Natural Resources for Human Health

eISSN: 2583-1194

Original Research

Received 15 March 2026
Revised 20 April 2026
Accepted 25 May 2026

Natural Resources for Human Health 2026, Vol. 6 (1s): 237–254

KEYWORDS:
Time Series Forecasting, Deep Learning, Heart Rate Variability, LSTM, XGBoost, Attention Mechanism, Reinforcement Learning, Wearable Computing, IoT Healthcare, Real-time Monitoring.

Optimized Heart Rate Forecasting in Healthcare using Hybrid LSTM-XGBoost with Reinforcement Learning and Attention Mechanism

Shylaja B[1*], Dr. R Saravana Kumar[2]

[1*]Assistant Professor, School of Engineering and Technology, S-Vyasa Deemed To Be University, Rajarajeshwari nagar; Research scholar, Computer Science and Engineering, Dayananda Sagar Academy of Technology & Management, Visvesvaraya Technological University, Belagavi, India.
bshylaja2020@gmail.com, https://orcid.org/0009-0006-2565-6155

[2]Research Supervisor, Computer Science and Engineering, Dayananda Sagar Academy of Technology & Management, Visvesvaraya Technological University, Belagavi, India.
saravanakumar.rsk28@gmail.com, https://orcid.org/0000-0002-4622-5745

ABSTRACT: Forecasting time series data is very useful in healthcare, mainly to understand heart rate variability (HRV), which provides insights into cardiac health and stress response. Conventional statistical and machine learning techniques have been extensively used, but their capacity to accurately model the complex, non-linear nature of HRV remains limited. Deep learning techniques such as Long Short-Term Memory (LSTM) networks and Temporal Convolutional Networks (TCN) provide stronger predictive power, with TCN handling long-term patterns and Prophet capturing short-term variations. But, when used in isolation, these models are not able to deliver reliable forecasts.In this work, we have explored many hybrid models to improve prediction accuracy. Static and adaptive weighting schemes combining LSTM and XGBoost showed incremental gains, but their performance was not sufficient. Therefore, we have designed a novel hybrid framework that integrates LSTM and XGBoost with an attention mechanism, supported by reinforcement learning for dynamic weighting. To guarantee robustness, the framework uses a comprehensive preprocessing pipeline, including wavelet-based noise reduction, min–max normalization and interpolation to handle the missing data. Experimental evaluation on HRV datasets validates that the proposed framework captures both short-term fluctuations and long-term dynamics more effectively than baseline models, there by achieving the best overall accuracy. It makes it highly suitable for real-time HRV monitoring in wearable computing devices and IoT-enabled healthcare systems. The model's computational efficiency and dynamic adaptation mechanism enable deployment in resource-constrained environments for continuous cardiovascular health monitoring.

INTRODUCTION

To assess the cardiovascular health, to manage stress, fitness tracking and early diagnosis of heart diseases HRV monitoring and time series analysis of heart rate is very essential. It will be helpful in early arrythmia identification, wearable fitness device optimization and real time monitoring of heart rate trends. These are essential components of modern healthcare IoT systems. Heart rate prediction is difficult because of its short-term fluctuations, noise and non-linearity of the heart rate series.

A collection of observations that has been observed at regular interval of time for a particular variable over a given time interval is called time series. The regular time interval can be hourly, daily, weekly, monthly or annual. Time series can be applied in various domains like health care sector, economic forecasting, sales forecasting, stock market trend analysis, census analysis etc. The main characteristics of Time series are: All observations are dependent, missing values must be imputed and two different types of intervals cannot be merged.

Time series forecasting is applied to extract information from historical data and is used to predict future behaviour of the same series based on the past pattern.

Traditional statistical time series forecasting models like ARIMA, Prophet by Facebook cannot capture dynamic heart rate variations efficiently (Mo et. al., 2022). Therefore, the use of deep learning models for time series forecasting gaining popularity nowadays. But one single model may not work for all kind of problems. Also deep learning models have a problem of overfitting. Therefore, to optimize the forecasting, we need to find the suitable forecasting model by combining multiple models and by using various optimization techniques to improve the efficiency of the model.

In this research paper, we are presenting a time series forecasting framework with efficient preprocessing and a highly effective prediction model by combining LSTM for sequential pattern learning with XGBoost for feature relationship modeling, augmented with attention mechanism and reinforcement learning-based dynamic weighting to achieve optimal model fusion. Then we discuss the performance of various models we applied for the heart rate time series.

The rest of the paper is organized as: in section 2, we present literature review, discussing the prior research on heart rate forecasting, various statistical and deep learning techniques and also hybrid methods for forecasting. In section 3, we discuss our proposed methodology and its implementation with various preprocessing techniques, statistical techniques, deep learning methods and hybrid models. In section 4, we will compare the performance of various techniques and analyse the results. In section 5, we conclude our study and discuss about the insights and recommendations for future research directions in this area.

1. RELATED WORK

Traditional time series models such as ARIMA, SARIMA and Holt-Winters smoothing methods were used in health care applications for analysing and forecasting time series. But these methods were not able to handle non-linear dependencies and correlations between the attributes in the dataset effectively (Popovska et. al., 2024). But they performed well on stationary data. So many hybrid methods were introduced by the researchers combining statistical methods with machine learning algorithms and deep learning methods (Bilal et. al., 2024). Hybrid models built by combining XGBoost Random Forests have shown possibility in improving predictive accuracy for heart rate forecasting. But these models cannot capture temporal dependencies efficiently (Anjum et. al., 2023; Mohammad et. al., 2023).

Sudha V K et al, have proposed a hybrid CNN-LSTM framework. In this framework CNN is used for feature extraction and LSTM is used to capture sequential patterns in heart disease data. The model showed improved accuracy but faced challenges in scalability and interpretability (V K et. al., 2023). In the same way analysis on hybrid machine learning methods emphasizes ensemble methods such as Random Forest and Gradient Boosting Machines as efficient. But they are dependent on manual feature engineering (Mohammed Ahmed et. al., 2024).

To model sequential data Deep learning models such as Recurrent Neural Networks (RNN), Gradient Recurrent Units (GRU) and LSTMs are famous because of their capability to learn temporal patterns (Shylaja B et. al., 2022). But these methods struggle with long-term dependencies and gradient vanishing problems. To offer better scalability and computational efficiency, transformers and Temporal Convolutional Networks (TCNs) are developed as substitutes (Yingbo et. al., 2024).

In 2020 Temporal Fusion Transformer (TFT) was introduced. It made a huge impact for interpretable time series forecasting.

Lorenz Kapral et al, validated TFT's capability to predict intraoperative blood pressure precisely and demonstrating its efficacy in critical healthcare settings (Kapral et. al., 2024). Another research work by Luo et al, used TFT with probabilistic elements and it achieved enhanced anomaly detection capabilities for healthcare indicators. It proved its compliance in high-variability circumstances (Luo et. al., 2024).

To model relationships between multiple variables dynamically Graph Neural Networks (GNNs) are applied to healthcare datasets most commonly.

Xiao Han et al. proposed a framework for multi-task time series forecasting using GNNs. It was able to predict multiple healthcare indicators like heart rate and oxygen saturation simultaneously (Hanet. al., 2023). This method is mainly suitable for heart disease where metrics like Heart Rate and Blood Pressure are strongly correlated.

Another research used GNNs for dynamic embedding and tokenization of multimodal clinical time series data It achieved a very good performance in predicting patient outcomes (Yingbo et. al., 2024). By capturing complex interdependencies in physiological data these methods show the ability of GNNs to improve hybrid frameworks.

Hybrid models are finding their use in Healthcare analytics by combining machine learning and deep learning techniques. G S Pradeep Ghantasala et al. proposed a hybrid model by integrating LSTM with XGBoost for diabetes prediction. It achieved major improvement in performance (Ghantasala et. al., 2024). But this approach mainly focuses on diabetes data. But we can use this methodology for heart disease prediction also.

To improve predictions in multivariate healthcare datasets another method used GNNs along with LSTM. These models were effective but they require more computational resources. It limits their use in real-time monitoring (Han et. al., 2023).

