IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 August 2023, accepted 24 November 2023, date of publication 5 December 2023,
date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339584

RESEARCH ARTICLE

Temporal Modeling of Instantaneous Interbeat Interval Based on Physical Activity

HAMED MOJTAHED[1,2], (Student Member, IEEE), RAMESH R. RAO[1], (Fellow, IEEE), CHRISTOPHER PAOLINI[2], (Member, IEEE), AND MAHASWETA SARKAR[2], (Member, IEEE)
[1]Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA
[2]Department of Electrical and Computer Engineering, San Diego State University, San Diego, CA 92182, USA

Corresponding author: Hamed Mojtahed (hmojtahed@ucsd.edu)

ABSTRACT Heartbeat serves as a vital sign of health, aiding the diagnosis of various health issues. The autonomic nervous system (ANS) is responsible for regulating heartbeat, and physical activity (PA) can influence the heart rhythm. Changes in heart rate can signal alterations in PA. However, poor measurements or extreme conditions can result in the loss of heartbeat data, which can negatively impact the analysis of heartbeats. To accurately monitor the Heart Rate Variability (HRV) and cardiovascular health, a proper model to compensate for lost data is necessary. This study investigated the effect of different PAs on InterBeat Interval (IBI) prediction and the possibility of using models trained on unrelated activities to predict the next IBI. The IBI series is divided into piecewise stationary sections based on PA, that is, running, walking, and sitting, as verified by a statistical test. Various machine and deep learning methods have been used to model the IBIs related to specific activities. The models were then used to predict the next IBI for the testing sets of related and unrelated activities, and the error changes were compared for each permutation of training and testing. The models were tested using a Physionet archived dataset. The findings suggest that linear models offer the least prediction error, whereas PA-relevant training could minimize errors in most scenarios. However, in cases where specific PA data are not available, the proposed CNN model demonstrated superior generalization capabilities. These findings can improve HRV error correction techniques and enhance cardiovascular health and ANS monitoring.

INDEX TERMS Machine learning, piecewise stationarity, time series prediction, heart rate, interbeat interval, physical activity.

I. INTRODUCTION

The electrical conduction system in the heart consists of a specialized conduction pathway that is responsible for cardiomyocyte contractions and blood pumping. Conduction is initiated by the sinoatrial (SA) node located in the right atrium, which is the spontaneous source of cardiac impulses. After contraction of the muscles in the atria, the atrioventricular (AV) node conducts the impulse as the blood enters the ventricle to the His bundle and passes to the Purkinje fibers, causing the papillary muscles to contract and constrict the chordae tendineae. Subsequently, ventricular muscle contraction occurs. This depolarization of the myocardium can be recorded by an electrocardiogram (ECG) signal [1], [2]. ECG signals reflect physiological changes in response to physical activity (PA) [3]. Sympathetic and parasympathetic inputs from the autonomic nervous system (ANS) to the SA node control heart rhythm. The sympathetic nervous system is triggered by PA, emotional excitement, or different pathological circumstances. Activation of the sympathetic nervous system causes the release of norepinephrine from the adrenal gland, a neural ganglion located on the kidneys, and an increase in heart rate. Shorter InterBeat Intervals (IBI), i.e., R-R intervals, could result from increased sympathetic input to the SA node. The vagus nerve in the parasympathetic nervous system controls the SA node, which lowers the heart rate and can cause longer IBI [4], [5], [6]. This effect is illustrated in Figure 1. Therefore, IBIs may exhibit

The associate editor coordinating the review of this manuscript and approving it for publication was Aasia Khanum.

nonstationarity in response to changes in PA. The relationship between IBI in (s) and heart rate (HR) in (beats/min) is shown in Equation 1. The normal human IBI range varies from 0.6-1 s that corresponds to normal sinus rhythms of 100-60 bpm [1], [7].

Modeling the human heartbeat based on previous observations is useful in various applications, e.g., heart rate variability (HRV) analysis [8], error correction [9], personalized healthcare monitoring, early detection of heart abnormalities [10], and sleep apnea diagnosis [11]. Therefore, paying attention to the sufficiency and relevance of the recorded data when such models are trained is crucial. Additionally, modeling IBIs based on PA may open an avenue for identifying changes in PA. In their study, Svane et al. demonstrated the effectiveness of Recurrent Neural Networks (RNNs) in correcting artifacts during physical exertion. The study found that RNNs employed for error correction outperform existing techniques in deriving HRV measures [12].

Our approach aims to establish the feasibility of accurately predicting the next IBI by utilizing related and unrelated segments of the IBI series input based on PA by employing several machine learning models. This study addresses the question of whether utilizing piecewise stationary sections of the heart IBI series can enhance the prediction accuracy. To achieve this, the IBI time series was sectioned into piecewise stationary sections based on the PA. We investigated the possibility of more accurate modeling and improved prediction based on machine learning models trained using these sections.

Factors such as electromagnetic interference, muscle movement, transmission media, variations in device usage, and signal processing limitations affect the occurrence of artifacts and errors when measuring heartbeats. These uncertainties make it difficult to construct deterministic or physics-based models that can accurately predict IBIs. Machine learning techniques excel when the underlying phenomenon is broad and uncertain. They can learn complex patterns and relationships that traditional rule-based or deterministic models cannot easily capture. In addition, the inherent variability in the heart rate renders traditional statistical analysis methods ineffective. Machine learning algorithms can handle the variability and noise present in the data, allowing them to adapt and make predictions based on the learned patterns, even in the presence of artifacts and errors, in contrast to deterministic models that may struggle to account for the diverse and complex factors that affect IBI measurements.

The remainder of this paper is organized as follows. Section II reviews previous related studies. Section III describes the methods and metrics used for the detailed evaluation. Subsection III-B presents a statistical test to assess the stationarity of the IBI series for each subject, both collectively and individually, based on the PA. This analysis sets the foundation for evaluating the changes in accuracy, as discussed in Section V. In Sections IV and V, we examine the performance of the models and explore how they are impacted through the connections to the autonomic nervous system. Section VII presents recommendations for further study and potential areas of exploration.

$$HR(bpm) = 60(\text{s/min}) \times \frac{1}{IBI(\text{s})} \tag{1}$$

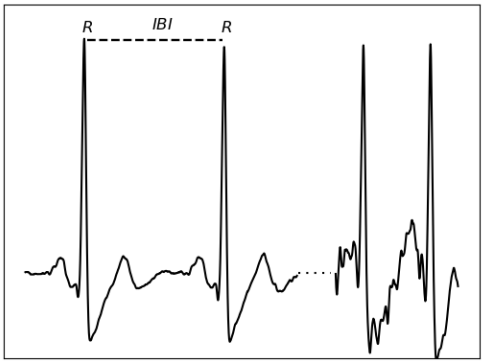

FIGURE 1. Demonstration of IBI, the duration between two R-Waves, indicated with a dotted line in an ECG signal. The left segment corresponds to a period of rest, and the right segment corresponds to a period of High-Intensity activity.

