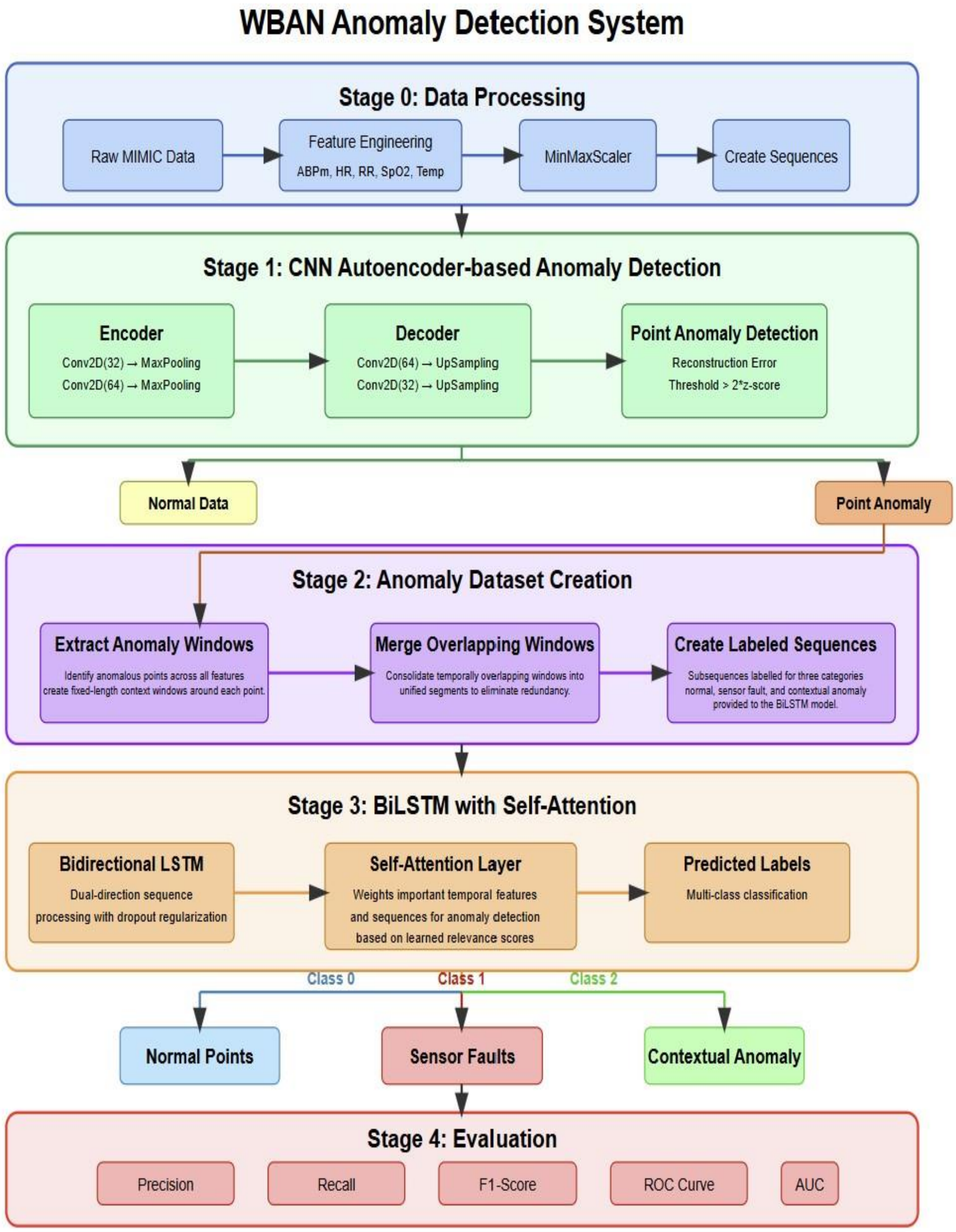
Sensor faults and contextual anomalous points.