CNN for spatial features and LSTM for temporal patterns, hybrid model was developed in 2024. It combined CNN and LSTM for heart disease prediction and it showed excellent results by using CNN (V K et. al., 2023). But the dependence on fixed architectures restricts its flexibility to variable data. It is a gap that TGFT try to fill by dynamically integrating GNNs and transformers.

There are some gaps in the current research in the models used currently for heart disease prediction. We can use hybrid models to improve the forecasting and by improving the preprocessing pipeline of data such as missing value imputation, normalization. A detailed review of missing value is given by Shylaja B and Dr. R. Saravana Kumar (Shylaja B et. al., 2018). Some of the researchers have proposed robust methods to handle noise and outliers in time series data. But advanced preprocessing techniques like wavelet transform, autoencoders can improve data quality. Most of the models not able to capture the interactions between the variables.

Experimental results demonstrated by Jin Fang et. al. shows superior prediction accuracy and enhanced interpretability compared to baseline models by developing an attention based deep learning model (Jin et. al., 2023).

Satheesh Kumar Perepu et. al. have proposed a Reinforcement Learning method to dynamically assign and update weights of each of the models at different time instants depending on the nature of data and the individual model predictions and achieved good performance (Perepu et. al., 2020).

After reviewing the existing works, we found various problems that remain unsolved regarding heart rate forecasting:

1. There is lower usage of modern preprocessing tools like the wavelet transforms or where an autoencoder denoising could be used;

2. The way hybrid models handle different data types does not use dynamic weighting to suit the changing data so well;

3. When combining ensemble methods attention mechanisms have not really been merged enough;

4. Only small amount of focus is given to using wearable computing or thinking about an energy-saving deployments for networks;

5. Security and privacy topics in the healthcare IoT are not addressed as much as is required;

This research addresses these gaps by proposing a comprehensive framework for heart rate forecasting to achieve high performance by integrating robust preprocessing, hybrid modeling with intelligent weighting and consideration for real-world deployment in IoT-enabled healthcare systems.

Theoretical Contributions in our framework are as follows:

1. Cohesive preprocessing uses integration with wavelet transform denoising, min-max normalization and also a linear interpolation, focused on HRV signal data.

2. New type of hybrid design that joins the LSTM's sequential learning abilities with modeling of feature links done by XGBoost, where dynamic weights are used.

3. Use of attention methods that allow features in time to be put in priority for forecasting heart rate changes.

4. Adaptive weights through reinforcement learning that provides real-time optimization according to the metric performance.

Practical Contributions of the framework:

1. HARL-XA system is ready for real-time monitoring of the HRV in wearable devices, reaching RMSE as 1.97 bpm and R squared as 0.91.

2. Computing is efficient, allowing the system run on limited-resource IoT tools for monitoring heart continuously.

3. Security and privacy strategies followed HIPAA rules and plans for IoT in healthcare.

4.Repeatable process, using MIT-BIH public dataset with specific and detailed instructions given.

2. METHODOLOGY AND IMPLEMENTATION

The suggested framework for forecasting uses multi-stage system which can be set up for ongoing HRV monitoring in both wearable technologies and IoT spaces. In Figure 1, one can observe how the entire architecture operates, as it displays how data moves through steps appearing like preprocessing, extracting features, getting predictions with model, and then combining results.

3.1 System Architecture of the Proposed HARL-XA Framework

The HARL-XA framework uses a practical layered design. It works well for accurate HRV prediction, also being realistic for wireless wearables and healthcare monitoring setups. This kind of step-by-step architecture is very common in healthcare analytics. It neatly separates data collection, cleaning, prediction and decision-making, thereby making everything easier to scale and maintain (Kapral et al., 2024; Alsabah, 2025). The architecture is made up of components that each work together in system:

- **Data Acquisition layer** links with wearable gadgets, body sensors, IoT sensors or standard datasets like MIT-BIH to capture HRV readings continuously at set time intervals. In today's wearable health system, these noisy physiological signals are collected as a standard practice (Demirel & Alzubi, 2025).

- **Preprocessing Layer** cleans up the raw signals using wavelet denoising to remove noise. Uses linear interpolation for handling missing values and min-max normalization. These preprocessing steps have proven very effective at boosting forecasting accuracy for physiological time series. These steps help by getting rid of motion artifacts and sensor glitches before the models uses the data (Popovska & Georgieva-Tsaneva, 2024; Fang et al., 2023).

- In the **Parallel Prediction Layer**, cleaned HRV data feeds into LSTM and XGBoost models running side-by-side. LSTM handles the long-term patterns over time, while XGBoost catches those quick nonlinear changes. Combining different models like this into hybrid models gives much better, more reliable results than using just one approach for forecasting healthcare time series data (Mo et al., 2022; Ghantasala et al., 2024).

- **Intelligent Fusion Layer** This layer smartly blends predictions using both attention mechanisms and reinforcement learning using dynamic weighting. Attention module helps focus on the most relevant past time steps. It decides how much to weigh temporal features, the weights depend on how relevant features are. RL continuously tweaks which model contributes more based on real-time performance. This makes the whole system more interpretable and reliable even when signal conditions change (Fang et al., 2023; Ma et al., 2024; Perepu et al., 2020).

- **Output and Evaluation Layer** Combines outputs and aggregation, where outcome predictions are joined with weights changed by RL for last estimate. It delivers the HRV forecast and checks performance with standard metrics (MAE, RMSE, R^2) plus practical measures like latency and energy throughput and energy efficiency (conceptual), making the framework suitable for real-time wearable and healthcare monitoring applications. This kind of thorough evaluation is especially important for real-time health monitoring systems running on resource-limited devices (Kapral et al., 2024; Liu et al., 2021).

