DRL-based Anomaly Detection in WBANs

Pratyush Bindal*
f20220119@hyderabad.bits-pilani.ac.in

Birla Institute of Technology and Science, Pilani
Hyderabad, Telangana, India

Abstract

Wireless Body Area Networks (WBANs) are critical in health monitoring, leveraging wearable sensors to collect physiological data. However, anomaly detection in WBANs is challenging due to the dynamic nature of human activity, sensor noise, and data variability. In this report, we meticulously compare various Deep Learning (DL) and Reinforcement Learning (RL) methods for anomaly detection in WBANs by integrating LSTM and transformer-based autoencoders with RL methods like Deep Q-Networks (DQN) and Advantage Actor-Critic (A2C). The proposed framework extracts sequential and periodic dependencies using the LSTM and Transformer frameworks and derives anomaly scores through reconstruction errors. These scores are then used to train a DRL agent, which optimises anomaly classification decisions based on learnt policies. We evaluate our approach on the PAMAP2 dataset and compare the efficacy of different models. Results indicate that the LSTM-Transformer-DQN model achieves superior accuracy, outperforming other architectures in anomaly classification. Integrating DRL with sequence modelling significantly enhances robustness, reducing false positives and improving generalisation across various activities. This work underscores the potential of DRL-based frameworks in WBAN anomaly detection and their applicability in real-world healthcare scenarios.

Keywords

Anomaly Detection, LSTM, Transformer, DRL, Autoencoders, DQN, PPO, A2C, WBAN

ACM Reference Format:
Pratyush Bindal. 2025. DRL-based Anomaly Detection in WBANs. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

*Both authors contributed equally to this research.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Wireless Body Area Networks (WBANs) have emerged as a transformative technology for continuous health monitoring, enabling real-time collection and analysis of physiological signals. These systems support applications such as chronic disease management, early detection of medical emergencies, and personalised healthcare. However, WBANs are prone to anomalies caused by sensor malfunctions, environmental interference, and irregular user behaviour. Detecting these anomalies accurately ensures reliable health monitoring and prevents erroneous clinical decisions.

Traditional anomaly detection methods rely on statistical models, threshold-based techniques, or classical machine learning approaches. These methods often struggle to adapt to physiological data's complex, non-stationary nature. Recent advancements in deep learning have enabled more sophisticated sequence modelling through architectures such as Long Short-Term Memory (LSTM) networks and Transformers. These models effectively capture temporal dependencies and contextual patterns, making them suitable for anomaly detection in WBANs.

This work proposes a novel DRL-based anomaly detection framework that integrates LSTM and transformer autoencoders with reinforcement learning. We leverage LSTM and transformer models to capture physiological data's local and global temporal dependencies, enhancing anomaly detection performance. Unlike conventional approaches that rely solely on reconstruction errors, our framework employs a DRL agent to optimise anomaly classification decisions based on learnt policies. Additionally, we compare the proposed framework against multiple baseline methods to determine the most effective approach for anomaly detection in WBANs.

By integrating DRL with sequence modelling, our approach improves generalisation and reduces false positives, making it a viable solution for real-world healthcare applications.

2 Overview

The proposed multi-context Deep Reinforcement Learning (DRL)-based framework represents a novel approach to detecting anomalies in Wireless Body Area Networks (WBANs). The architecture addresses the unique challenges of WBAN sensor data by integrating multiple temporal contexts through specialised autoencoders utilising deep learning (DL) techniques such as LSTM and transformers, and employing reinforcement learning (RL) for optimal decision-making. Fig. 1 illustrates the overall process of the proposed framework, which consists of five integrated stages: data preprocessing, sequential context modelling, periodic context modelling, feature fusion and DRL-based decision-making.

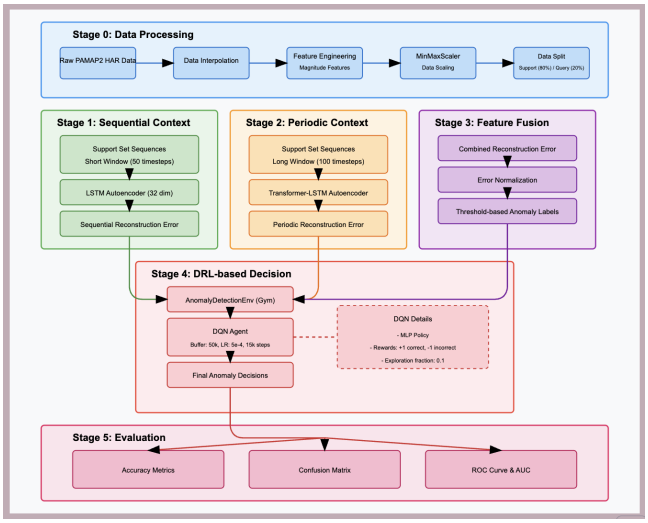

Figure 1: Detailed process overview for anomaly detection in WBANs

2.1 Reconstruction-Based Anomaly Detection

In this work, we used three different types of autoencoder architecture:

- **LSTM Autoencoder** models sequential dependencies using stacked long-short-term memory (LSTM) layers. The encoder compresses the input sequence into a latent representation while the decoder reconstructs the original input. Reconstruction errors, calculated as the mean absolute difference between the input and the output, are a quantitative measure of anomalous behaviour.
- **Transformer Autoencoder** utilises multihead self-attention mechanisms to capture periodic and global temporal patterns. The encoder-decoder structure incorporates transformer encoder layers with layer normalisation and dropout. Global average pooling is applied in the encoder to obtain a compact representation, and the decoder reconstructs the input sequence. Reconstruction errors are computed similarly to the previous case.
- **CNN Autoencoder** is implemented to capture point anomalies. The architecture comprises convolutional layers with ReLU activations, maximum pooling layers for dimensionality reduction, and upsampling and convolution layers with sigmoid activations to reconstruct the original input.