II. LITERATURE REVIEW

The prediction of heartbeat has been the primary focus of previous studies; however, IBI prediction has not been directly explored. In a study by Luo et al., a multi-feature long short-term memory (LSTM) model network was developed using various physiological parameters to predict heartbeat [13]. Staffini et al. conducted a similar analysis on HR data by comparing three different forecasting models (an autoregressive (AR) model, a long short-term memory network, and a convolutional long short-term memory network), and found that the AR model provided the most accurate predictions [14]. Alharbi et al. used a number of deep learning techniques for an HR time series, including an RNN, LSTM, gated recurrent unit (GRU), and bidirectional LSTM for the prediction of HR 5-15 min in advance, and concluded that the GRU model works best in 5 min scenarios [15]. Another study by Oyeleye et al. studied several machine learning models on an accelerometer's univariant HR time series and found that the ARIMA model with a walk-forward validation and linear regression works best for data with an arbitrary duration, and the decision tree regressor and support vector regressor work best for more extended duration data [10]. Masum et al. [16] explored different machine learning algorithms for predicting blood pressure and HR using univariate and multivariate forecasting models employing LSTM, BI-LSTM and CNN (Convolutional Neural Network) machine learning algorithms. In [17], to estimate the HR, the authors used a convolutional neural network architecture that takes the time-frequency spectra of synchronized photoplethysmography (PPG) and accelerometer signals and outputs the estimated HR. The use of Feedforward Neural Networks (FFNN) for

heartbeat prediction based on activity has been studied in several prior research efforts. Biswas et al. [18] presented a deep learning framework for estimating heart rate using a single-channel PPG signal. In [19] and [20], the authors suggested a multilayer FFNN that used PA and HR data to make predictions. In [21], a predictive model based on cycling cadence was suggested, which outputs HR predictions based on HR and cadence.

III. METHODS

The methods used in this research consist of the use of several machine learning and deep learning algorithms to predict the next IBI, namely:

- Autoregressive (AR)
- Autoregressive Moving Average (ARMA)
- Generalized Linear Model (GLM)
- K-Nearest Neighbors (KNN)
- Gradient Boosting (GB)
- Artificial Neural Network (ANN)
- Convolutional Neural Network (CNN)
- Long Short-term Memory (LSTM)

For each subject, these models were trained on the training set chosen from one activity, and then the trained model was used to make predictions on the test set of the same activity and the two other activities. For example, we used the AR model trained on the Run series of a given subject to make predictions on the test set of the Run, on the Sit, and the Walk series. The error measures obtained for each activity were then compared across subjects. The error measures used to measure the accuracy of the models were the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Average Error (MAE):

$$\begin{aligned} MSE &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \\ RMSE &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \\ MAE &= \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \end{aligned} \tag{2}$$

where, in Equations 2, $\hat{y}$ is the prediction made by the models and y is the true value. The efficiency of each model, when used to make predictions for each activity, was compared using the Percentage Error Change (PEC) and Mean Absolute Scaled Error (MASE) [22]:

$$PEC = \frac{e_{oth} - e_{rel}}{e_{rel}} \times 100 \tag{3}$$

$$MASE = \frac{MAE_{rel}}{MAE_{oth}} = \frac{\frac{1}{n}\sum_{i=1}^{n}\left|y_{rel/i} - \hat{y}_{rel/i}\right|}{\frac{1}{m}\sum_{i=1}^{m}\left|y_{oth/i} - \hat{y}_{oth/i}\right|} \tag{4}$$

where in Equation 3, e_{rel} is the error obtained from the relevant (same) activity, e_{oth} is the error obtained from the other activities when a relevant model, i.e., the model trained on the relevant activity, is used to make the prediction. The *PEC* is the percentage amount in which the prediction error changes when other activities are used to test the model. $PEC > 0$ indicates an increase in the error when using the IBI series other than the original activity on which the model was trained. The numerator in the MASE in Equation 4 is the MAE obtained from the test set of the pertinent model, and the denominator is the MAE obtained from the error measures of the two other activities. A $MASE > 1$ indicates a deterioration in the predictions.

A. DATASET

This study used the Pulse Transit Time PPG dataset from the Physionet archive to predict the instantaneous IBI. The dataset contains ECG recordings acquired at 500Hz from 22 healthy subjects from university students without a history of cardiovascular or respiratory disease, female and male, with an age range of 20 to 53 years, and with a mean of 28.52 years while performing three physical activities, i.e., sitting, stationary walking, and running in random order [23], [24]. The selection of PAs is aimed to encompass a wide range of heart rate and cardiovascular responses at various levels of physical stress. Conducting these activities in a random order helps to ensure that the results are not biased by the sequence in which the activities are performed. The study compares IBI prediction results for each of the subjects' PAs with themselves and then aggregate results; consequently, the impact of confounding variables such as age and sex is less significant. Each participant acts as their control, which can inherently account for individual differences. This approach helps isolate the effect of the variable of interest, i.e., IBI response to different PAs, by reducing the variability caused by individual differences [25], [26]. The study did not specifically target faculties or departments more likely to include athletes in the participant group. During each experiment, participants were instructed to perform walking and running exercises in place with controlled movements with minor forward/backward motion. Walking was performed at a moderate pace, which was neither relaxed nor rushed. Running was also performed at a light pace, not completely relaxed, but not rushed. Both activities were performed at similar speeds for all the subjects, with targets of approximately 5 km/h for walking and 9 km/h for running. Participants maintained a short break of less than 5 min between trials to gather various physiological measurements [27]. The duration of recording for each activity was approximately 8 min long. The R-peaks in the ECG signals were annotated using the Kubios HRV software [28], manually inspected, and corrected to obtain the corresponding IBI series. Thus, we ensured that no noise was present in the time series. The distribution of the average IBI for all the subjects performing the activities is shown in Figure 2. The statistics for all subjects performing running indicated a mean of 0.67 s (0.08), whereas sitting exhibited a higher mean of 0.8 s (0.11), and walking has a mean of 0.69 s (0.09). The means for walking and running activities were

similar while sitting demonstrated a much more distinct mean than movement activities. Owing to the similarity in the mean values of the IBI, it is challenging to differentiate between running and walking solely based on the observational mean.