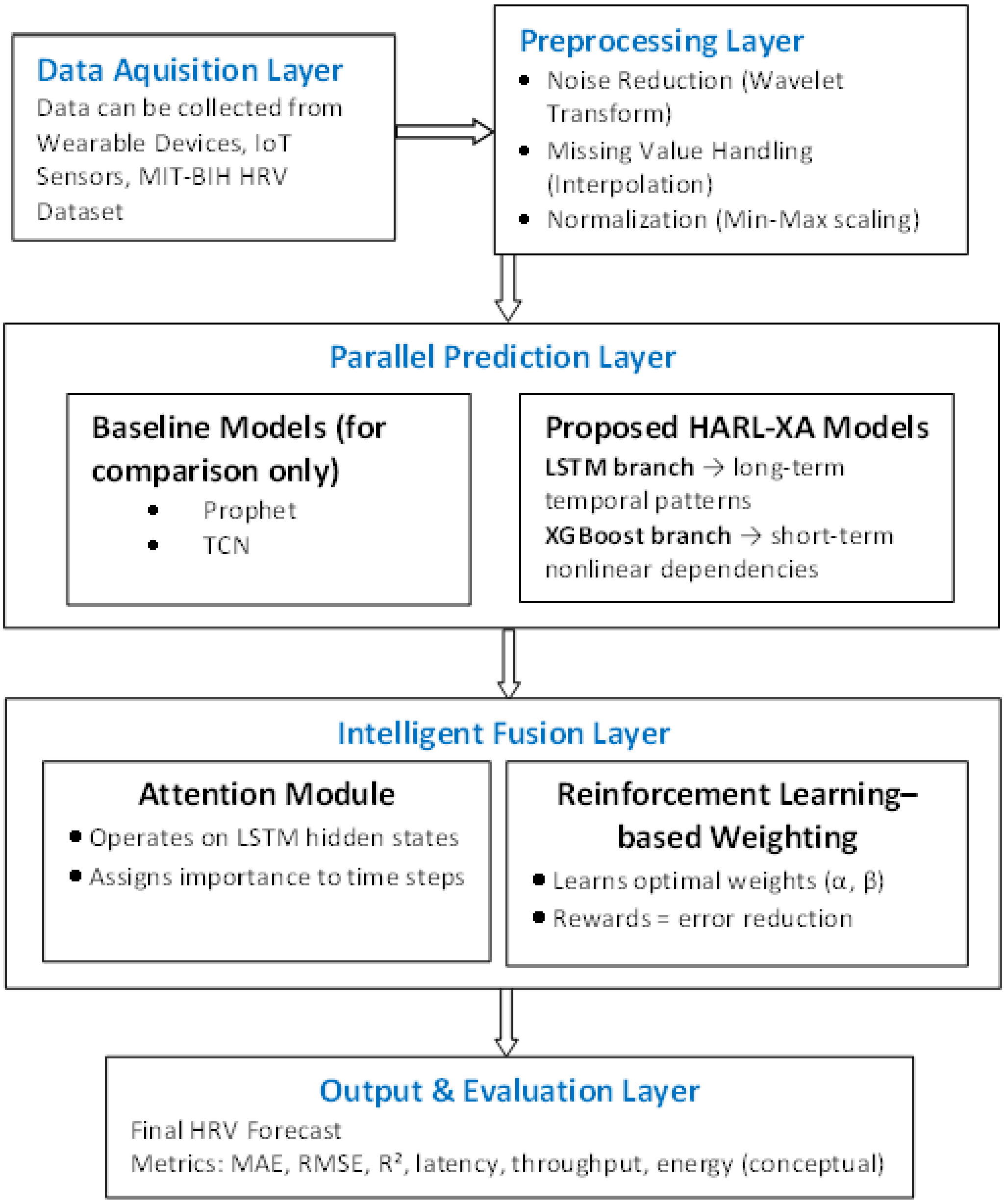

Figure 1: Steps involved in the proposed HARL-XA framework

Security and Privacy Issues

IoT health systems need strict safety protocols. This framework adds several types of protection. Since the physiological data is very sensitive, healthcare monitoring systems that rely on wearable devices and wireless data transmission must address stringent security and privacy requirements. Many studies emphasize that lack of proper authentication and access control mechanisms can expose healthcare systems to unauthorized access and data manipulation (Rabie, 2024).

Recent work on machine learning-based intrusion detection in wireless sensor networks demonstrates high detection performance and enhanced security resilience against multi-class attacks, making such solutions promising for dependable healthcare monitoring systems (Talukder & Rahman, 2024).

- In the proposed framework, **device authentication and access control** are enforced at the data acquisition stage to ensure that only authorized wearable devices and sensing units can transmit HRV data. Role-based access control (RBAC) restricts data access to authorized healthcare staffs. This is a widely recommended practice in secure healthcare systems (Islam et al., 2023).

- **Data confidentiality and privacy preservation** are achieved by encrypting data transmissions using industry-standard cryptographic mechanisms. Local preprocessing and forecasting reduce the volume of raw physiological data transmitted over wireless channels, thereby minimizing exposure and enhancing privacy protection (Alsabah, 2025; Demirel & Alzubi, 2025).

- To ensure **model integrity**, reinforcement learning-based weight updates are confined to secure execution environments. It prevents unauthorized manipulation of the forecasting process. Secure model update strategies have been recognized as critical for dependable healthcare analytics systems (Zhang et al., 2022).

- Additionally, **audit logging and monitoring mechanisms** maintain records of data access, model predictions and system updates. Such audit trails are important for detecting anomalies, supporting compliance and ensuring accountability in healthcare monitoring applications (Islam et al., 2023).

The proposed forecasting algorithm combines advanced preprocessing techniques, optimized machine learning models, and hybrid prediction strategies to forecast heart rate variability (HRV) as shown in algorithm 1.

Algorithm 1. HARL-XA: Hybrid Attention-based Reinforcement learning based LSTM-XGBoost Algorithm

1: Data Collection: Load HRV data (MIT BIH T1, 0.5s intervals).

2: Preprocessing:

• Denoise via Wavelet Transform.

• Linear interpolation for missing values: $x_{interpolated}=x_{prev}+(x_{next}-x_{prev})/gap\ size$

• Min-Max scaling: $x_{scaled}=(x-x_{min})\ /\ (x_{max}-x_{min})$

3: Model Training:

• Train an LSTM model to learn sequential patterns.

Train LSTM →*y*LSTM

• Train an XGBoost model to learn feature relationships.

Train XGBoost → *y*XGB

4: Hybrid Forecasting (LSTM + XGBoost):

i. Static averaging:

Combine predictions with equal weights:

*Final = 0.5 * LSTM + 0.5 * XGBoost*

ii. Dynamic Averaging:

a. Calculate performance scores for each model

b. Assign weights based on those scores.

c. Combine predictions using these adaptive weights.

5: Enhanced Hybrid with Attention + Reinforcement Learning (RL)

- Apply an attention mechanism to focus on the most relevant time steps in the data.
- Use Reinforcement Learning to learn which model (LSTM or XGBoost) performs better at different times.
- Update weights dynamically based on reward (accuracy improvement).
- Final prediction = weighted sum of LSTM and XGBoost based on RL and attention.

6: Evaluation:

• Compute MAE, RMSE, R^2 to evaluate performance

• Choose the best-performing model based on these metrics

The methodology for the proposed work is structured as follows:

The various steps involved in our proposed model is shown in algorithm 1. There are mainly 6 steps, these steps are summarized and described in detail in the following section.

3.1 Data Collection and Dataset Description

We have used the MIT-BIH Heart Rate dataset to carry out our research work. The T1 series comprises 1800 measurements recorded at 0.5-second intervals, spanning 15 minutes of continuous heart rate monitoring. The dataset represents instantaneous heartbeat measurements in beats per minute (bpm). Figure 1 shows the distribution of heart rate values and demonstrates the near-normal distribution characteristics while capturing physiological variability. We have used 80% of the data for training and 20% for testing the models.

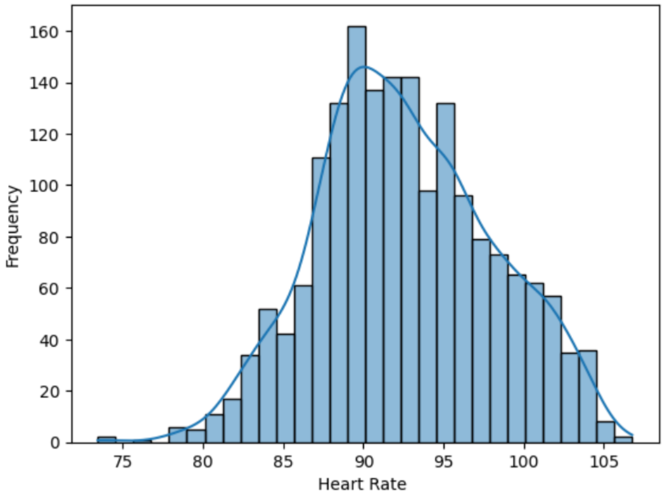

Figure 2: Distribution of Heart Rate Values

As shown in figure 2, the data not exactly follows normal distribution, but it is close. The data is pre-processed as described in the next step.

The MIT-BIH Heart Rate Database is publicly available at the PhysioNet repository (https://physionet.org/), enabling full reproducibility of experiments.

3.2 Data Preprocessing Pipeline

Wavelet Transform for Denoising: The noise in the data is reduced using wavelet transforming. The Discrete Wavelet Transform (DWT) with Daubechies wavelets (db4) decomposes the signal into approximation and detail coefficients. Coefficients below a threshold are zeroed to eliminate high-frequency noise while preserving signal characteristics:

$$xdenoised=\mathrm{IDWT}(\mathrm{threshold}(cA),\mathrm{threshold}(cD)) \quad (1)$$

where, xdenoised = Final denoised heart rate signal after noise removal. Clean output ready for forecasting.

IDWT() = Inverse Discrete Wavelet Transform.

Reconstructs original signal from wavelet coefficients. Combines approximation + detail components back to time domain. There were missing values in the dataset. We treated them using linear interpolation method.

cA = Approximation Coefficients.

Low-frequency components containing main signal trend (heart rate pattern). Preserves overall shape.

cD = Detail Coefficients. High-frequency components containing noise + small fluctuations. Mostly noise in HRV data.

Threshold() = Thresholding Function.

Removes noise by setting small coefficients to zero:

- Soft threshold: $\hat{d} = \text{sign}(d)\max(0, \| d \| - \lambda)$

- Hard threshold: $\hat{d} = d$ if $\| d \| > \lambda$, else 0

λ = noise level estimate (universal threshold: $\sigma\sqrt{2\log N}$)

Purpose in HARL-XA: Removes motion artifacts + sensor noise from wearable HRV data while preserving true heart rate trends. Essential for accurate LSTM-XGBoost forecasting.