For each autoencoder, a threshold is determined using the 95th percentile of the reconstruction error computed over the training set. Sequences with errors exceeding this threshold are flagged as anomalous. In addition, reconstruction errors from models emphasising sequential (LSTM) and periodic (Transformer) contexts are aggregated to form a combined anomaly score.

2.2 Deep Reinforcement Learning for Anomaly Refinement

We implemented a custom Gym environment where the reconstructed error of each sequence serves as the observation. Binary anomaly labels are derived from the aggregated error using a fixed threshold. Several DRL agents like Deep Q-Networks (DQN) and Advantage Actor-Critic (A2C) are then trained to refine the classification of anomalies. The agents receive a reward signal based on the correctness of their anomaly detection decisions, and the best-performing architecture achieves high accuracy by leveraging the stability and sample efficiency of the underlying DRL algorithm.

3 Dataset Description

The experiments utilise the PAMAP2 Monitoring dataset, a widely used benchmark for Human Activity Recognition (HAR) with wearable sensors. The dataset comprises time-series data collected from 9 subjects performing 18 activities, using three Inertial Measurement Units (IMUs) placed on the hand, chest, and ankle, along with a heart rate monitor.

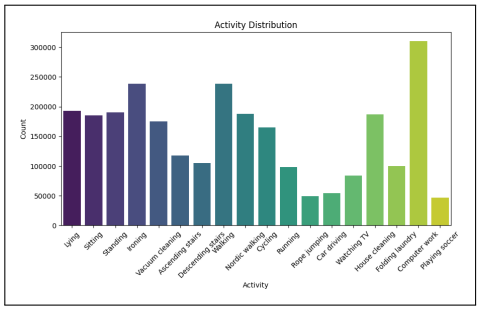

Figure 2: Activity distribution chart

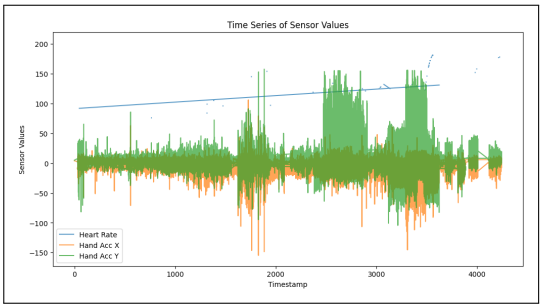

Figure 3: Time series plots for sensor values

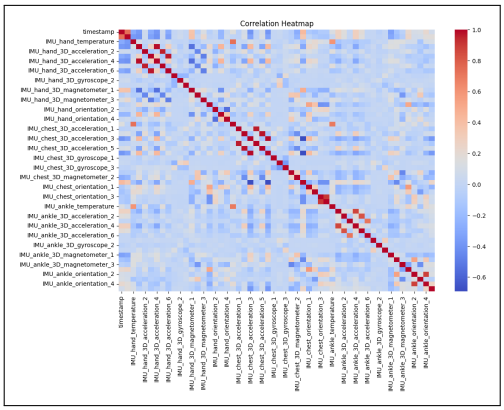

Figure 4: Correlation heatmap

Table 1 summarises the primary columns present in the raw dataset:

Column Name	Description
timestamp	Time (s)
heart_rate	Heart rate (bpm)
IMU_hand_temperature	Hand temperature reading (°C)
IMU_hand_3D_acceleration_1	Hand x-axis acceleration (m/s^2, ±16g scale)
IMU_hand_3D_acceleration_2	Hand y-axis acceleration (m/s^2, ±16g scale)
IMU_hand_3D_acceleration_3	Hand z-axis acceleration (m/s^2, ±16g scale)
IMU_hand_3D_acceleration_4	Hand x-axis acceleration (m/s^2, ±6g scale)
IMU_hand_3D_acceleration_5	Hand y-axis acceleration (m/s^2, ±6g scale)
IMU_hand_3D_acceleration_6	Hand z-axis acceleration (m/s^2, ±6g scale)
IMU_hand_3D_gyroscope_1	Hand x-axis angular velocity (rad/s)
IMU_hand_3D_gyroscope_2	Hand y-axis angular velocity (rad/s)
IMU_hand_3D_gyroscope_3	Hand z-axis angular velocity (rad/s)
IMU_hand_3D_magnetometer_1	Hand x-axis magnetometer reading (μT)
IMU_hand_3D_magnetometer_2	Hand y-axis magnetometer reading (μT)
IMU_hand_3D_magnetometer_3	Hand z-axis magnetometer reading (μT)
IMU_hand_orientation_1	Specific orientation of the hand
IMU_hand_orientation_2	Specific orientation of the hand
IMU_hand_orientation_3	Specific orientation of the hand
IMU_hand_orientation_4	Specific orientation of the hand
IMU_chest_temperature	Chest temperature reading (°C)
IMU_chest_3D_acceleration_1	Chest x-axis acceleration (m/s^2, ±16g scale)
IMU_chest_3D_acceleration_2	Chest y-axis acceleration (m/s^2, ±16g scale)
IMU_chest_3D_acceleration_3	Chest z-axis acceleration (m/s^2, ±16g scale)
IMU_chest_3D_acceleration_4	Chest x-axis acceleration (m/s^2, ±6g scale)
IMU_chest_3D_acceleration_5	Chest y-axis acceleration (m/s^2, ±6g scale)
IMU_chest_3D_acceleration_6	Chest z-axis acceleration (m/s^2, ±6g scale)
IMU_chest_3D_gyroscope_1	Chest x-axis angular velocity (rad/s)
IMU_chest_3D_gyroscope_2	Chest y-axis angular velocity (rad/s)
IMU_chest_3D_gyroscope_3	Chest z-axis angular velocity (rad/s)
IMU_chest_3D_magnetometer_1	Chest x-axis magnetometer reading (μT)
IMU_chest_3D_magnetometer_2	Chest y-axis magnetometer reading (μT)
IMU_chest_3D_magnetometer_3	Chest z-axis magnetometer reading (μT)
IMU_chest_orientation_1	Specific orientation of the chest
IMU_chest_orientation_2	Specific orientation of the chest
IMU_chest_orientation_3	Specific orientation of the chest
IMU_chest_orientation_4	Specific orientation of the chest
IMU_ankle_temperature	Ankle temperature reading (°C)
IMU_ankle_3D_acceleration_1	Ankle x-axis acceleration (m/s^2, ±16g scale)
IMU_ankle_3D_acceleration_2	Ankle y-axis acceleration (m/s^2, ±16g scale)
IMU_ankle_3D_acceleration_3	Ankle z-axis acceleration (m/s^2, ±16g scale)
IMU_ankle_3D_acceleration_4	Ankle x-axis acceleration (m/s^2, ±6g scale)
IMU_ankle_3D_acceleration_5	Ankle y-axis acceleration (m/s^2, ±6g scale)
IMU_ankle_3D_acceleration_6	Ankle z-axis acceleration (m/s^2, ±6g scale)
IMU_ankle_3D_gyroscope_1	Ankle x-axis angular velocity (rad/s)
IMU_ankle_3D_gyroscope_2	Ankle y-axis angular velocity (rad/s)
IMU_ankle_3D_gyroscope_3	Ankle z-axis angular velocity (rad/s)
IMU_ankle_3D_magnetometer_1	Ankle x-axis magnetometer reading (μT)
IMU_ankle_3D_magnetometer_2	Ankle y-axis magnetometer reading (μT)
IMU_ankle_3D_magnetometer_3	Ankle z-axis magnetometer reading (μT)
IMU_ankle_orientation_1	Specific orientation of ankle
IMU_ankle_orientation_2	Specific orientation of ankle
IMU_ankle_orientation_3	Specific orientation of ankle
IMU_ankle_orientation_4	Specific orientation of ankle
activityID	Numeric activity label
activity_name	Human-readable activity label