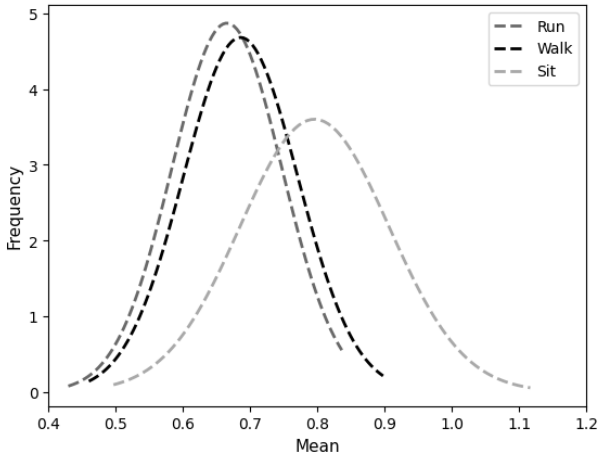

FIGURE 2. Distribution of mean IBI for activities among the 22 subjects.

B. STATIONARITY

Considering the parasympathetic and sympathetic inputs to the SA node, the IBI series exhibits dependence. By splitting each IBI time series into sections of the respective PAs, we can dominate the effect of sympathetic or parasympathetic input and construct stationary IBI segments, i.e., the statistical properties of the series, such as mean and variance, do not change over time, and the covariance of each process depends on temporal information by the time difference [29] to improve prediction. To assess this assumption, the stationarity hypothesis of the individual sections of the IBI time series for each activity and the sequence of the three time series as one piece for each subject were evaluated using the Augmented Dickey-Fuller (ADF) test. ADF verifies the stationarity of the signal by rejecting the null hypothesis, where the unit root is present [30], [31]. The number of lags for the regression model in the stationarity test is selected based on the Akaike information criterion (AIC) [32], [33]:

$$AIC = 2k - \ln(\hat{L}) \tag{5}$$

where k is the number of parameters and $\hat{L}$ is the log-likelihood. When the p-value is not less than the significance level ($p > 0.05$), to reject the null hypothesis, the test statistic and critical values are used to determine stationarity.

Based on the results obtained from the ADF test of the IBI series shown in Table 11, we divided the subjects into three groups: 1) Subjects 4, 8, 9, 11, 13, 15, 19, 20, and 22 had stationary IBI. 2) The ADF does not conclude stationarity for subjects 1, 2, 3, 5, 6, 7, 10, 12, 14, 16, 17, and 18. 3) The null hypothesis for Subject 21 was rejected only under 5% and 10% critical values, but not under 1%. For simplicity, the results of Groups 2 and 3 will be analyzed collectively in future sections. The stationarity results obtained for the subjects in Group 1 could be due to the running pace during the experiment or the subjects' fitness level, which did not result in complete activation of the sympathetic nervous system response. Testing the stationarity of the IBI series segments divided by activities for each individual subject demonstrated the stationarity of the time series belonging to each individual PA. Figure 3 illustrates an example of stationary (Subject 4) and non-stationary (Subject 10) IBI series. The IBI series for the non-stationary subject exhibited visible variations in magnitude based on a change in PA.

C. AR MODELING

The autoregressive model assumes that earlier heartbeat observations can be used to forecast the value at subsequent time steps. Considering the AR model with order $n, n > 0$, the objective is to model the IBI train series to make the prediction of $\hat{r}_{k+1}$ given a series of $R_{k-n:k} = \langle r_{k-n}, r_{k-n+1}, \ldots, r_k \rangle, k \geq n$ where r denotes the successive distance between the two R waves. The predicted $\hat{r}_{k+1}$ value can be obtained as a linear combination of the input time series $R_{k-n:k}$ as follows:

$$\hat{r}_{k+1} = \theta_0 + \theta_1 r_k + \theta_2 r_{k-1} + \ldots + \theta_n r_{k-n} \tag{6}$$

where the coefficients θ_i, $i = 0, \ldots, n$ are obtained by optimizing the model on the training series using the Conditional Maximum Likelihood conditioned on the first observation [33], [34]. The model producing the lowest AIC in Equation 5 was chosen to determine the number of lags (n). As an example, the Partial Autocorrelation Function (PACF) and Autocorrelation Function (ACF) for subject 3 are shown in Figure 4. The figures indicate a strong correlation between a few neighboring beats. An 80:20 ratio is utilized for training and testing.

D. ARMA MODELING

An ARMA(p, q) process can model a stationary IBI series $R_{k-n:k}$ to obtain prediction $\hat{r}_{k+1}$ by Equation 7.

$$\begin{aligned}\hat{r}_{k+1} = {} & \theta_1 r_{k-1} + \ldots + \theta_p r_{k-p} + \varepsilon_k \\ & + \phi_1 \varepsilon_{k-1} + \ldots + \phi_q \varepsilon_{k-q}\end{aligned} \tag{7}$$

The residual ε_t was obtained from the difference between the actual and predicted values, i.e., $\varepsilon_t = y_t - \hat{y}_t$. The parameters in the ARMA model were obtained by a step-wise search to minimize the AIC by maximum likelihood estimation [35], [36]. By incorporating MA terms, ARMA models can capture recurring trends in residuals. Consequently, any systematic deviation from the pure autoregressive behavior that the AR model may not fully account for can be addressed more effectively.

E. GLM MODELING

The GLM model assumes that the response variable, future IBI prediction, has a variance that can be described by a one-parameter exponential probability distribution. In this model, a systematic component connects the linear predictor η with

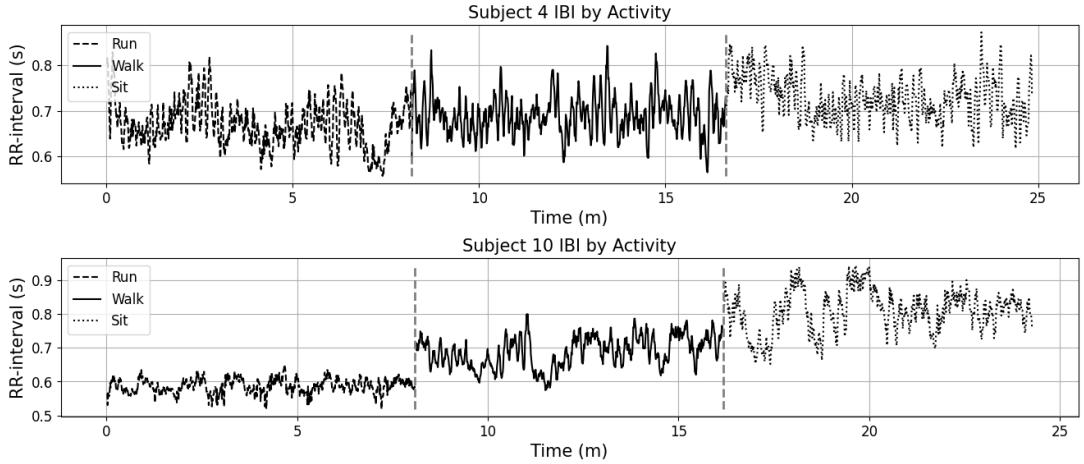

FIGURE 3. IBI series for subject 4 (Stationary) and subject 10 (Non-Stationary).