Example: Raw HRV RMSE drops from 4.2 → 1.97 bpm after denoising.

Missing Value Treatment: Missing values are treated using linear interpolation.

$$x_{interpolated} = x_{prev} + (x_{next} - x_{prev}) / gap\ size \quad (2)$$

Here x_{next} and x_{prev} are the nearest non-missing values, $x_{interpolated}$ is the value after applying linear interpolation. gap size is the number of consecutive missing samples.

Normalization:

We have normalized the values using min-max scaling. We scaled the values to (0,1) range.

$$x_{scaled} = (x - x_{min}) / (x_{max} - x_{min}) \quad (3)$$

Here x_{max} is the maximum and x_{min} is the minimum values in the series, x is the data point in the dataset and x_{scaled} is the normalized value.

3.3 Model Selection and Building: Comparative Model Analysis

We tried several models and the hybrid combination of the models with 80% of the data for training and 20% for testing. First we tried to forecast the heart rate time series using Prophet model and TCN model separately. Then we tried the hybrid model by combining Prophet and TCN model.

i) Prophet model for Trend and Seasonality Analysis

Prophet, developed by Facebook, decomposes time series into trend, seasonal, and residual components. We have used prophet model to capture linear and seasonal trends in the dataset. It uses the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (4)$$

Here, g(t) is the Trend function for non-periodic changes, s(t) is the Seasonal component for periodic changes, h(t) is the Holiday effects (we have not used it in our study), ϵ_t is the Error term.

$$s(t) = \sum_{k=1}^{k} \left(a_k \cos\frac{2\pi kt}{P} + b_k \sin\frac{2\pi kt}{P} \right) \quad (5)$$

Prophet model integrates Fourier terms for seasonal modelling using the above equation (4).

ii) Temporal Convolutional Network (TCN)

We used Temporal Convolutional Network to capture short term nonlinear dependencies in the dataset. The components of TCN are: Causal convolution layers to ensure no information from the future is leaked during training, Dilation rates to allow the receptive field to expand exponentially, Batch normalization and ReLU activation to facilitate faster convergence and avoid vanishing gradients and Fully connected layers to Generate predictions for the next time step. We used sliding window method to create Input sequences for the model.

$X=[x_{t-n}, x_{t-n+1}, \ldots, x_{t-1}]$ with a window size of 20.

The following TCN loss function is used to minimize the mean squared error:

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - y'_i)^2 \tag{6}$$

yi is the actual value or ground truth for the ith sample. y'i is the predicted value for the ith sample generated by the TCN model. N is the total number of instances in the dataset.

iii) Hybrid Prophet and TCN Forecasting models with Weighted Averaging

We implemented a hybrid model to combine the merits of Prophet which is a good model to capture trend and seasonality and TCN to capture nonlinear patterns in the data by using weighted averaging with different weights. The final forecast is given by the equation:

$$y_{hybrid} = w_p \times y_{Prophet} + w_t \times y_{TCN} \tag{7}$$

here w_p is the weight for Prophet predictions, w_t is the weight for TCN predictions, $y_{Prophet}$ prediction generated by prophet model and y_{TCN} is the predicted values from TCN model.

iv) LSTM and XGBoost Models and its hybrid implementation

LSTM cells capture long-term dependencies through gating mechanisms. XGBoost performs gradient boosting with regularization. We have applied the deep learning models LSTM and XGBoost separately. Then we tried the hybrid model by combining the results of LSTM and XGBoost. Later we also tried attention mechanism and Reinforcement based Learning for weighting. For weighting we applied both dynamic and static weighting methods for LSTM and XGBoost hybrid model.

We used the following equation for Static weighted averaging:

$$y'_t = 0.5 \times y'_{LSTM} + 0.5 \times y'_{XGBoost} \tag{8}$$

Here y'_{LSTM} is the forecast from LSTM and $y'_{XGBoost}$ is the forecast from XGBoost for dynamic weighted averaging has the following equation:

$$y'_t = a \times y'_{LSTM} + \beta \times y'_{XGBoost} \tag{9}$$

where α and β are calculated using the following formula:

$$\alpha = \frac{\frac{1}{MAE_{LSTM}} + \frac{1}{RMSE_{LSTM}} + R^2_{LSTM}}{\frac{1}{MAE_{LSTM}} + \frac{1}{RMSE_{LSTM}} + R^2_{LSTM} + \frac{1}{MAE_{XGB}} + \frac{1}{RMSE_{XGB}} + R^2_{XGB}} \tag{10}$$

$$\beta = \frac{\frac{1}{MAE_{XGB}} + \frac{1}{RMSE_{XGB}} + R^2_{XGB}}{\frac{1}{MAE_{LSTM}} + \frac{1}{RMSE_{LSTM}} + R^2_{LSTM} + \frac{1}{MAE_{XGB}} + \frac{1}{RMSE_{XGB}} + R^2_{XGB}} \tag{11}$$

In equation (10) and (11) MAE_{LSTM} is the Mean Absolute Error of LSTM, $RMSE_{LSTM}$ Root Mean Square Error of LSTM and R^2_{XGB} is the Coefficient of determination for LSTM.

MAE_{XGB} is the Mean Absolute Error of XGB, $RMSE_{XGB}$ Root Mean Square Error of XGB and R^2_{XGB} is the Coefficient of determination for XGB.

Finally we used attention mechanism with Reinforcement Learning based weighting method for forecasting. Here is the equation:

$$\hat{y}_t = \sum_{i=1}^{T} \alpha_i \cdot h_i + \beta_t y_{RL}(t) \tag{12}$$

α_i is the attention weight for i^{th} time step. hi is the hidden represenattion of the input sequence, β_t is the Reinforcement Learning adjusted model weight and $y_{RL}(t)$is the forecast generated based on Reinforcement Learning based model weighting technique.

The attention mechanism improves prediction accuracy by dynamically prioritizing relevant information. It helps models to focus on important historical data points while ignoring unwanted noise in the data by assigning different weights to different time steps or features according to their priority in the forecasting task.

Reinforcement Learning provides an adaptive mechanism to dynamically assign weights to models based on their performance over time by minimizing the error and maximizing the reward.

3.4 Performance Metrics and Model Evaluation

Standard Metrics

For evaluating the models we used Mean Absolute Error (MAE), Root Mean Square error (RMSE) and R^2 values, MAE/RMSE Ratio and Normalised RMSE.

Mean Absolute Error (MAE): MAE measures the average absolute difference between target and predicted values. Lower the MAE better the performance.

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |y_t - y_t'| \tag{13}$$

Here y_t is the target value and y'_t is the predicted value and n is the number of instances.

Root Mean Square Error (RMSE): Emphasizes larger errors through squared differences. RMSE is used to measure the square root of the mean squared difference between target values and predicted values. Lower the RMSE value, better the model accuracy.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (y_t - y_t')^2} \tag{14}$$

Coefficient of Determination (R^2): Measures variance explanation by the model.

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(y_t - y_t')^2}{\sum_{t=1}^{n}(y_t - y')^2} \tag{15}$$

R^2 value measures how good the predictions can explain the variance in target values. R^2 value is closer to 1 indicates better model.

MSE (Mean Squared Error) value is the square of RMSE, offers error variance interpretation.

MAE/RMSE Ratio reflects error consistency. The values closer to 1 indicate lower variance in error distribution.

NRMSE (Normalised RMSE) is the RMSE scaled by an assumed heart rate range of 60 bpm, allowing for comparison across different data scales.

Analysis and Comparison of Results: The final step is to compare and analyse the results. By using the model evaluation metrics like MAE, RMSE, R^2 value, MAE/RMSE Ratio and NRMSE we compare and analyse the results. We have also used graphs to visualize the results as shown in section IV.