Table 1: Dataset Description

4 Methodology

The methodology consists of four main stages: data preprocessing and feature engineering, sequence generation and data splitting, reconstruction-based anomaly detection using autoencoders, and DRL-based decision refinement.

4.1 Data Preprocessing and Feature Engineering

We first import raw sensor data from the PAMAP2 Human Activity Recognition dataset. We use linear interpolation to address missing values, and subsequent removal of any remaining NaNs ensures data integrity. Then, we perform feature engineering on the dataset by computing the Euclidean norm of multi-dimensional inertial measurement unit signals, including acceleration, gyroscope, and magnetometer readings across different body parts. Redundant raw sensor channels are then dropped. Moreover, the resultant feature matrix is normalised using min–max scaling to ensure that all features lie within a consistent range, thereby facilitating the convergence of neural network models.

4.2 Sequence Generation and Data Splitting

The preprocessed dataset is partitioned into support (training) and query (testing) sets based on an 80-20 split criterion. Sequential data samples are generated by applying a sliding window approach to form time-series sequences using two sequence lengths: a shorter window (30 - 50 timesteps) for capturing sequential context and a longer window (100 timesteps) to model periodic patterns.

4.3 Proposed Architecture

As shown in Fig. 5, the proposed architecture implements a hierarchical, multi-level anomaly detection system for HAR data. We combine deep unsupervised representation-based models, such as LSTM and transformer autoencoders, with deep reinforcement learning (DRL) frameworks like Deep Q-Network (DQN) to create a robust anomaly detection framework.

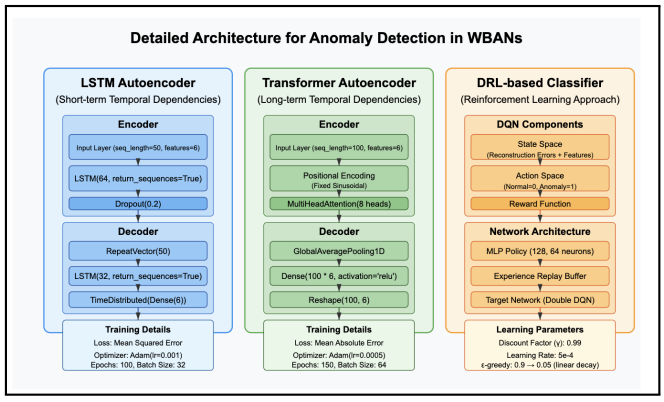

Figure 5: Detailed architecture for anomaly detection in WBANs

4.3.1 *Short-term Temporal Dependencies (Sequential Context)*

. We employ an LSTM-based autoencoder to model short-term temporal dependencies in the sensor data, which is crucial for capturing local patterns and immediate sequential relationships between consecutive sensor readings.

The LSTM autoencoder consists of an encoder-decoder structure designed to reconstruct input sequences, consisting of an input layer which accepts sequences with 50-timestep windows of sensor data. Then, a single LSTM layer, along with the ReLU activation function, is utilised as an encoder, which helps to capture temporal patterns in the input sequence and then compress them into a fixed-dimension representation. A dropout layer is also incorporated to reduce the chances of overfitting. Next, a repeat vector layer replicates the

encoded representation, which is followed by an LSTM layer acting as a decoder that helps to reconstruct the original sequence.

The LSTM autoencoder is trained on the support set using the mean squared error loss function and Adam optimiser. After training, we apply the model to the query set to compute reconstruction errors that act as anomaly scores. High reconstruction errors indicate that the sequences deviate from standard learned patterns, potentially representing anomalous behaviour.

Let the input sequence be represented by $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$, where $\mathbf{x}_t \in \mathbb{R}^d$ is the sensor input at time t.

The encoder LSTM updates its internal states as follows:

$$\begin{aligned}
\mathbf{f}_t &= \sigma\left(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f\right) \\
\mathbf{i}_t &= \sigma\left(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o\right) \\
\tilde{\mathbf{c}}_t &= \tanh\left(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c\right) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}$$

After encoding, the final hidden state $\mathbf{h}_T$ is repeated T times and fed into the decoder LSTM to generate the reconstructed sequence $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_T\}$.