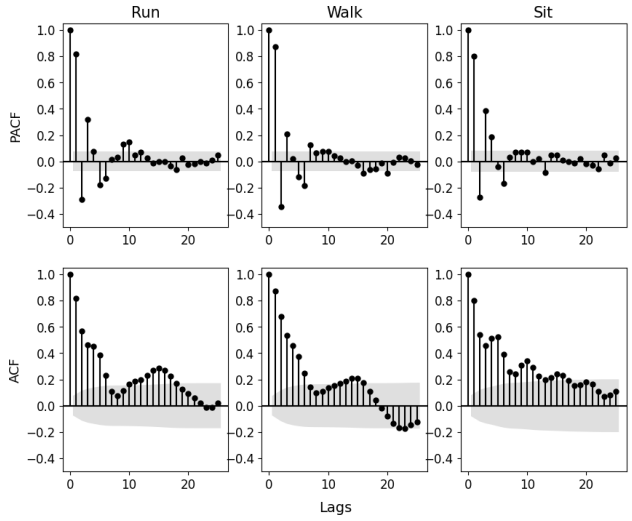

FIGURE 4. PACF and ACF plots for subject 3 across PAs).

the predictor matrix R or the previous IBI observations and the parameter θ, i.e. $\eta = R \cdot \theta$. A differentiable and monotonic link function g links the response and systematic components with $E(r) = g^{-1}(\eta)$. The mean of the response can be expressed as a linear combination of predictors [37]:

$$E[r] = \theta_0 + \theta_1 R_1 + \ldots + \theta_{n-1} R_{n-1} \tag{8}$$

The number of exogenous variables for this model was chosen based on the AR model described in Section III-C. The parameter θ was optimized using iteratively re-weighted least squares [33], [38], and the model *goodness-of-fit* was assessed using the coefficient of determination R^2. This model comprised inverse Gaussian (IG) in Equation 9 and the identity link function $\eta = \mu$.

$$p_R(r; \lambda, \mu) = \sqrt{\frac{\lambda}{2\pi r^3}} \exp\left[\frac{-\lambda(r-\mu)^2}{2\mu^2 r}\right], \; r > 0, \lambda, \mu > 0 \tag{9}$$

F. KNN MODELING

KNN is a classification and regression algorithm that relies on previous observations. Regression enables the use of KNN for numerical forecasting. In regression, a set of features and a target or predicted value are used to describe each data point. KNN works by finding the k most similar previously stored observations to a new data point by computing a distance metric and outputting the aggregation of their target values for an unknown target [39]. When KNN is used to predict a time series, the n lags in the time series construct a feature vector. The observation after the n lags is the target value [40]. Consider the training set comprising the n feature vectors of window length L and the corresponding target values:

$$\begin{aligned} &r_1, r_2, r_3, \ldots, r_L : r_{L+1} \\ &r_2, r_3, r_4, \ldots, r_{L+1} : r_{L+2} \\ &\ldots \\ &r_n, r_{n+1}, r_{n+2}, \ldots, r_{n+L} : r_{n+L+1} \end{aligned}$$

For a given vector in the test set, $r_m, r_{m+1}, r_{m+2}, \ldots, r_{m+L}$, the KNN model finds the k-nearest neighbor to the feature vector in the training vector based on the Euclidean distance and aggregates the target values as a prediction of the given vector. An example of 5-NN training and prediction using this model for a window size of three is illustrated in Figure 6. In this case, the set of observations, represented as lags of the time series, is visualized as points in 3D space. The graph on the right displays the corresponding target values for the top 13 closest neighbors. The prediction is made by aggregating the values of the five closest neighbors obtained by the Euclidean distance.

AIC has been used in earlier studies to determine the number of closest neighbors [41], [42]. A grid search was conducted to determine the optimal number of neighbors (k), and window of the IBIs (n) by minimizing the AIC using least squares regression [43].

$$AIC = 2k - n\ln(\frac{\sum \epsilon^2}{n}) = 2k + n\ln(MSE) \tag{10}$$

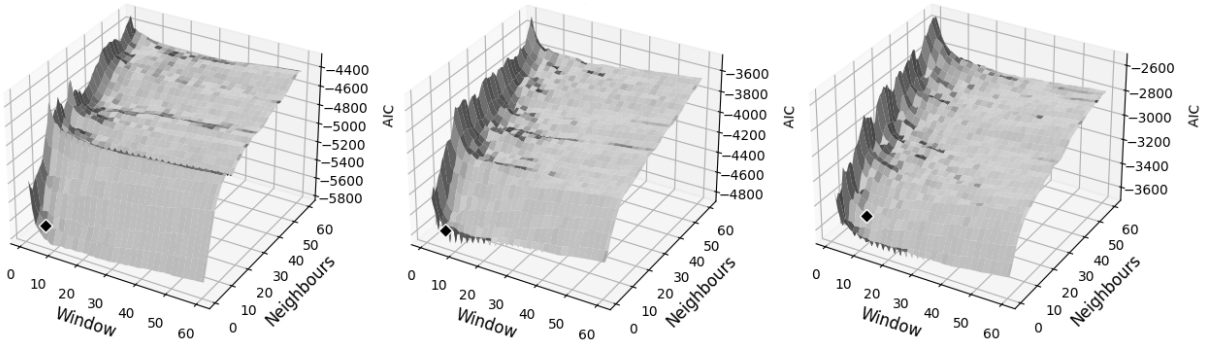

FIGURE 5. Surface plots generated by minimizing AIC for subject 5 (Arranged from left to right: Run, Walk, and Sit sections). The Minima are marked.

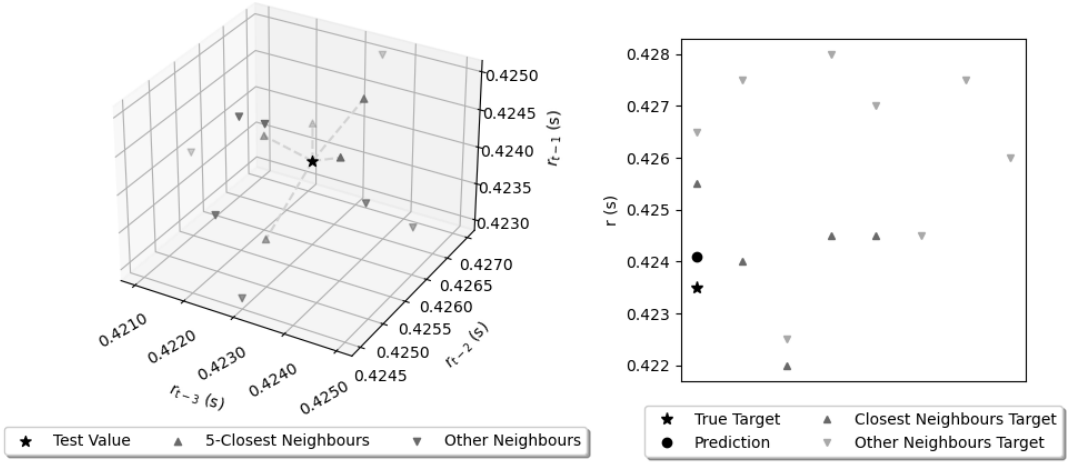

FIGURE 6. Visualization of training and test data with a window size of 3 and 5 neighbors (Left), and model predictions (Right).

where MSE is the MSE of the training set in Equation 10. Figure 5 illustrates the surface plot for Subject 5 using a grid search over the AIC.