Proposed Model

The proposed anomaly detection process consists of 2 steps: first, the CNN autoencoder reconstructs the input data; if, for any point, the error is greater than a set threshold, then the point is labelled as point anomaly. Then, the encoded data is passed to the LSTM classifier, classifying it as being contextually anomalous or not. Thus, we have proposed a combination of CNN and LSTM models to identify contextual anomalies present in the sensor values.

CNN Autoencoders focus on reconstructing the same data and back propagating error as difference between encoded and real data. Also, for better detection of anomalies we trained CNN Autoencoders on nonanomalous data and testing on data with anomalies for better results.

LSTM Architecture becomes a best fit for capturing temporal dependencies of features based on long term dependencies of feature values , we could also add Self Attention Layers to decide what attention does other correlated features pays to each other and capturing temporal dependencies by RNN network.


WBAN Anomaly Detection System
Stage 0: Data Processing
Raw MIMIC Data
Feature Engineering
ABPm, HR, RR, SpO2, Temp
MinMaxScaler
Create Sequences
Stage 1: CNN Autoencoder-based Anomaly Detection
Encoder
Conv2D(32) → MaxPooling
Conv2D(64) → MaxPooling
Decoder
Conv2D(64) → UpSampling
Conv2D(32) → UpSampling
Point Anomaly Detection
Reconstruction Error
Threshold > 2*z-score
Normal Data
Point Anomaly
Stage 2: Anomaly Dataset Creation
Extract Anomaly Windows
Identify anomalous points across all features
create fixed-length context windows around each point.
Merge Overlapping Windows
Consolidate temporally overlapping windows into
unified segments to eliminate redundancy.
Create Labeled Sequences
Subsequences labelled for three categories
normal, sensor fault, and contextual anomaly
provided to the BiLSTM model.
Stage 3: BiLSTM with Self-Attention
Bidirectional LSTM
Dual-direction sequence
processing with dropout regularization
Self-Attention Layer
Weights important temporal features
and sequences for anomaly detection
based on learned relevance scores
Predicted Labels
Multi-class classification
Class 0
Class 1
Class 2
Normal Points
Sensor Faults
Contextual Anomaly
Stage 4: Evaluation
Precision
Recall
F1-Score
ROC Curve
AUC

Dataset

The dataset used in this study is MIMIC-4 dataset. The dataset consists of detailed time series based medical records for ICU patients. The data we are concerned with are the physiological vitals: the pulse, blood oxygen saturation (SpO2), heart rate (HR), mean arterial blood pressure (ABPm), and the respiration rate (RR).

Preprocessing Pipeline:

1.Major Concerns of dataset preparation:

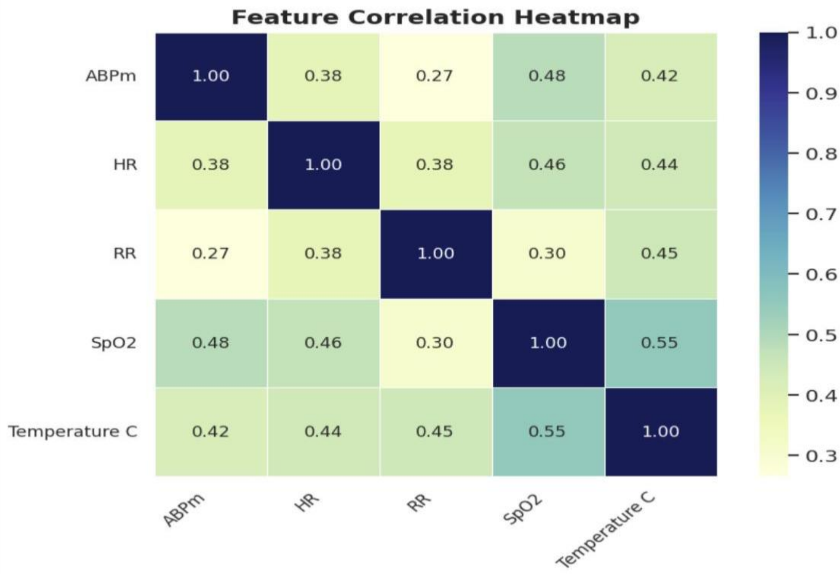
1.1 The dataset involves different patients with different physiological features.
1.2 Missing values for some timestamp and no common uniform reading based on timestamp
1.3 Less no of records for individual patients for model training.