The reconstruction loss is calculated using the mean squared error (MSE):

$$\mathcal{L}_{\text{recon}} = \frac{1}{T} \sum_{t=1}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2$$

The anomaly score for a query sequence is defined as:

$$\text{AnomalyScore}(\mathbf{X}) = \mathcal{L}_{\text{recon}}(\mathbf{X}, \hat{\mathbf{X}})$$

Higher values of the anomaly score suggest potential deviations from the learned normal patterns.

4.3.2 *Long-term Temporal Dependencies (Periodic Context)*

. We utilise a transformer-based autoencoder to capture the sensor data's long-range dependencies and periodic patterns. This component complements the LSTM autoencoder by modelling relationships between distant time points, which is essential for detecting complex anomalies spread over long time horizons.

The Transformer-LSTM autoencoder combines the attention mechanism of transformers with the sequential modelling capability of LSTMs. It consists of an input layer accepting sequences with 100-timestep windows of sensor data to capture extended temporal patterns. A multi-head attention mechanism acts as an encoder which learns the time-distant relationships. Normalised attention outputs are pooled across the temporal dimension to create a global representation. Next, a repeat vector layer followed by an LSTM acts as a decoder, which helps to reconstruct the whole sequence, with a final dense layer producing the output. The Transformer-LSTM autoencoder is trained on more extended sequences from the support set with larger batch sizes to accommodate increased computational requirements.

Like the LSTM autoencoder, we compute the reconstruction errors on the query set as anomaly scores. The attention mechanism enables the model to identify complex patterns that might be missed by the LSTM autoencoder, providing complementary anomaly detection capabilities.

Let the input sequence be $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T] \in \mathbb{R}^{T \times d}$, where $T = 100$. Positional encodings $\mathbf{P} \in \mathbb{R}^{T \times d}$ are added to the inputs:

$$\mathbf{Z}_0 = \mathbf{X} + \mathbf{P}$$

Each Transformer encoder layer computes:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

$$\mathbf{Q} = \mathbf{Z}_l \mathbf{W}_Q, \quad \mathbf{K} = \mathbf{Z}_l \mathbf{W}_K, \quad \mathbf{V} = \mathbf{Z}_l \mathbf{W}_V$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are learnable projection matrices, and d_k is the dimensionality of keys.

Multi-head attention combines multiple such heads:

$$\text{MultiHead}(\mathbf{Z}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_O$$

Each head is an independent attention mechanism:

$$\text{head}_i = \text{Attention}(\mathbf{Z}\mathbf{W}_Q^i, \mathbf{Z}\mathbf{W}_K^i, \mathbf{Z}\mathbf{W}_V^i)$$

After applying feedforward layers and normalisation, we obtain a contextualised representation $\mathbf{Z}_L$ from the final encoder layer. This is pooled over time:

$$\mathbf{z}_{\text{global}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{Z}_L[t]$$

The pooled vector is repeated across T steps and fed into the LSTM decoder to reconstruct the sequence $\hat{\mathbf{X}} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_T\}$.

The reconstruction loss is computed similarly using MSE:

$$\mathcal{L}_{\text{recon}} = \frac{1}{T} \sum_{t=1}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2$$

Anomaly scores for query sequences are:

$$\text{AnomalyScore}(\mathbf{X}) = \mathcal{L}_{\text{recon}}(\mathbf{X}, \hat{\mathbf{X}})$$

This formulation enables the model to learn periodic or long-range dependencies not easily captured by LSTMs alone.

4.3.3 *Multi-Context Fusion*

. We integrate the anomaly scores from both autoencoder models to create a unified representation that leverages their complementary strengths. The fusion approach helps to combine information from different temporal contexts, enabling more robust anomaly detection. The combined errors are then normalised to the [0,1] range to create standardised anomaly scores, and based on the normalised scores, we establish initial anomaly labels using a threshold at the 80th percentile. These labels serve as a baseline for training the RL component, further refining anomaly detection decisions.

Let $\mathcal{A}_{\text{LSTM}}(\mathbf{X})$ and $\mathcal{A}_{\text{Trans}}(\mathbf{X})$ be the reconstruction-based anomaly scores from the LSTM and Transformer autoencoders respectively. The combined score is computed as:

$$\mathcal{A}_{\text{fused}}(\mathbf{X}) = \alpha \cdot \mathcal{A}_{\text{LSTM}}(\mathbf{X}) + (1 - \alpha) \cdot \mathcal{A}_{\text{Trans}}(\mathbf{X})$$

where $\alpha \in [0, 1]$ is a weighting coefficient controlling the fusion bias.

To standardise scores, we perform min-max normalisation:

$$\mathcal{A}_{\text{norm}}(\mathbf{X}) = \frac{\mathcal{A}_{\text{fused}}(\mathbf{X}) - \min(\mathcal{A}_{\text{fused}})}{\max(\mathcal{A}_{\text{fused}}) - \min(\mathcal{A}_{\text{fused}})}$$

Initial anomaly labels are determined by applying a threshold at the 80th percentile of normalised scores:

$$\text{Label}(\mathbf{X}) = \begin{cases} 1, & \text{if } \mathcal{A}_{\text{norm}}(\mathbf{X}) \geq \text{Percentile}_{80} \\ 0, & \text{otherwise} \end{cases}$$

These binary labels are then used to guide the reward signals for the reinforcement learning-based refinement module.

4.3.4 ***DRL-based Learning for Decision Optimisation.*** The final stage of the framework employs DRL to optimise the anomaly detection decision-making process, which offers significant advantages over static thresholds by learning adaptive decision boundaries that aim to maximise detection accuracy. We implemented a custom Open AI Gym environment that encapsulated anomaly detection problems as a sequential decision process in which a Deep Q-Network (DQN) agent learns the optimal policy for anomaly classification based on normalised reconstruction errors. We train the DQN agent for 15,000 timesteps, during which it learns to discriminate between normal and anomalous patterns based on the reward signals. After training, the agent's learned policy makes the final anomaly detection decisions.