G. GRADIENT BOOSTING MODELING

Gradient boosting (GB) is a highly effective machine-learning algorithm that utilizes weak learners, typically decision trees, to create a stronger model [44], [45]. GB can be applied to classification and regression problems. While predicting a time series, the GB is trained using historical data to forecast future data. The input to the GB is a series of previous IBI and the output is the next IBI. A decision tree with a depth of three for a window size of three is shown in Figure 7. The final prediction is obtained by adding the weak learners' predictions (trees) to the initial prediction, i.e., the mean target value of the training data, where the learning rate scales each tree's prediction. Subsequent learners are trained to predict the negative gradient (pseudo-residuals) with respect to the loss function of the preceding iteration. The hyperparameters of the GB model were obtained through Bayesian optimization with Gaussian processes during the 5-fold hv-block cross-validation [46], [47], [48].

H. ANN MODELING

Artificial neural networks are powerful time series prediction tools that can learn from past data and make accurate predictions of future values. They are highly effective in identifying complex patterns, which are capable of deducing nonlinear relationships between input data [49]. The proposed Fully Connected (FC) network shown in Figure 8 was constructed using five FC layers with Rectified Linear Unit (ReLU) activation and an output layer. Bayesian optimization with Gaussian Processes, in conjunction with hv-block cross-validation, were employed to determine the optimal values for the ANN model's hyperparameters, such as the number of layers, number of nodes, dropout rate, learning rate, optimizer, activation function, and number of epochs. During the training, the model was constructed using 70% of the training data and 10% of the data used for validation purposes. The remaining 20% of the data were used to test the model. All deep learning-based models utilized similar splits.

I. CNN MODELING

CNN-based architectures are widely used in computer vision (CV) applications owing to their high adeptness in extracting

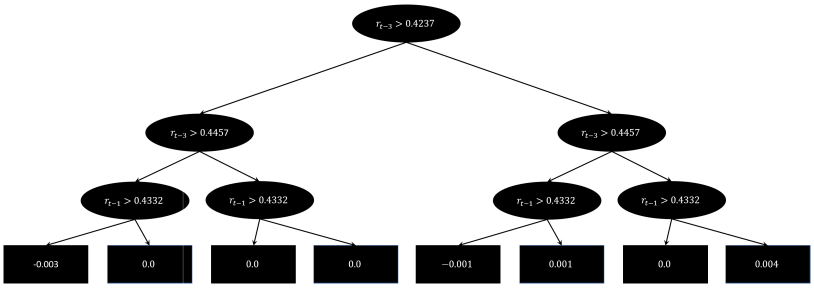

FIGURE 7. Gradient-Boosted decision tree employed for prediction. The leaves represent the residuals of the prediction.

FIGURE 8. Architecture of the ANN.

and learning features. A CNN can produce a smaller set of features owing to its convolution operation [50]. Because CNN can process the input sequence in parallel, it is likely to be quicker than RNN-based approaches. In this study, five distinct CNN architectures with the addition of extra widely used layers in CV were examined. The proposed CNN model comprises a CNN layer with a kernel dimension of 2×2, a dense layer with ReLU activation, and output layer. Similar to the ANN model in Section III-H, the hyperparameters for this model, i.e., window length of input sequences and, number of filters in the convolutional layer, number of perceptions, activation type used in the dense layer, loss function, number of epochs, learning rate, and optimizer were obtained through Bayesian optimization using Gaussian Processes during hv-block cross-validation.

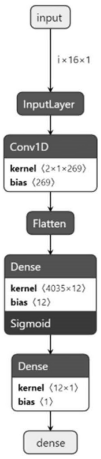

FIGURE 9. The structure of the CNN (CNN - Model 1).

J. LSTM MODELING

One of the proposed solutions to the vanishing gradient problem is the LSTM. Its stability and robustness for modeling long-range dependencies have been demonstrated in several previous studies [51], [52]. The components of an LSTM cell are an input gate, an output gate, and a forget gate. These three gates regulate the flow of information into and out of the cell, and the cell is capable of remembering values across arbitrary time intervals. Recurrent Neural Networks with memory, such as LSTM, can use cached hidden states to estimate network parameters in future time steps [53]. LSTM is capable of capturing long-term temporal links in data, which can improve prediction [54]. LSTM has applications in classifying, processing, and predicting time series with unknown lag [55]. The next IBI can be predicted by LSTM models given a succession of IBI series as inputs by learning the underlying patterns and relationships between the IBIs over time. Several LSTM layers can be utilized to obtain more subtle patterns in the data. Figure 10 depicts the LSTM network architecture employed in this study, which consists of the following layers from left to right: an LSTM layer, a dropout layer, two LSTM layers, another dropout layer, a fully connected layer (dense), and an output layer. The parameters for the LSTM network, i.e., the number of LSTM nodes, optimizer, dropout rate, number of dense nodes, activation type, learning rate, and number of epochs, were determined through Bayesian optimization utilizing Gaussian Processes.

IV. EXPERIMENTS AND RESULTS

Computation for the gradient boosting and deep learning models was performed using an NVIDIA DGX A100 server, equipped with eight NVIDIA A100 40 GB Tensor Core GPUs. Table 1 presents the selected hyperparameters for the deep learning models. The average prediction accuracies of these models across various subjects are detailed in Tables 3 through 10 located in the Appendix. Based on the results, the AR model had the highest error rate when using the Run and Sit models to predict Sit, whereas the model trained on the Walk series exhibited the lowest error rate when predicting Sit IBI. For the ARMA model, the average error was higher than that of the AR model. For this model, we observed a similar

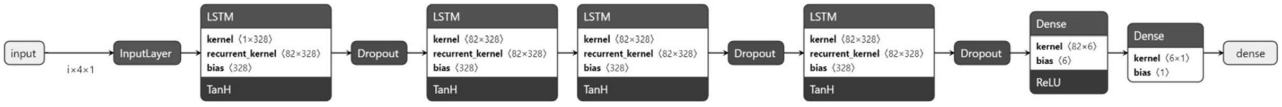

FIGURE 10. Architecture of the LSTM network.

error change pattern from the cross-trained model predictions as in the AR model.