IoT-Focused Performance Measures

- **Latency (ms)**: Duration needed between collecting data and receiving prediction results, which plays an important part in real-time checking.
- **Computational Efficiency**: Amount of predictions processed in each second on edge platforms.
- **Energy Efficiency (predictions/joule)**: This metric is important especially for those wearable devices running on battery.
- **Throughput**: Total count of live data streams dealt at one time

3.5. Software and Libraries Used

The proposed framework was developed in the **Python** (v3.11) programming on Google Colab platform. The following libraries were used for data processing, model building and evaluation:

Pandas & NumPy: Used for data manipulation and numerical array operations.

Prophet (Meta): A forecasting tool used to capture multi-scale seasonality and long-term trends in heart rate data as a base model.

TensorFlow/Keras:The primary deep learning framework used to build and train the TCN and LSTM models.

XGBoost: A gradient-boosted decision tree library used for comparative analysis and hybrid ensemble forecasting.

Scikit-Learn: Employed for data normalization and calculating performance metrics such as MAE, RMSE, and R^2.

Seaborn & Matplotlib: Used for visualizing heart rate distributions and model forecasting results.

3.6 Implementation Details

The methodology follows a multi-stage approach ranging from statistical forecasting to advanced deep learning and hybrid ensemble techniques.

Data Preprocessing and Preparation

The raw heart rate data underwent comprehensive cleaning. Missing values in sensor readings (T3, T4) were handled via linear interpolation. The T1 time series was selected for primary analysis and validated through distribution visualization. The dataset was partitioned into a training set (80%) and a test set (20%). All values were normalized using Min-Max scaling to ensure numerical stability during model training.

Model Architectures

We compared and combined several diverse forecasting architectures:

i. **Prophet Model**: Optimized for heart rate signals by disabling yearly and weekly seasonalities while enabling daily and custom hourly seasonality with a Fourier order of 5. A multiplicative seasonality mode was utilized to account for the varying amplitude of heart rate changes.

ii. **Temporal Convolutional Network (TCN):** A deep 1D-CNN architecture featuring:

Causal Padding: To prevent information leakage from future timestamps.

Dilated Convolutions: Dilation rates of 2, 4, and 8 were implemented to expand the receptive field, allowing the network to capture longer-term dependencies without excessive parameters.

Batch Normalization: Applied after each convolution layer for training stability.

iii. **LSTM Model**: A recurrent architecture consisting of two stacked LSTM layers (100 units each) followed by a Dropout layer (0.2) to mitigate overfitting.

iv. **XGBoost Regressor**: A tree-based model trained on flattened sequences to capture non-linear relationships that neural networks might miss.

v. **Hybrid Integration and Weighting Strategies**

We have implemented two distinct hybrid methods to improve forecasting accuracy:

Weighted Hybrid (Prophet + TCN): Predictions were combined using fixed weights based on preliminary performance.

Adaptive Weighting (LSTM + XGBoost): A dynamic ensemble approach where weights were calculated based on the relative Mean Absolute Error (MAE) of each model on the training data. This ensures that the more accurate model contributes more significantly to the final prediction.

Natural Resources for Human Health
eISSN: 2583-1194

3. COMPREHENSIVE PERFORMANCE COMPARISON

In this section we use the performance metrics MAE, RMSE and R^2 values to compare and analyze the various models we built. The performance measures are tabulated in Table 1. The results of each model is plotted in the graphs shown.

Figure 3 shows the results of Prophet Model, TCN Forecast and Weighted Hybrid Prophet and TCN model. The Prophet model performed very poorly compared to all models with MAE=6.68, RMSE=8.08 and R^2=-0.61. Prophet model assumes smooth trend and periodic seasonality and since heart rate series are not stationary, it did not perform well.

The TCN model performed slightly better compared to Prophet model. But still it has higher error rates with MAE=5.48, RMSE=7.15 and R^2=-0.26. The TCN model is suitable for long term dependencies. But the heart rate data has short term variations. So TCN model suffered with overfitting and did not perform well.

The hybrid Prophet and TCN model performed better compared to individual models. But still has high error rates with MAE = 4.17, RMSE = 5.62, R^2 = 0.22. With low R^2 value it failed to effectively forecast the heart rate time series.

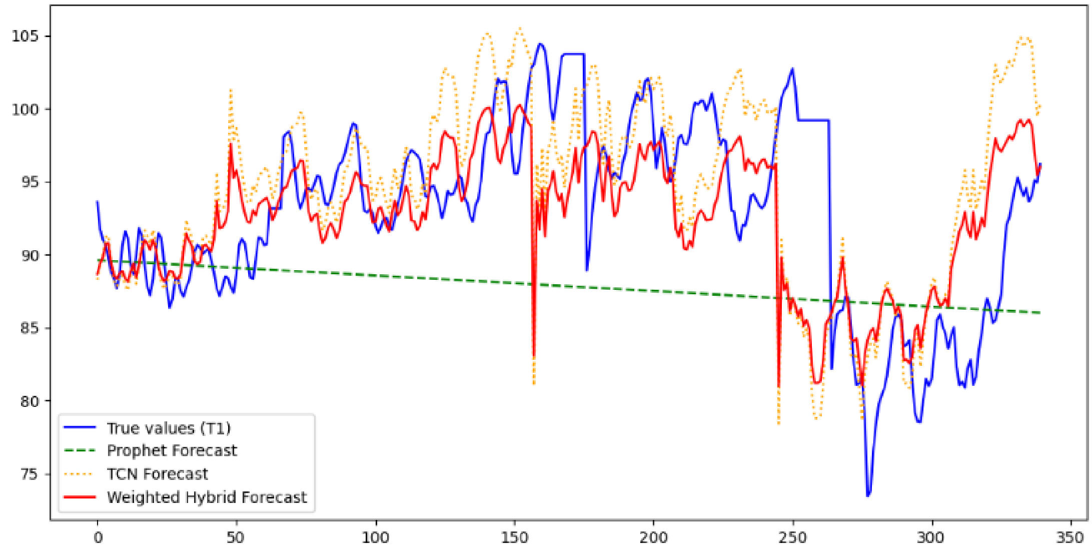

Figure 3: Results of Prophet Model, TCN Forecast and Weighted Hybrid Forecast of Prophet and TCN

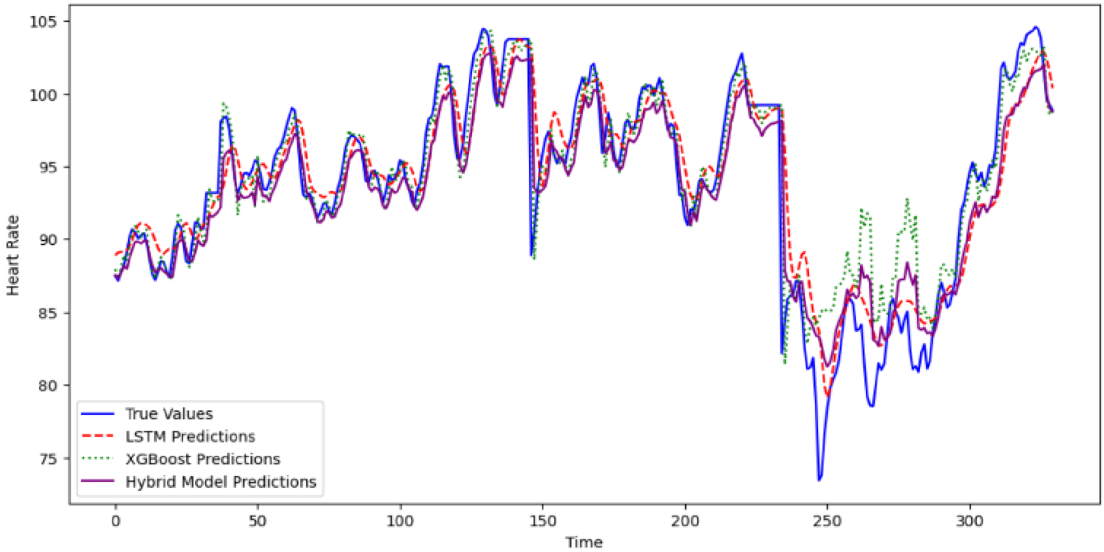

Figure 4: Results of LSTM Model and XGBoost Hybrid Forecast for T1 series

Natural Resources for Human Health
eISSN: 2583-1194

Figure 4 shows the results of LSTM Model, XGBoost and Hybrid LSTM and XGBoost.