The experimental results demonstrate that proposed framework performs better than traditional threshold-based approaches as the DRL component effectively learns the adaptive decision boundaries, maximising detection accuracy while minimising false alarms.

We formulate the anomaly detection process as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$, where:

- $\mathcal{S}$: State space — Each state $s_t \in \mathcal{S}$ is defined as the normalised anomaly score $\mathcal{A}_{\text{norm}}(\mathbf{X}_t)$.
- $\mathcal{A}$: Action space — Binary actions $a_t \in \{0, 1\}$ indicating whether the instance is normal or anomalous.
- $\mathcal{R}$: Reward function — Defined to reinforce correct classifications:

$$r_t = \begin{cases} +1, & \text{if correct classification (TP or TN)} \\ -1, & \text{if incorrect classification (FP or FN)} \end{cases}$$

- $\mathcal{P}$: Transition probabilities that are determined by the sequence of instances.
- γ: Discount factor controls long-term reward prioritisation.

The agent learns a Q-function $Q(s_t, a_t; \theta)$ that estimates the expected cumulative reward:

$$Q(s_t, a_t) = \mathbb{E}\left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') \mid s_t, a_t \right]$$

The DQN updates its parameters θ by minimising the temporal-difference loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} \left[\left(r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta) \right)^2 \right]$$

where θ^- are the parameters of a target network periodically updated to stabilise learning, and $\mathcal{D}$ is the replay buffer. The final anomaly label for a sample $\mathbf{X}_t$ is:

$$\text{Label}_{\text{DQN}}(\mathbf{X}_t) = \arg \max_{a \in \{0,1\}} Q(\mathcal{A}_{\text{norm}}(\mathbf{X}_t), a)$$

This DRL-driven decision process allows the model to dynamically adjust detection thresholds and optimise performance based on feedback from its environment.

4.4 Algorithm

The proposed architecture leverages multiple temporal modelling techniques and reinforcement learning to perform robust, context-aware anomaly detection. Each stage addresses a distinct type of temporal dependency and detection challenge:

- **Short-Term Sequential Modelling:** LSTM autoencoders effectively capture recent temporal dependencies through their gating mechanism, reconstructing short sequences to detect anomalies arising from sudden changes in behaviour.
- **Long-Term Periodic Modelling:** Transformer-LSTM hybrid autoencoders address long-range and periodic patterns by applying self-attention for global context modelling, followed by sequential decoding to reconstruct extended input sequences.
- **Multi-Context Fusion:** Combining reconstruction errors from both models integrates both local and global anomaly evidence. Percentile-based thresholding generates weak labels from the fused scores.
- **Adaptive Decision Optimisation:** A Deep Q-Network (DQN) agent refines the detection process by learning to map fused anomaly scores to detection actions. The agent is trained via reinforcement learning, optimising long-term reward through adaptive policy learning.

The entire flow is formalised in Algorithm 1.

Algorithm 1 Multi-Context Anomaly Detection with DRL Optimisation

Require: Sensor data $X = \{x_t\}_{t=1}^{T}$

Split X into support set X_s and query set X_q
// Short-Term Modeling with LSTM Autoencoder
Train LSTM Autoencoder M_{LSTM} on X_s (50-timestep windows)
$E_{\text{LSTM}} \leftarrow \text{ReconstructionError}(M_{\text{LSTM}}, X_q)$
// Long-Term Modeling with Transformer-LSTM Autoencoder
Train Transformer-LSTM Autoencoder M_{Trans} on X_s (100-timestep windows)
$E_{\text{Trans}} \leftarrow \text{ReconstructionError}(M_{\text{Trans}}, X_q)$
// Multi-Context Fusion
$E_{\text{fused}} \leftarrow \text{Normalize}(E_{\text{LSTM}} + E_{\text{Trans}})$
Define pseudo-labels y_{pseudo} using 80th percentile threshold
// DRL-based Anomaly Decision Optimization
Initialize DQN agent π_θ with experience buffer $\mathcal{D}$
for each timestep $t = 1$ to N **do**
    Observe state $s_t \leftarrow E_{\text{fused}}[t]$
    Choose action $a_t \leftarrow \pi_\theta(s_t)$ ▷ 0: Normal, 1: Anomaly
    Receive reward r_t based on a_t vs. $y_{\text{pseudo}}[t]$
    Store (s_t, a_t, r_t, s_{t+1}) in buffer $\mathcal{D}$
    Update π_θ via Q-learning from $\mathcal{D}$
return Final anomaly predictions from π_θ

Legend:

Symbol	Description
X	Full input time series
X_s, X_q	Support and query sets
$M_{\text{LSTM}}, M_{\text{Trans}}$	LSTM and Transformer-LSTM autoencoder models
$E_{\text{LSTM}}, E_{\text{Trans}}$	Reconstruction errors from respective models
E_{fused}	Normalised sum of both error scores
y_{pseudo}	Weak labels based on 80th percentile threshold
π_θ	DQN agent policy
a_t	Action at timestep t (0: normal, 1: anomaly)
r_t	Reward at timestep t
$\mathcal{D}$	Experience replay buffer
N	Total timesteps in training episode

5 Comparative Analysis

We analysed multiple methods based on LSTM, transformer, and hybrid architectures integrated with various reinforcement learning techniques.

5.1 LSTM-FIF (Feedback-based Isolation Forest)

This approach utilises an LSTM-based autoencoder that learns representative temporal patterns by compressing and accurately representing input sequences. The reconstruction errors are computed and leveraged within a modified Feedback-based Isolation Forest to flag anomalies, with the 95th percentile threshold serving as a reference for anomalous behaviour. We trained for 10 epochs and detected 4980 anomalies in the entire dataset.