In the GLM model, the Gaussian, Inverse Gaussian (IG), and Gamma distributions were tested along with logarithmic $\eta = ln(\mu)$, identity $\eta = \mu$, and inverse $\eta = \frac{1}{\mu}$ link functions. IG was tested using the inverse squared $\eta = \frac{1}{\mu^2}$ link function. As a result, the lowest error was obtained when IG in Equation 9, and the identity link functions were utilized to construct the model. This confirms previous research that modeled the IBI series using the IG as the underlying distribution [8], [56], [57]. The GLM IG average error change obtained was lower than the AR and ARIMA models. At the same time, the rise in MASE shows that the models are better at making predictions based on unseen data.

In the KNN model, when using a weight function that is proportional to the reciprocal of the distance of the neighbors, where the closest distance exhibits a greater influence, the average error metrics for MSE, RMSE, and MAE are 3.32%, 1.43%, and 1.38% higher, respectively. Overall, the KNN and GB model errors are higher.

We observed a higher error for the ANN and LSTM models than that for the linear models. To prevent overfitting in the ANN model, dropout layers can be used to set the input units to zero at random with a specified frequency at each training step. Suppose a dropout layer with a frequency of 0.1 is introduced to the network after each FC layer. In that case, the overall average MSE, RMSE, and MAE across all models increase by 536.48%, 151.7%, and 181.99%, respectively, for all subjects. This signifies a better model performance when dropout layers are absent. According to the hyperparameter search, the optimal loss function for ANN and CNN was found to be MSE, while for LSTM, MAE was identified to produce the least error.

The CNN model outperformed the average linear model error across all trains and test permutations, resulting from a lower error when non-relevant activity was used for training. A maxpooling layer is often included in CNN. The maxpooling operation decreases the dimensions of the output from the convolutional layer, thus simplifying the network's computations and aiding in reducing over-fitting. This layer can also compress features and extract essential features [58]. In addition to CNN architecture 1 in Figure 9, we experimented with architectures 2 to 5 by including the addition of dropout, maxpooling, or both layers in the architecture. With the addition of a dropout layer after the convolutional layer, the average MSE, RMSE, and MAE increased by 3.8%, 0.83%, and 1.13%, respectively. When a maxpooling layer with a pool size of 2 was used after the convolutional layer, these percentages changed to 85.75%, 12.17%, and 13.89%, respectively. If a dropout layer is used in conjunction with the maxpooling layer, the change increases to 340.46%, 118.29%, and 148.25%. Finally, if the order of dropout and maxpooling layer is replaced, i.e., the convolutional layer followed by the maxpooling layer and then the dropout layer, the metrics change to 19.91%, 9.94%, and 10.82%.

TABLE 1. Overview of parameters used in ANN, CNN, and LSTM networks. Note: "*" denotes values applied for experimental purposes.

	Models		
Parameters	ANN	CNN	LSTM
Epochs	300	288	297
Window size	16	16	4
No. dense layers	4	1	-
No. dense neurons	87	12	6
Dense activation	ReLU	Sigmoid	ReLU
No. convolutional layers	-	1	-
No. convolutional neurons	-	269	-
Kernel size	-	2	-
No. LSTM neurons	-	-	82
No. LSTM layers	-	-	4
LSTM activation	-	-	TanH
Maxpooling size	-	2*	-
Dropout rate	0.1*	0.1*	0.34358
Optimizer	Adam	Adam	Adam
Learning rate	0.000125	0.000732	0.000875
Loss function	MSE	MSE	MAE

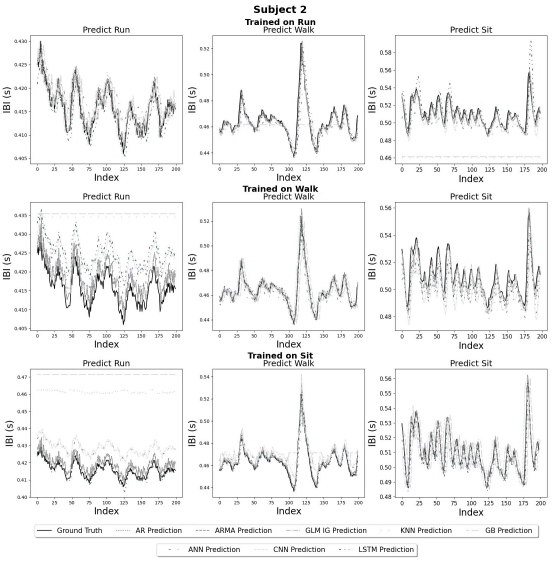

FIGURE 11. Predictions made for subject 2's IBI series utilizing permutation of run, walk, and sit data.

V. DISCUSSION

The present study was conducted to investigate the effect of PA on the accuracy of IBI prediction using machine learning

TABLE 2. Comparison of error measurement change indicators in models for group 1 (Non-Stationary) and groups 2 and 3 (Stationary) subjects.