2.Preprocessing pipeline:

2.1 To tackle the problem of missing values we first used backfill and forward fill method for imputation as this is most accurate approximation rather than filling features values with mean or mode , as latest reading to each missing values gives us better estimation for the timestamp.
2.2 For common physiological parameters or features we selected the parameters based on highest number of data records available.
2.3 For combining patients record for increasing number of data records for training we combined top 500 ICU patients records based on minimum number of null features.

3.Adding Point and Contextual Anomalies:

We added Point and Contextual anomalies by randomly selecting features based on their correlation and increasing their physiological values by adding noise . Then for point anomalies we used higher noise factor so model could better classify these anomalies and for Contextual anomalies we added noise over a continuous window of data points to simulate that subject was performing some heavy workout or energy consuming task during the timeframe.



Feature Correlation Heatmap
ABPm
HR
RR
SpO2
Temperature C
1.00
0.38
0.27
0.48
0.42
0.38
1.00
0.38
0.46
0.44
0.27
0.38
1.00
0.30
0.45
0.48
0.46
0.30
1.00
0.55
0.42
0.44
0.45
0.55
1.00
ABPm
HR
RR
SpO2
Temperature C
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3

CNN Autoencoders Model

The proposed model for anomaly detection is based on CNN Autoencoders, which are well-suited for reconstructing subsequence's of data in a square form. For this study, we focus on five physiological features: Heart Rate (HR), Blood Oxygen Saturation (SpO2), Respiration Rate (RR), Temperature (C), and Mean Arterial Blood Pressure (ABPm). The input data is structured into subsequence's of shape 5×5.

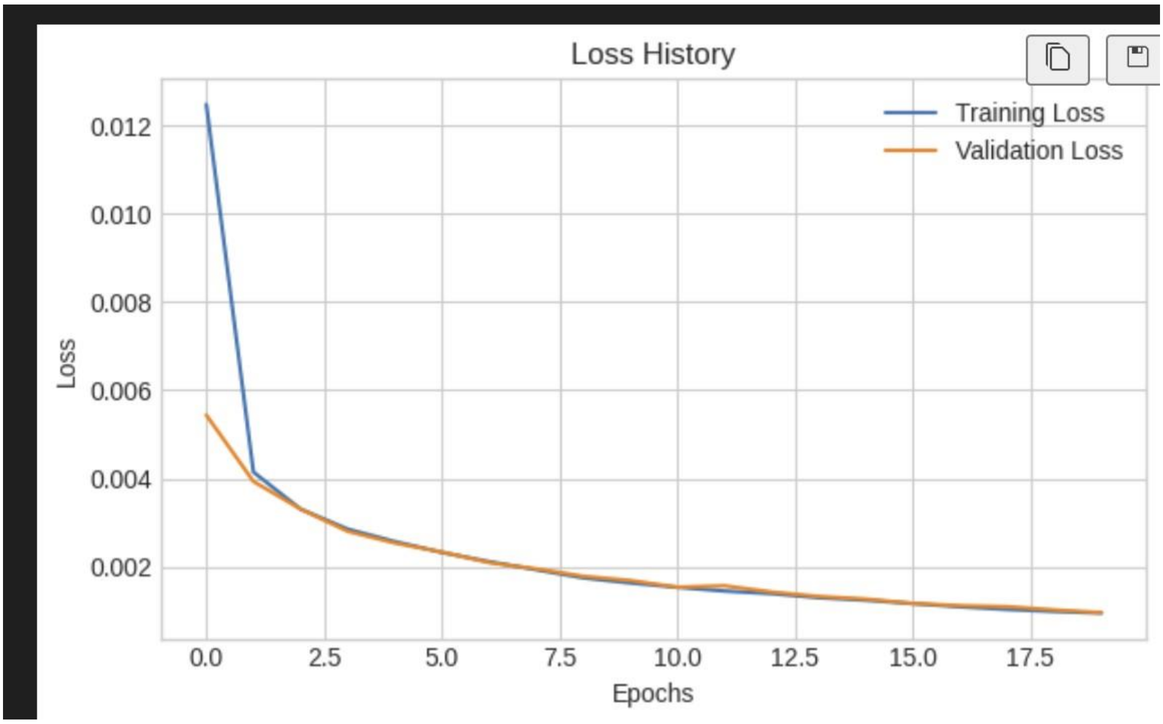
The encoder consists of five layers, including an input layer and two pairs of Convolutional and MaxPooling layers. The decoder mirrors the encoder with five layers, reconstructing the input data from the compressed latent representation. The Convolutional layers extract features and detect patterns, while the MaxPooling layers reduce spatial dimensions to enhance learning efficiency.