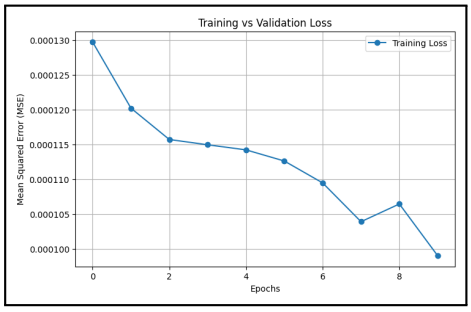

Figure 6: LSTM-FIF Training Loss Curve

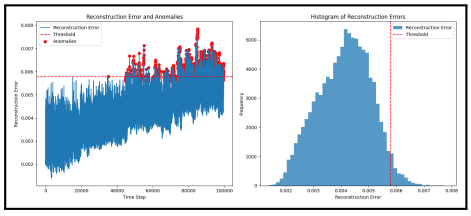

Figure 7: LSTM-FIF Anomaly Plot and Histogram of Reconstruction Errors

5.2 LSTM-PPO (Proximal Policy Optimisation)

An LSTM autoencoder, designed with stacked LSTM layers and a corresponding decoder, is trained to reconstruct input sequences. Reconstruction errors are computed as the mean absolute difference between the original and reconstructed sequences, and these errors serve to derive unsupervised rewards for further training. A PPO agent, implemented with fully connected layers, is subsequently trained using these rewards to refine anomaly classification. Finally, we evaluate the test set by thresholding the reconstruction errors at the 95th percentile. We trained the LSTM model for 20 epochs and the PPO algorithm for 10 epochs, obtaining 5980 anomalies in the entire dataset.

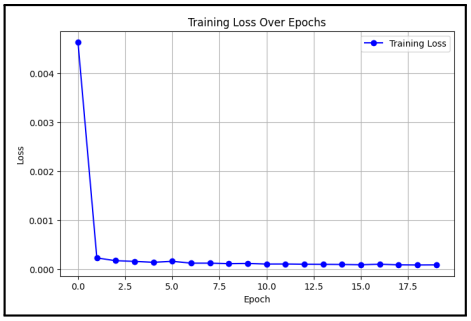

Figure 8: LSTM Training Loss Curve

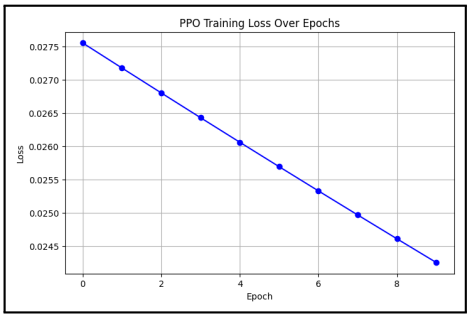

Figure 9: PPO Training Loss Curve

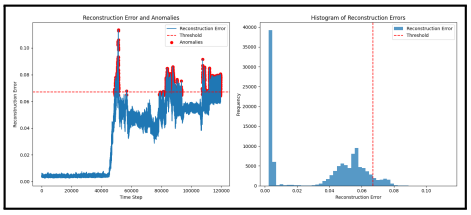

Figure 10: LSTM-PPO Anomaly Plot and Histogram of Reconstruction Errors

5.3 Transformer-PPO (Proximal Policy Optimisation)

The autoencoder employs transformer encoder layers featuring multi-head self-attention, feed-forward networks, layer normalisation, and dropout to learn compact representations and reconstruct input sequences. Reconstruction errors, computed as the

mean absolute difference between original and reconstructed sequences, are leveraged to derive unsupervised reward signals via 95th-percentile thresholding. A PPO agent, implemented with dense layers, is trained using these rewards to refine the anomaly detection process. We trained the transformer architecture for 20 epochs and the PPO model for 10 epochs, obtaining 5380 anomalies in the entire dataset.

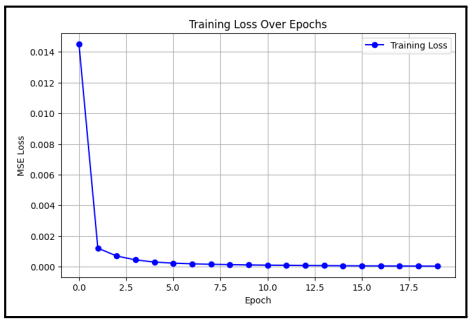

Figure 11: Transformer Training Loss Curve

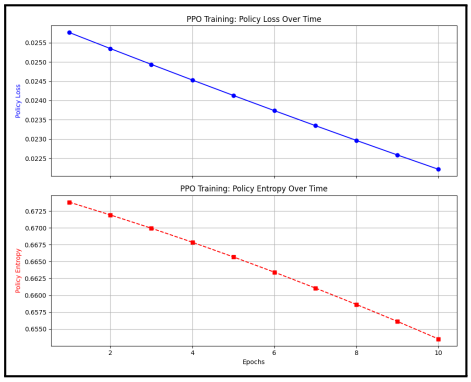

Figure 12: PPO Training Loss Curve and Entropy Plot

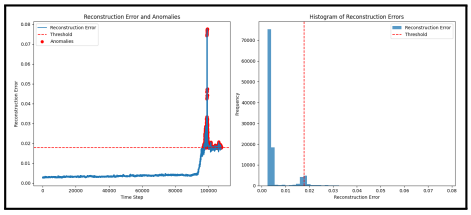

Figure 13: Transformer-PPO Anomaly Plot and Histogram of Reconstruction Errors

The experimental results indicate that three methods – LSTM-PPO, LSTM-FIF, and Transformer-PPO yield varying anomaly counts over the entire dataset. The LSTM-PPO and LSTM-FIF approaches use sequential context, effectively capturing temporal dynamics. However, by focusing solely on sequential information, these methods may overlook anomalies resulting from periodic patterns or mixed contextual features. Conversely, Transformer-PPO emphasises periodic context, which is beneficial for capturing recurring temporal patterns but may miss anomalies primarily defined by sequential dependencies. These observations suggest that each method, while robust within its designated context, might not fully capture mixed-context anomalies present in complex sensor datasets.