Models	Train/Test Activity Type	Group 1 (Non-stationary) Average Change (%)			Average	Groups 2 and 3 (Stationary) Average Change (%)			Average
		MSE	RMSE	MAE	MSE	RMSE	MAE	MASE	MSE
AR	Run/Walk	137.46	46.37	38.79	0.77	68.73	25.00	22.24	0.86
	Run/Sit	763.56	181.01	182.17	0.38	447.25	113.01	112.09	0.56
	Walk/Run	37.21	12.41	13.51	0.94	−14.13	−8.09	−8.86	1.11
	Walk/Sit	383.61	113.38	120.18	0.49	203.77	63.71	59.13	0.70
	Sit/Run	82.82	26.88	44.01	0.82	−11.23	−9.74	−5.07	1.14
	Sit/Walk	65.94	23.07	31.63	0.87	−1.60	−5.71	−3.80	1.12
	Mean	245.10	67.19	71.72	0.71	115.46	29.70	29.29	0.92
ARMA	Run/Walk	141.58	48.95	40.87	0.76	67.10	24.67	21.72	0.86
	Run/Sit	780.59	185.37	184.81	0.37	437.58	110.88	108.15	0.57
	Walk/Run	36.38	12.64	12.45	0.95	−11.45	−6.62	−7.81	1.10
	Walk/Sit	386.64	114.69	120.26	0.49	202.27	64.36	59.24	0.69
	Sit/Run	77.79	26.63	41.76	0.82	−3.13	−4.53	1.04	1.05
	Sit/Walk	57.44	21.65	28.65	0.86	4.33	−2.37	0.16	1.06
	Mean	246.74	68.32	71.47	0.71	116.12	31.06	30.42	0.89
GLM IG	Run/Walk	108.55	38.82	35.70	0.79	54.02	20.68	20.06	0.87
	Run/Sit	685.50	169.21	174.64	0.39	416.48	106.70	106.84	0.58
	Walk/Run	36.77	12.06	13.18	0.95	−16.59	−9.36	−9.69	1.12
	Walk/Sit	354.97	106.54	116.87	0.50	181.05	58.00	54.73	0.72
	Sit/Run	74.85	23.70	39.57	0.85	−12.61	−10.88	−6.71	1.18
	Sit/Walk	45.26	14.81	27.11	0.90	−4.50	−7.76	−5.36	1.16
	Mean	217.65	60.86	67.85	0.73	102.97	26.23	26.64	0.94
KNN	Run/Walk	763.78	135.29	129.17	0.60	136.00	41.25	33.74	0.80
	Run/Sit	10385.04	771.64	896.34	0.15	2306.29	304.23	304.19	0.37
	Walk/Run	256.38	56.89	57.45	0.84	−12.36	−7.15	−5.79	1.07
	Walk/Sit	3133.29	414.58	463.25	0.23	685.91	155.75	154.68	0.49
	Sit/Run	1646.55	230.26	298.12	0.44	110.79	35.33	43.26	0.87
	Sit/Walk	515.47	128.26	152.34	0.50	80.28	24.31	26.74	0.93
	Mean	2783.42	289.49	332.78	0.46	551.15	92.29	92.80	0.76
GB	Run/Walk	561.96	110.50	104.67	0.66	119.07	35.20	27.13	0.85
	Run/Sit	8327.60	687.91	797.86	0.17	2034.11	279.15	275.67	0.40
	Walk/Run	279.65	58.49	63.20	0.85	−14.73	−8.76	−7.57	1.10
	Walk/Sit	3002.50	398.90	453.52	0.24	688.93	152.72	153.72	0.52
	Sit/Run	1518.73	191.21	247.35	0.53	146.00	42.93	51.02	0.81
	Sit/Walk	441.59	105.69	123.30	0.61	109.41	27.65	30.74	0.93
	Mean	2355.34	258.78	298.32	0.51	513.80	88.15	88.45	0.77
ANN	Run/Walk	167.54	54.79	51.38	0.72	80.32	26.34	21.61	0.88
	Run/Sit	863.41	187.18	188.62	0.40	516.72	116.55	114.81	0.59
	Walk/Run	19.59	4.25	7.76	1.01	−22.86	−13.06	−11.99	1.15
	Walk/Sit	372.91	110.55	117.19	0.50	230.33	71.30	73.62	0.64
	Sit/Run	39.97	9.86	24.40	0.95	−16.73	−11.85	−9.05	1.17
	Sit/Walk	28.94	7.32	18.43	0.99	−9.57	−9.00	−8.29	1.16
	Mean	248.73	62.32	67.96	0.76	129.70	30.05	30.12	0.93
CNN	Run/Walk	133.82	41.52	36.85	0.80	51.11	18.97	16.15	0.90
	Run/Sit	678.13	157.94	158.75	0.44	298.73	80.03	75.61	0.71
	Walk/Run	22.31	5.97	8.28	1.00	−17.30	−9.91	−11.29	1.15
	Walk/Sit	204.44	70.48	74.19	0.61	162.41	53.80	50.89	0.72
	Sit/Run	−7.53	−9.10	−0.33	1.21	−33.37	−21.58	−20.79	1.36
	Sit/Walk	−8.91	−9.39	−1.93	1.22	−23.68	−16.04	−16.49	1.27
	Mean	170.38	42.90	45.97	0.88	72.98	17.54	15.68	1.02
LSTM	Run/Walk	185.38	58.78	59.60	0.74	82.07	28.04	26.29	0.85
	Run/Sit	1450.19	242.86	267.14	0.35	651.96	147.27	151.21	0.49
	Walk/Run	85.41	26.19	31.36	0.89	−13.91	−7.89	−7.87	1.10
	Walk/Sit	563.22	146.34	156.67	0.46	311.25	92.25	91.94	0.60
	Sit/Run	274.23	69.12	97.66	0.70	34.52	7.38	14.96	1.09
	Sit/Walk	185.79	49.08	71.34	0.78	27.06	5.04	9.63	1.07
	Mean	457.37	98.73	113.96	0.65	182.16	45.35	47.69	0.87

techniques. Physical activity highly impacts the duration of IBI, as it causes the cardiovascular system to increase cardiac output, leading the heart to pump more blood. This decreases the duration between heartbeats, resulting in shorter IBI. During the rest period, with reduced oxygen demand, the heart rate decreases, resulting in greater IBIs. Specifically, in high-intensity PAs, such as running, the IBIs become shorter; in moderate PAs, such as walking, the IBIs increase; and in low-intensity physical activities, such as sitting, the IBIs continue to increase. In the majority of training/test scenarios, with some exceptions, the significance of using relevant activities by training on piecewise stationary sections of the IBI series can be observed. Table 2 presents a comparison of the error change for the permutation of training and tests across PAs. An analysis of the results reveals that the lowest error for most models in most scenarios can be achieved when models are trained on data specific to the activity in which predictions are being made. This can be observed in the predictions of Subject 2 in Figure 11, where the diagonal plots signify the training and test on the same activity. When predicting the Run, the AR model trained on the Run data achieved the lowest error, followed by the GLM IG and ARMA models, compared to the rest of the models. Similarly, the AR model trained on Walk data achieved the lowest error when predicting Walk, followed by the GLM IG and ARMA models. Nevertheless, when predicting Walk, the error disparity of the models, i.e., the prediction error gap between the models trained on Walk versus the other activities, was lower when training was performed on Run data than when training on Sit. In this case, the CNN model trained on Run achieved nearly the same error as that trained on Walk. Similarly, the models trained on the Sit data achieved the lowest error when predicting the Sit. In this case, the GLM IG model achieved the lowest error, followed by the AR and ARMA models. When making predictions using models trained on data not specific to the activity being predicted, the GLM IG model best predicted Run from Walk data and Walk from Run or Sit data. The CNN model achieved the lowest error when predicting the Run from Sit, Sit from Run, and Sit from Walk. When trained on Run data, the models' most accurate predictions after the Run series are made when tested on the Walk series, while predictions made on the Sit series produce more errors across all models. The CNN and GLM IG models produced the slightest error predicting both Walk and Sit, whereas the KNN and GB models produced the highest errors. When trained on Walk data, the majority of the models (AR, ARMA, GLM IG, ANN, and CNN) made smaller errors in predicting Run than making a prediction on Walk, whereas the KNN, GB, and LSTM models made the least error when predicting Walk from Walk. On the other hand, The CNN model produced the least error when predicting Sit from Walk data, followed by the GLM IG model, and the KNN model produced the highest error, followed by the GB model. When trained on Sit data, all models except the CNN model, made more accurate Sit predictions than those of Run or Walk. The KNN model made the least accurate Run or Walk from Sit predictions. The CNN model demonstrated a superior ability to predict data from non-relevant data and reached the lowest error when training and testing across various activities amongst the rest of the models. The addition of dropout and maxpooling layers resulted in a decrease in the model's accuracy. This was due to the disruption caused by the sequential nature of the data and the temporal dependency between the previous observations. The dropout layer randomly sets some input units to zero during each iteration, resulting in a loss of critical temporal data. Similarly, maxpooling reduces the data dimensionality, leading to the loss of vital temporal patterns. In contrast to CNN, the KNN and GB models produced higher errors during prediction when tested on non-relevant activities due to the fact that these models' predictions highly depend on previously observed (training) data. Therefore, the error rate obtained from these models could be an indicator of changes in the PA. The limitations