2.Model Architecture:

3.Training and Optimization

The training process involves optimizing the model for classification of normal and anomalous points .The key components include:

- Training data: The model is trained on non-anomalous data , so that it could better results while classifying anomalous and normal points.
- Learning Rate and Optimization: The Adam optimizer is used with a learning rate scheduler.
- Overfitting: Regularization techniques such as dropout and early stopping are applied to prevent overfitting.
- Loss function: Loss function used here is reconstruction error i.e. difference between value(predicted) −value(actual) > value(standard deviation of diff) for each feature .



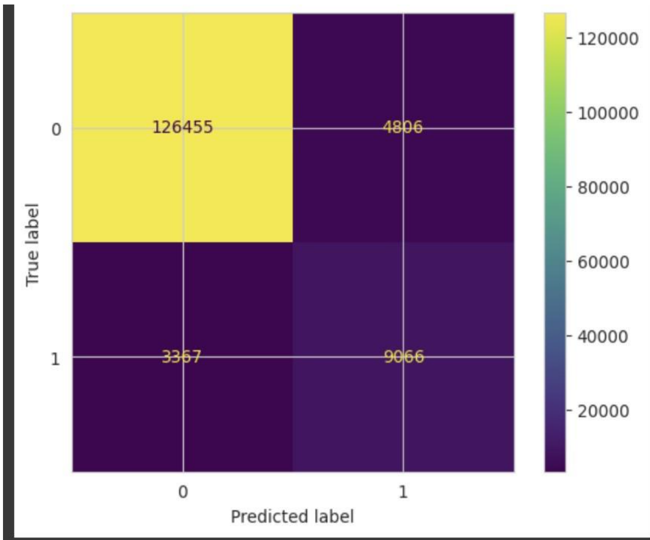
4.Evaluation Metrics

The model’s performance is evaluated on:

- Classification Metrics: Accuracy, precision, recall, and F1-score.
- Point Anomalies: Individual points for each feature which are flagged by model are combined and passed to second layer of arch for a fixed window size to further classify the point and contextual anomalies.
- Confusion Matrix: Analyzing physiological features values abnormality and patterns by calculating metrics such as TP,FP,TN,FN.

Results

Confusion Matrix:




```
##Metrics

print(f"Accuracy for point anomaly detection {accuracy_score(Y['point_anomaly'],anomalies_label)}")
print(f"Precision for point anomaly detection {precision_score(Y['point_anomaly'],anomalies_label)}")
print(f"Recall for point anomaly detection {recall_score(Y['point_anomaly'],anomalies_label)}")
```

```
Accuracy for point anomaly detection 0.9431221902097513
Precision for point anomaly detection 0.6535467128027682
Recall for point anomaly detection 0.7291884500924958
```