Therefore, a potential future direction would be integrating sequential and periodic features to achieve a more comprehensive anomaly detection framework. Furthermore, due to computational constraints, we restrict further models discussed to the first 25000 entries of the dataset.

5.4 CNN AutoEncoder

This approach uses a CNN autoencoder that comprises convolutional, max-pooling, and upsampling layers to reconstruct input sequences. We show reconstruction errors using the mean absolute difference between original and reconstructed sequences, with anomalies marked as sequences exceeding the 95th percentile threshold. We trained for 20 epochs and detected 249 anomalies.

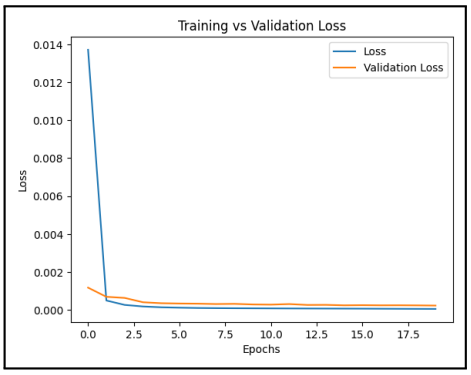

Figure 14: CNN AutoEncoder Training vs. Validation Loss Curve

Experimental results reveal that the CNN AutoEncoder detects 249 point anomalies. The model's focus on individual data points results in a limited capacity to capture collective anomalies. Hence, while this method effectively identifies isolated outliers, it may overlook anomalies from temporal or contextual dependencies. This limitation arises because CNN-based approaches primarily exploit spatial features but lack the sequential modelling required for time-series data. Therefore, to address these challenges, we build advanced DRL-based architectures that can better capture complex temporal dynamics and improve anomaly detection performance in intricate datasets.

5.5 LSTM-Transformer-A2C (Advantage Actor-Critic)

In this model, two autoencoder models are employed: an LSTM autoencoder for capturing short-term sequential context and a transformer autoencoder for modelling long-term periodic patterns. Reconstruction errors from both models are computed, normalised, and aggregated to generate final anomaly labels. A custom Gym environment is defined using these aggregated errors, and an A2C agent is trained to refine anomaly detection decisions. The performance of the DRL agent is evaluated by comparing its actions with the derived anomaly labels.

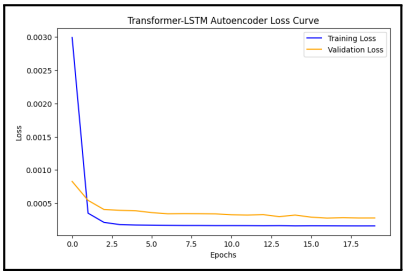

Figure 15: LSTM-Transformer Training vs. Validation Loss Curve

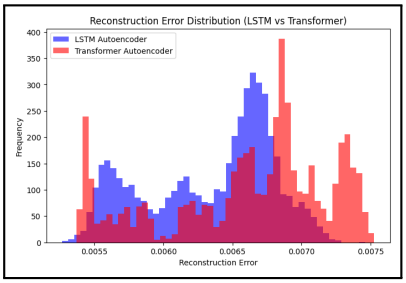

Figure 16: Combined Reconstruction Error Distribution

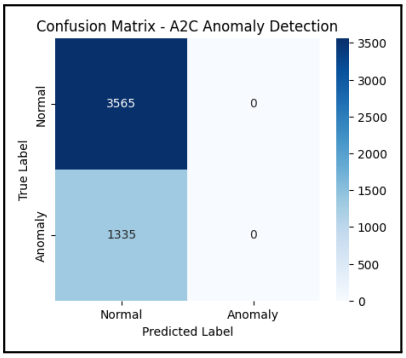

Figure 17: LSTM-Transformer-A2C Confusion Matrix

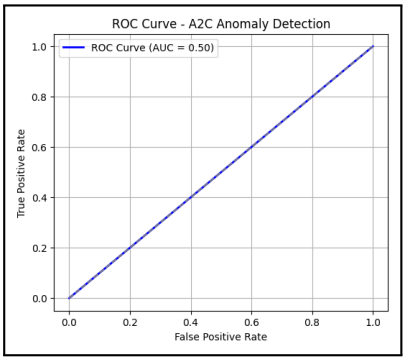

Figure 18: LSTM-Transformer-A2C ROC Curve

The A2C agent, when combined with LSTM and transformer autoencoder features, yielded 1335 detected anomalies at a 72.76% accuracy. This suboptimal performance is primarily due to the inherent sample inefficiency of the A2C algorithm in complex time-series environments. The diverse sequential and periodic patterns in the data require extensive sampling to converge effectively, and the limited data interactions hinder the agent's ability to discriminate between normal and anomalous events accurately.

Furthermore, the stochastic nature of policy gradient methods like A2C often leads to high variance in training, which can impede stable learning in scenarios with sparse or delayed rewards, such as anomaly detection. The complexity of temporal dependencies further exacerbates this challenge, as the agent must infer subtle deviations over long horizons. Additionally, the exploration-exploitation balance in A2C can result in slower convergence when faced with highly imbalanced anomaly distributions, often causing the agent to under-explore rare but critical anomalous patterns. These limitations suggest that while A2C provides a principled approach to reinforcement learning, alternative algorithms with improved sample efficiency may be better suited for intricate anomaly detection tasks in time-series data. Addressing these challenges requires integrating techniques such as prioritised experience replay or combining A2C with model-based components to enhance learning efficiency and robustness.

5.6 LSTM-Transformer-DQN (Deep Q-Networks)

Building upon the dual-autoencoder framework, this approach integrates LSTM and transformer architectures to capture both short-term sequential dependencies and long-term periodic patterns in time-series data. The LSTM autoencoder models fine-grained temporal dependencies within short windows, while the transformer autoencoder attends to longer sequences to capture global temporal features.