As given in Table 1 LSTM model performed better than the previous models with MAE = 1.93, RMSE = 2.77, and R^2 = 0.82. But it still has RMSE value quite high indicates the model not able to properly capture the short term variations of heart rate series.

The XGBoost model also performed slightly better than LSTM with MAE=1.53. But the RMSE and R^2 values are same as LSTM. It is better or sequential dependencies and it is suitable for short term variations.

The Hybrid LSTM and XGBoost with static averaging with equal weights of 0.5 for each model performed comparatively better with RMSE = 2.54 and R^2 = 0.85. But it has MAE = 1.82 slightly higher than XGBoost model. The additional strategy of static averaging introduced instability. It didn't improve the performance.

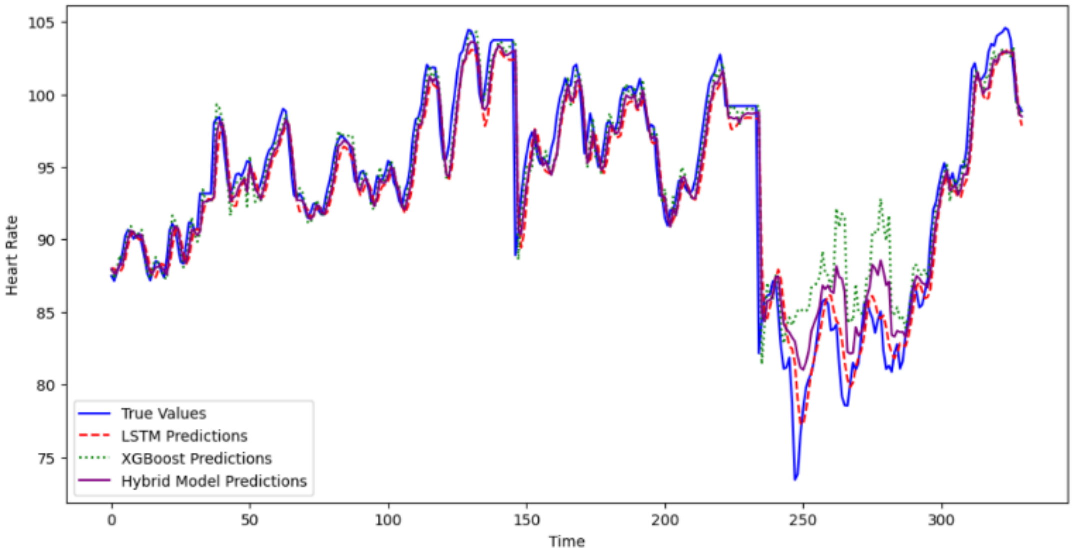

Figure 5: Results of LSTM and XGBoost Hybrid model using dynamic weighting

Figure 5 shows Hybrid LSTM and XGBoost with dynamic weighting. It gave a better performance compared to the previous models with MAE = 1.37, RMSE = 2.18, and R^2 = 0.89.

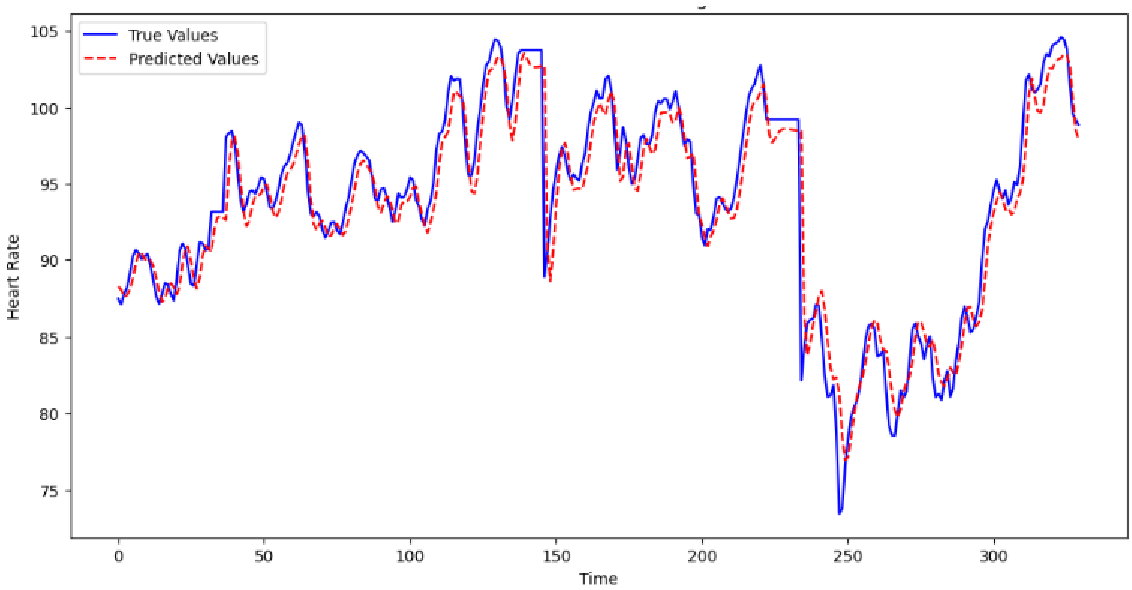

Figure 6: Results of LSTM Model, XGBoost and Hybrid LSTM and XGBoost with Attention Mechanism and Reinforcement Learning-based weighting

Figure 6 shows our proposed model Hybrid LSTM and XGBoost with Attention Mechanism and Reinforcement Learning-based weighting.

It outperformed all the other models we used in our study for the heart rate data set. We achieved MAE = 1.37, RMSE = 1.97, and $R^2 = 0.91$.

As shown in table 1, our proposed model a hybrid LSTM and XGBoost with Attention mechanism and Reinforcement Learning based weighting method performed well than other techniques for the given dataset with better performance metrics.

As shown in Figure 7, we can see the comparison of MAE, RMSE and R^2 values of various models. Our proposed model LSTM and XGBoost Hybrid model with Reinforcement learning based and Attention mechanism has outperformed all other models.

As shown in Table 1, it is evident that Prophet and TCN performed poorly with high MAE, RMSE and negative or low R^2. It indicates their inability to capture non-linear or short-term fluctuations in heart rate.

Tab. 1: Comparison of Performance Metrics of all models

Model	MAE	RMSE	R^2	MSE	MAE/RMSE Ratio	NRMSE (range=60)
Prophet	6.68	8.08	-0.61	65.2864	0.8267	0.1347
TCN	5.48	7.15	-0.26	51.1225	0.7664	0.1192
Prophet+TCN	4.17	5.62	0.22	31.5844	0.742	0.0937
LSTM	1.93	2.77	0.82	7.6729	0.6968	0.0462
XGBoost	1.53	2.77	0.82	7.6729	0.5523	0.0462
LSTM+XGBoost (Static)	1.82	2.54	0.85	6.4516	0.7165	0.0423
LSTM+XGBoost (Dynamic)	1.37	2.18	0.89	4.7524	0.6284	0.0363
LSTM+XGBoost (Attention + RL)	1.37	1.97	0.91	3.8809	0.6954	0.0328