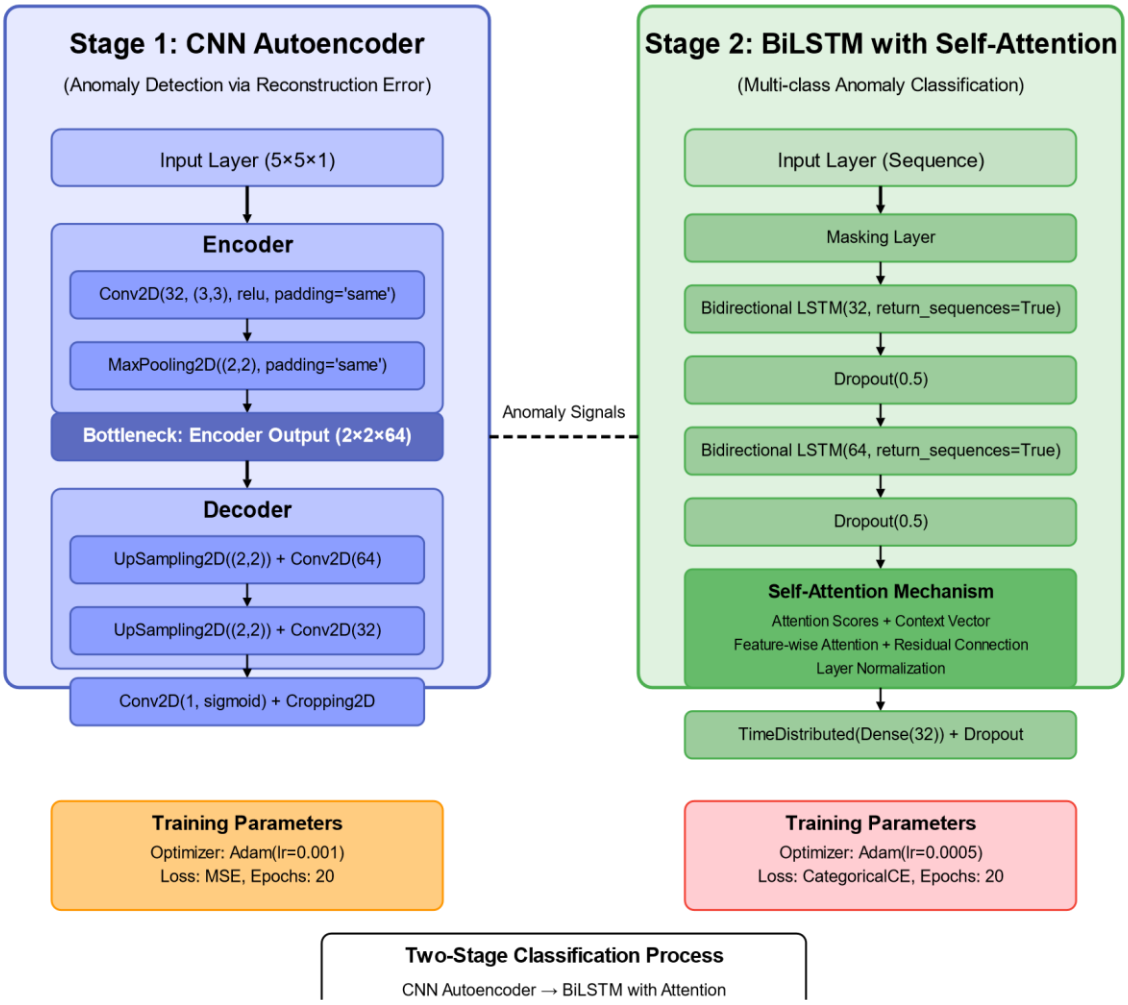
LSTM + SELF ATTENTION Model

The idea behind using LSTM models comes from fact that out dataset is time series dataset with long term time dependencies also called temporal dependencies. For capturing these long term dependencies RNN Layers such as LSTM becomes highly efficient. Also on further analysing our dataset and use-case we came to conclusion that since features are corelated due to activities involving significantly raising values of multiple features at a time Self Attention Layers could also be incorporated into this model. The model focuses on classifying marked anomalous points from CNN Autoencoders to label it as contextual or sensor faults by analysing long term and corelation among features i.e. temporal and spatial dependencies .Adding Self attention Layers gives attention weights to more coorelated features and helps us to capture relevant information of features and among same sequences , which provides us with better classification results .