Reconstruction errors from both models are normalised and aggregated to form a unified anomaly score, which is used to label segments as normal or anomalous via a statistical threshold. To go beyond static thresholding, a custom OpenAI Gym environment is defined where these aggregated scores serve as state inputs. A Deep Q-Network (DQN) agent is trained with reward signals based on alignment between its decisions and ground-truth anomaly labels derived from reconstruction errors.

The DQN agent learns complex decision boundaries, improving anomaly detection in cases where patterns are context-dependent. Training leverages experience replay and target networks to enhance stability and sample efficiency, allowing the model to adapt dynamically to evolving data distributions and reduce false positives. By continuously interacting with the environment and receiving feedback, the agent refines its policy to better distinguish true anomalies from benign deviations, resulting in more reliable and interpretable detection outcomes. Evaluation against threshold-based methods demonstrates improved accuracy, precision, recall, and robustness, showcasing the potential of combining deep sequential modelling with reinforcement learning for complex, non-stationary, and noisy time-series anomaly detection. This adaptability makes the framework well-suited for real-world applications such as industrial monitoring, cybersecurity, and predictive maintenance, where anomaly characteristics evolve over time.

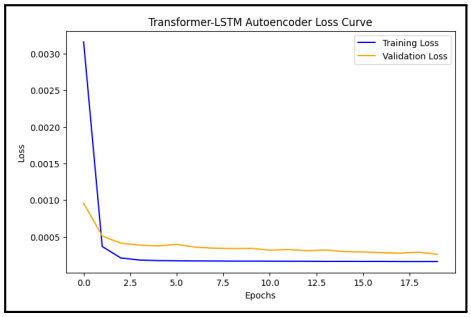

Figure 19: LSTM-Transformer Training vs. Validation Loss Curve

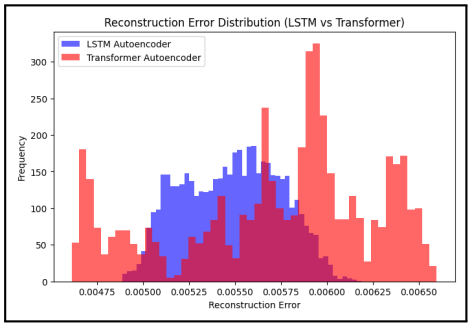

Figure 20: Combined Reconstruction Error Distribution

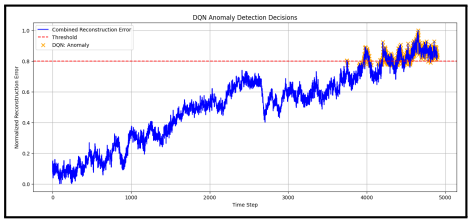

Figure 21: LSTM-Transformer-DQN Anomaly Plot

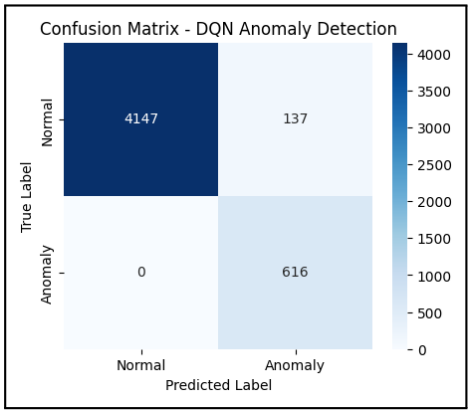

Figure 22: LSTM-Transformer-DQN Confusion Matrix

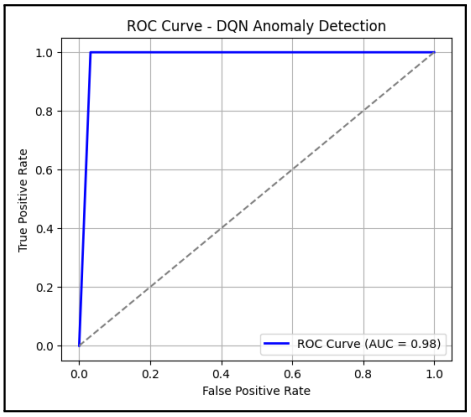

Figure 23: LSTM-Transformer-DQN ROC Curve

The LSTM-Transformer-DQN method achieved 753 detected anomalies with a classification accuracy of 97.20%. This superior performance is attributed to the stability of the DQN algorithm, which is employed as the reinforcement learning component in this hybrid framework. DQN exhibits enhanced sample efficiency and robust convergence properties in complex time-series environments compared to A2C. The hybrid design integrates LSTM and transformer autoencoders for capturing sequential and periodic patterns, which ensures a comprehensive data representation, while DQN's stable training dynamics facilitate reliable anomaly detection decisions. Consequently, the LSTM-Transformer-DQN approach outperforms alternative methods by achieving a high accuracy rate and a consistent anomaly detection capability.

6 Conclusion

The increasing reliance on Wireless Body Area Networks (WBANs) for continuous health monitoring necessitates strengthened anomaly detection mechanisms to ensure data security and reliability. This study explored integrating deep learning and reinforcement learning techniques to enhance anomaly detection performance in WBANs. By leveraging LSTM and transformer-based autoencoders alongside reinforcement learning agents, the proposed framework effectively captured both sequential and periodic dependencies in physiological data.

Experimental evaluations demonstrated that deep reinforcement learning-based models, particularly the LSTM-Transformer-DQN approach, significantly outperformed traditional methods regarding accuracy and robustness. The results underscore the effectiveness of combining sequential modelling with reinforcement learning-driven decision refinement, improving anomaly classification while minimising false positives.

Future research can explore lightweight architectures to enhance real-time applicability and investigate self-supervised learning methods to reduce reliance on labelled anomaly data. Extending this framework to multi-sensor fusion scenarios could improve adaptability in real-world healthcare applications. The findings highlight the potential of deep reinforcement learning in WBAN anomaly detection, paving the way for more intelligent and adaptive health monitoring systems.