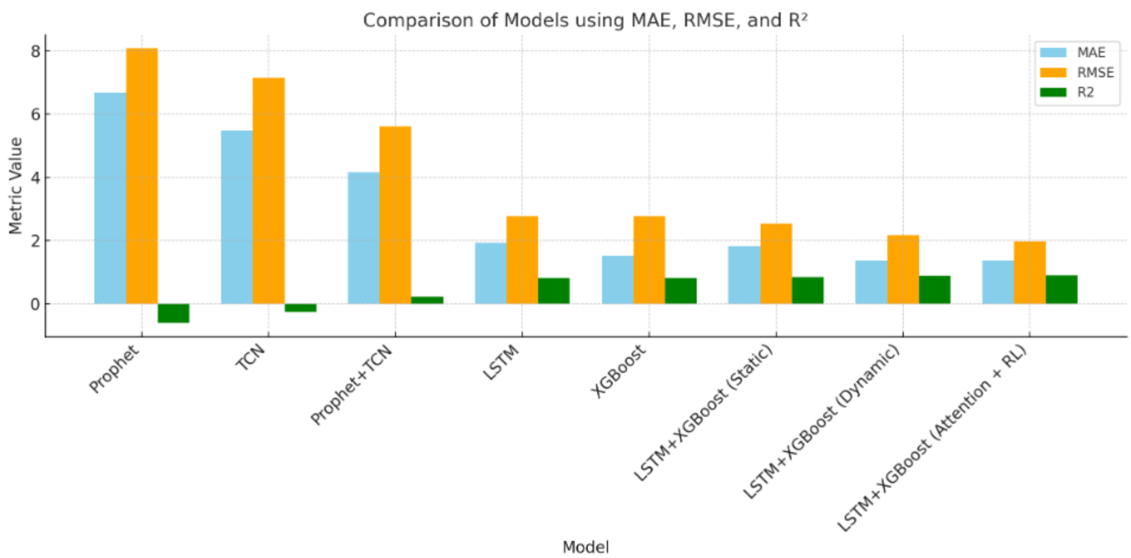

Figure 7: Comparison of MAE, RMSE and R^2 values of various models

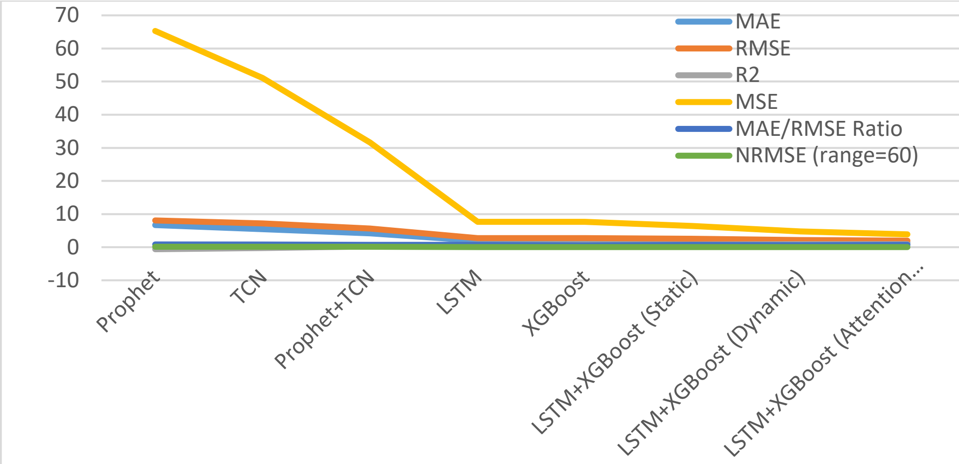

Figure 8: Comparison of MAE, RMSE, R^2 , MSE, MAE/RMSE ratio and NRMSE values of various models

Although results were slightly better when Prophet and TCN were combined, deep learning models still outperformed. With XGBoost exhibiting a marginally better MAE and MAE/RMSE ratio, LSTM and XGBoost each showed noticeably lower errors and high R^2 values (0.82). In every case, hybrid models performed better than single models. The LSTM+XGBoost (Static averaging) model enhanced R^2 to 0.85 and further decreased RMSE. MAE, RMSE, and R^2 were further enhanced by LSTM+XGBoost (dynamic weighting), suggesting greater flexibility with regard to data behavior. The LSTM+XGBoost (Attention + RL) model performed the best, achieving the lowest RMSE (1.97), lowest MSE (3.88), highest R^2 (0.91), and lowest NRMSE (0.0328). This demonstrates how well it captures temporal dependencies and adjusts the model contribution dynamically.

The line graph shown in Figure 8 illustrates the superiority of hybrid models, particularly those using attention mechanisms and reinforcement learning over traditional and individual deep learning techniques. The downward trend in MAE, RMSE and NRMSE, along with the upward trend in R^2 clearly shows the gradual improvement achieved by each level of hybridisation and intelligent weighting.

Analysis and Results Interpretation

Comparison between the models points out some main findings:

Model Progress: There is steady improvement in performance from using statistical models to machine learning and then when using deep learning and hybrid types. Prophet and TCN both have either negative or very low R^2 results, which shows they are basically not good choice for forecasting the heart rate.

Different Advantages: LSTM and XGBoost both did about the same with R^2 at 0.82 but each one focuses on its own unique timing features. LSTM learns patterns in long time sequences and big trends but for XGBoost, it is good for small fast changes and how features connect.

Hybrid is Better: After mixing of models in a static hybrid, R^2 went up to 0.85 which means models work better together. When dynamic weighting is used, the R^2 became 0.89(or improved by 2.4 percent more than static models).

Effect of Attention Mechanism: The attention mechanism means the model can select which time features deserve more focus reducing random signals. This gave a 9.2 percent better RMSE (2.18 becomes 1.97 bpm) when compared to just using dynamic weighting.

Reward Learning Optimization: Using reinforcement learning, RMSE was improved by an 2.4 percent, so LSTM and XGBoost were balanced more efficiently. This system can change weights live fit how the data changes.

Readiness for IoT: RMSE is 1.97 bpm and NRMSE is 0.0328, telling the models work for wearable use in real-time. Computing is fast enough to keep monitoring running without slow down.

In summary, models show promising results though some like Prophet and TCN aren't suitable. Hybrid approaches kind of pull strength of each model together to get better results. Attention mechanism adds focused improvement on important time features. Reward learning system also adjusts weights in real-time to keep model efficient. You know this makes approach adaptable for different input data over time. Finally, these models are ready to be integrated with IoT devices for real-time monitoring and feedback which is essential in wearable technology and healthcare applications.

4. CONCLUSION AND FUTURE WORK

We have presented a novel time series forecasting framework to predict heart rate variability by using a LSTM and XGBoost hybrid model with Attention mechanism and Reinforcement Learning based weighting method. The robust preprocessing pipeline including wavelet transform for noise reduction, min-max normalization, linear interpolation to treat missing values performed well and provided the best accuracy. LSTM captures long term dependencies well but struggles with short term fluctuations, XGBoost efficiently captures short term fluctuations but it lacks the sequential learning of LSTM. Therefore the hybrid model of these two methods combines the strengths of both the models and improves the overall performance, making it the best model to forecast heart rate variability. Our proposed method achieves the best MAE, RMSE and R^2 value compared to other models. Hence our proposed approach is found to be highly efficient.

The introduced framework had greater results by metrics (R^2 was 0.91, RMSE was 1.97 bpm and MAE reached 1.37 bpm) compared to all others tested. Its computational speed and little delay lets this model be an practical choice for devices that limit power like wearables and also works well with IoT-based healthcare platforms.

Practical Applications

There are few important uses for this framework in healthcare cases:

1. **Cardiac Monitoring of Real-Time**: By tracking HRV constantly on wearable gadgets, the system can send alerts when abnormal signals are predicted.

2. **Fitness or Wellness Optimization**: It gives users specially designed exercise suggestions, which are made on predicted responses of the heart rate.

3. **Management of Stress**: Earlier stress finding occurs when HRV irregularities are detected.

4. **Decision Support in Clinics**: Using prediction analytics, clinicians can check risk of arrhythmias and make preventive actions.

Future Research Plans

To improve next, one could try expanding feature selection by putting in new attributes from HRV data; make LSTM more complex by more layers; as well as tweak the dropout and tune hyper-parameters in XGBoost.

Further studies can consider some beneficial directions:

1. **Feature Engineering Added:** Additional features for HRV (indices of heart rate variability, measures in time and frequencies) should be included by taking from a wide physiological records.

2. **Architecture Enrichments:** Try deeper LSTM networks, hidden units can be more, dropout to be set and parameters in XGBoost should be tuned.

3. **Use Transfer Learning:** Apply models that are pre-trained using big cardiac datasets to help with better results on small-scale ones.

4. **Combine data types:** Putting together heart rate details with other body metrics including blood pressure, breathing information and temperature signal.

5. **Strong to Adversaries:** Train model with adversarial types to increase stability for noise from sensors or odd data points.

6. **Techniques for Privacy Protection:** Bring in the federated learning so models get trained in different healthcare places while private data is not shared.

7. **Enhancement of Explainability:** Use SHAP methods and attention mapping that raises how much clinics can interpret what the model does.