The self-attention mechanism integrated with our LSTM model significantly enhanced the WBAN detection capabilities by allowing the network to focus on the most relevant temporal features. Unlike the sequential processing of LSTMs alone, self-attention created direct connections between all sequence positions, enabling the model to weigh the importance of different time steps based on their contextual relevance rather than just their sequential proximity. This proved particularly valuable for identifying subtle wireless body area network intrusion patterns that manifest as relationships between non-adjacent signal features or time points.

The multi-head self-attention component computed multiple attention distributions in parallel, giving the model the ability to jointly attend to information from different representation subspaces at different positions. Each attention head learned to recognize distinct patterns in the WBAN traffic data, with some heads focusing on short-term anomalies while others captured long-range dependencies in the signal. When combined with the LSTM's capability to maintain state information, this approach resulted in improvement in detection accuracy compared to traditional sequence models, especially for sophisticated intrusion attempts that manipulated temporal patterns across varying time scales.

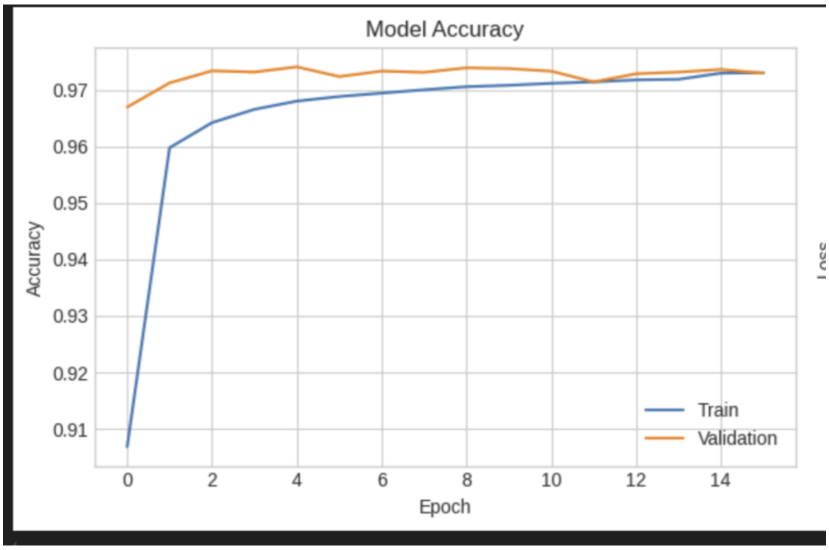
2.Model Architecture:

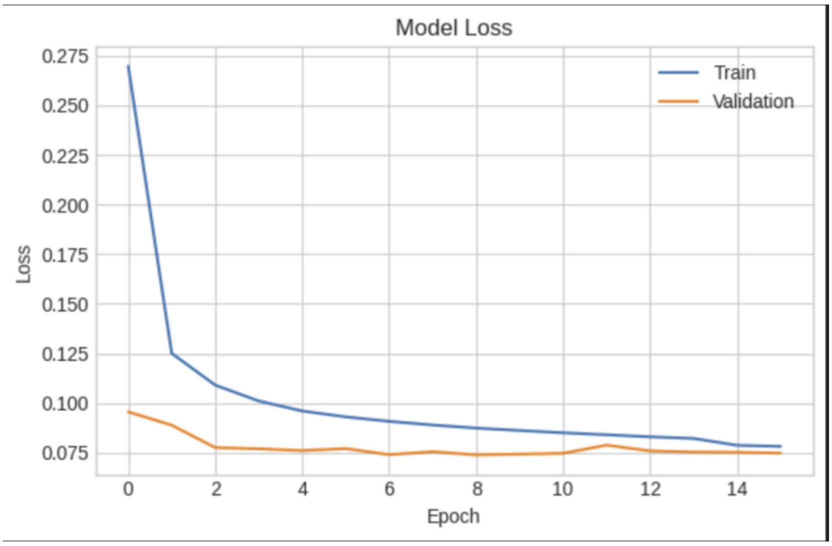


3.Training and Optimization

The training process involves optimizing the model for classification of contextual and sensor faults. The key components include:

- Training data: The model is trained on data that is marked by CNN Autoencoders , window size for a specified data points are passed to model as subsequences.
- Learning Rate and Optimization: The Adam optimizer is used with a learning rate scheduler.
- Overfitting: Regularization techniques such as dropout and early stopping are applied to prevent overfitting.
- Loss function: Loss function used here is sparse categorical cross entropy, as it is multiclass classification problem with activation function as SoftMax.





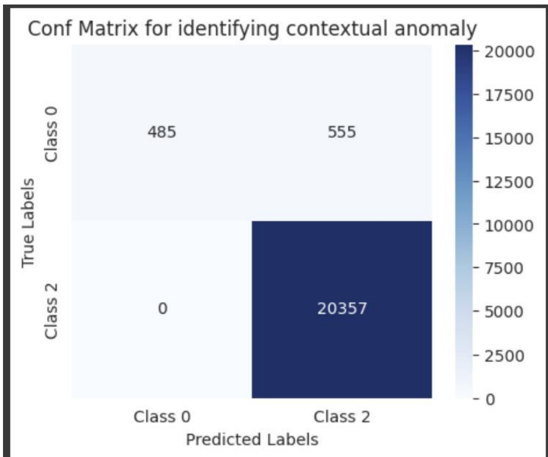
2. Evaluation Metrics

The model's performance is evaluated on:

- Classification Metrics: Accuracy, precision, recall, and F1-score.
- Contextual vs Sensor Faults vs Normal points : It gives us detailed idea about model is able to classify the point , sensor and contextual points.
- Confusion Matrix: Analyzing physiological features values abnormality and patterns by calculating metrics such as TP,FP,TN,FN.

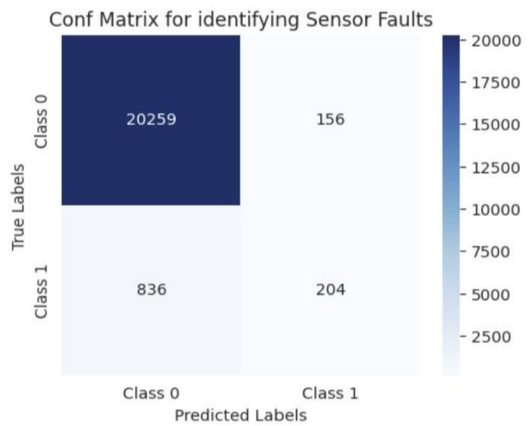
Results

Confusion Matrix:



Classification Report:

	precision	recall	f1-score	support
Class 0	1.00	0.47	0.64	1040
Class 2	0.97	1.00	0.99	20357
accuracy			0.97	21397
macro avg	0.99	0.73	0.81	21397
weighted avg	0.97	0.97	0.97	21397



Classification Report:

	precision	recall	f1-score	support
Class 0	0.96	0.99	0.98	20415
Class 1	0.57	0.20	0.29	1040
accuracy			0.95	21455
macro avg	0.76	0.59	0.63	21455
weighted avg	0.94	0.95	0.94	21455

Evaluation Metrics

The Architecture's performance will be evaluated using standard classification metrics:

- **Accuracy**: The percentage of correctly classified labels.
- **Precision and Recall**: These metrics will be critical, especially in medical diagnosis, where false positives (incorrectly diagnosing a healthy patient) and false negatives (failing to detect a heart condition) have serious implications.
- **F1-Score**: A balance between precision and recall, especially important for imbalanced datasets.
- **Confusion Matrix**: A detailed analysis of misclassifications, providing insights into common errors made by the model.