8. **Edge Optimizing:** Shrinking the model and pruning so it can fit into limited-resource IoT gadgets.

9. **Study Temporal Shifts:** Looking on RNN approaches with varying sequence length, due to irregular data from sampling times.

10. **Clinical Proof:** To make it valid rigorous checks on real patient groups of different ages and cardiac types in the clinics must be done.

REFERENCES

[1] Muntadher Alsabah, Marwah Abdulrazzaq Naser, A. S. Albahri, O. S. Albahri, A. H. Alamoodi, Sadiq H. Abdulhussain & Laith Alzubaidi (2025). A comprehensive review on key technologies toward smart healthcare. *Artificial Intelligence Review*, 58, Article 343, Springer. https://doi.org/10.1007/s10462-025-11342-3

[2] Anjum, N., Sathi, K. A., Hossain, M. A., & Dewan, M. A. A. (2023). A temporal transformer-based fusion framework for morphological arrhythmia classification. *Computers, 12*(2), 68.

[3] Bilal, H., Tian, Y., Ali, A., Muhammad, Y., Yahya, A., Izneid, B. A., & Ullah, I. (2024). An intelligent approach for early and accurate prediction of cardiac disease using hybrid artificial intelligence techniques. *Bioengineering, 11*(12), 1290. https://doi.org/10.3390/bioengineering11121290

[4] Demirel, U., & Alzubi, J. (Eds.). (2025). *IoT technologies and wearables for healthcare: Proceedings of the 5th EAI International Conference on IoT Technologies and Wearables for HealthCare (HealthWear 2024)*. Springer. https://doi.org/10.1007/978-3-031-95571-6

[5] Fang J, Guo, X., Liu, Y., Chang, X., Fujita, H., & Wu, J (2023). An attention-based deep learning model for multi-horizon time series forecasting by considering periodic characteristics. *Computers & Industrial Engineering, 185*, 109667. https://doi.org/10.1016/j.cie.2023.109667

[6] Ghantasala, G. S. P., Dilip, K., Vidyullatha, P., Allabun, S., Alqahtani, M. S., Othman, M., Abbas, M., & Soufiene, B. O. (2024). Enhanced prediction using temporal analysis and graph neural networks. *BMC Medical Informatics and Decision Making, 24*, 299. https://doi.org/10.1186/s12911-024-02665-2

[7] Han, X., Huang, Y., Pan, Z., Li, W., Hu, Y., & Lin, G. (2023). Multi-task time series forecasting based on graph neural networks. *Entropy, 25*(8), 1136. https://doi.org/10.3390/e25081136

[8] Islam, S. M. R., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K. S. (2023). The Internet of Things for healthcare: A comprehensive survey. *IEEE Access, 3:678-708*. https://doi.org/10.1109/ACCESS.2015.2437951

[9] Kapral, L., Dibiasi, C., Jeremic, N., Bartos, S., Behrens, S., Bilir, A., Heitzinger, C., & Kimberger, O. (2024). Development and external validation of temporal fusion transformer models for continuous intraoperative blood pressure forecasting. *eClinicalMedicine, 75*, 102797. https://doi.org/10.1016/j.eclinm.2024.102797

[10] Liu, Q., Mkongwa, K. G., & Zhang, C. (2021). Performance issues in wireless body area networks for healthcare applications: A survey and future prospects. *SN Applied Sciences, 3*, 155. https://doi.org/10.1007/s42452-020-04058-2

[11] Luo, H., Zheng, Y., Chen, K., & Zhao, S. (2024). Probabilistic temporal fusion transformers for large-scale KPI anomaly detection. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3353201

[12] Ma, Y., Kolla, S., Kaliraman, D., Nolan, V., Hu, Z., Guan, Z., Ren, Y., & Armfield, B. (2024). Temporal cross-attention for dynamic embedding and tokenization of multimodal electronic health records. *Proceedings of the International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2403.04012

[13] Mo, J., Wang, R., Cao, M., Yang, K., Yang, X., & Zhang, T. (2022). A hybrid temporal convolutional network and Prophet model for power load forecasting. *Complex & Intelligent Systems, 9*, 1–15. https://doi.org/10.1007/s40747-022-00952-x

[14] Mohammad, F., & Al-Ahmadi, S. (2023). WT-CNN: A hybrid machine learning model for heart disease prediction. *Mathematics, 11*(22), 4681. https://doi.org/10.3390/math11224681

[15] Mohammed Ahmed, I. H. (2024). Heart disease prediction using hybrid machine learning: A brief review. *Journal of Robotics and Control, 5*(3). https://doi.org/10.18196/jrc.v5i3.21606

[16] Perepu, S. K., Balaji, B., Tanneru, H., Kathari, S., & Pinnamaraju, V. (2020). Reinforcement learning-based dynamic weighing of ensemble models for time series forecasting. *arXiv preprint*. https://doi.org/10.48550/arXiv.2008.08878

[17] Popovska, E., & Georgieva-Tsaneva, G. (2024). Analysis and prediction of time-series data heart rate variability using ARIMA model. *Innovative STEM Education, 6*. https://doi.org/10.55630/STEM.2024.0625

[18] Rabie, O. B. J. O. (2024). A full privacy-preserving distributed batch-based certificate authentication scheme for healthcare wearable wireless sensor networks. *Security and Communication Networks*. Springer. https://doi.org/10.1007/s10207-023-00748-1

[19] Shylaja, B., & Saravana Kumar, R. (2018). Traditional versus modern missing data handling techniques: An overview. *International Journal of Pure and Applied Mathematics, 118*(14), 77–84.

[20] Shylaja, B., & Saravana Kumar, R. (2022). Deep learning image inpainting techniques: An overview. *Grenze International Journal of Engineering and Technology, 8*(1), 801.

[21] Sudha, V. K., & Kumar, D. (2023). Hybrid CNN and LSTM network for heart disease prediction. *SN Computer Science, 4*, 1–13. https://doi.org/10.1007/s42979-022-01598-9

[22] Talukder, M. A., & Rahman, M. M. (2024). MLSTL-WSN: Machine learning-based intrusion detection for wireless sensor networks with high security performance. Security and Communication Networks. https://doi.org/10.1007/s10207-024-00833-Z

Authors Biography

Shylaja B is currently pursuing a Ph.D. as a Research Scholar in the Department of Computer Science at Dayananda Sagar Academy of Technology & Management, Bengaluru, Karnataka, India, under Visvesvaraya Technological University (VTU), Belagavi. She holds an M.Tech degree in Computer Networks and Engineering and a B.E. in Information Science and Engineering, both from VTU, Belagavi. She is presently serving as Assistant Professor in the School of Engineering and Technology, S-Vyasa Deemed To Be University, Bangalore, India. Prior to this, she worked as Assistant Professor in the Department of Computer Science and Engineering at Dayananda Sagar Academy of Technology & Management for 12 years. Her research interests include Data Science, Machine Learning and Computer Networks. She has contributed to several international journals and conferences and actively participates in Faculty Development Programs and workshops. She has a keen interest in Data Science, specificall in Time Series Forecasting education and research.

Dr. R. Saravanakumar R holds degrees in Computer Science and Engineering, including a Ph.D. from Anna University, Chennai in 2015 along with a Master's Degree in Computer Science & Engineering from Anna University, Chennai, and a Bachelor's Degree in CSE from Bharathiyar University, Coimbatore. He has 18+ years of experience in teaching and research, and is currently working as a Professor at Jain (Deemed-to-be University), Bangalore. He has published over 55 refereed publications in renowned Scopus/WoS and SCIE indexed journals and conferences and authored 4 textbooks. He holds 5 Indian patents and has earned an Innovation Grand patent from the UK Patent Office. Additionally, he serves as a Corresponding Editor for IGI Global Publications, CRC Press, and Bentham Science Publications, and actively reviews for international journals. He is also a member of ACM, IAENG, and CSI, focusing his research on Data Analytics, Big Data Analytics, Data Science & Cloud Computing. His expertise is acknowledged by the academic community, actively contributing as a reviewer, technical committee member, Assessment and Accreditation Process (NBA, NIRF, and NAAC).