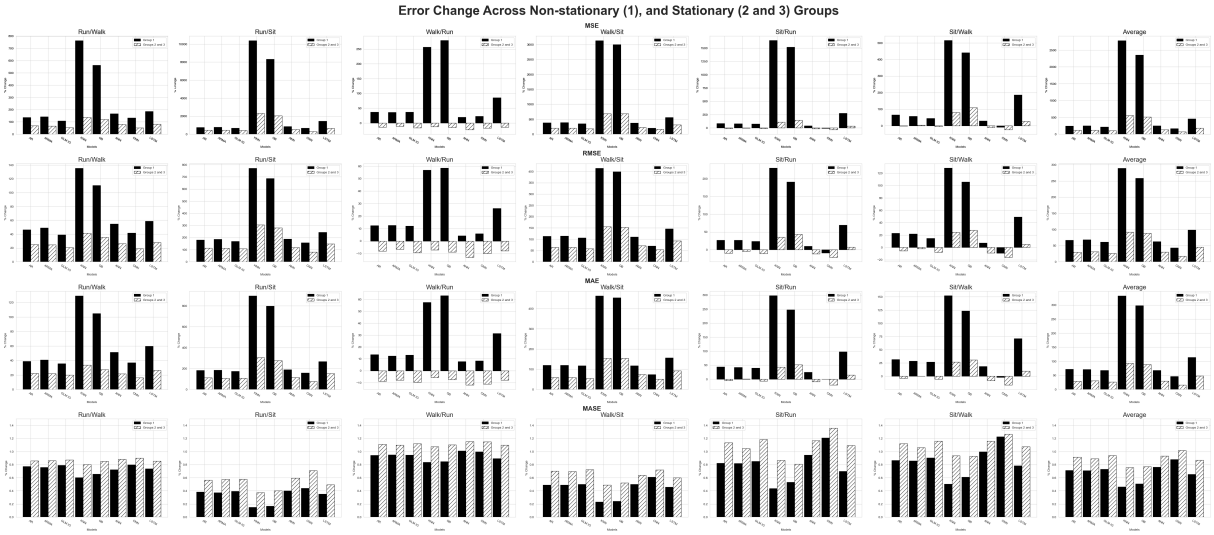

FIGURE 12. Comparison of error change metrics: MSE, RMSE, MAE, and MASE for non-stationary and stationary subjects.

of KNN in generalizing to test data stem from its inability to learn meaningful representations and patterns from the training data. Instead, it relies on the similarity between data points, causing it to struggle with high-dimensional or complex datasets and leading to suboptimal generalization to unseen examples. In the GB, the selection of the bias value as the average of the target values in the training data may adversely affect the performance of the model on the test data. The initialization bias may result in a shift in the bias during the boosting procedure, which can contribute to the insufficient generalization of the test data. On the other hand, linear models rely highly on previous observations when making predictions, making them well-suited for sequential data, where the prediction of the current data point depends on the previous ones. Deep learning-based models offer advantages in terms of learning patterns and capturing complex relationships from data during training. They are adept at handling heterogeneous data and extracting relevant features, without requiring manual feature engineering. This enables them to be more effective in prediction tasks where the underlying patterns may be complex and challenging to discover using conventional machine learning algorithms. When examining the results with respect to groups distinguished by stationarity in III-B for Group 1 (i.e., subjects with non-stationary IBI series) versus Group 2, and Group 3 (i.e., subjects with stationary IBI series), the models trained on Group 1 generated the highest increase in error and smaller MASE when non-relevant activity was used for testing. In contrast, for Groups 2 and 3, when trained on Walk and predicting Run, all models showed a decrease in error, and $MASE > 1$ signified an improvement in prediction accuracy. When trained on Sit and predicting Walk, the CNN, ANN, GLM IG, AR, and ARMA models also made more accurate predictions. A comparison of the predictions for the stationary and non-stationary groups is shown in Figure 12. The deviation in Groups 2 and 3 could be due to a lack of appropriate control in collecting Run data that did not lead to a complete activation of the sympathetic nervous system by reaching a certain age-based maximum HR threshold during the experiment, considering the fitness level of the subjects.

A. LIMITATION

The data collection procedure was constrained by both walking and running being performed on the spot or with a minor backward/forward motion, which may not precisely reflect actual walking or running conditions [59]. Consequently, distinguishing between these PAs for some subjects proved challenging, as identified through the ADF test.

VI. CONCLUSION

In conclusion, due to the impact of PA on the duration of the IBI, which is caused by the cardiovascular system, the choice of training data is crucial for predicting future IBIs. The analysis of the results suggests that in most cases, the models trained on data specific to the predicted activity achieved the lowest error, indicating the importance of using relevant activity or stationary sections of the IBI series based on the capability of the model of choice. Furthermore, the CNN model demonstrated a superior predictive ability when tested on non-relevant activities, whereas the KNN and GB models produced the highest errors. The deviation in the cross-activity prediction of subjects with stationary IBI series could be due to a lack of appropriate control in collecting running data, which did not lead to complete activation of the sympathetic nervous system and generation of IBI series that were distinct from those generated during walking. Overall, the findings highlight the significance of PA on cardiovascular function, its effect on the IBI series and emphasize the importance of using appropriate models based on the available training data to develop accurate predictive models. Additionally, while training such models it is crucial

TABLE 3. AR model prediction error and error change measures in cross-activity models with permutations of train and test data for group 1 (Non-Stationary) and groups 2 and 3 (Stationary) subjects.

Train/Test Activity Type	Prediction Error			Group 1				Groups 2 and 3			
	Average			Average Change (%)			Average	Average Change (%)			Average
	MSE	RMSE	MAE	MSE	RMSE	MAE	MASE	MSE	RMSE	MAE	MASE
Run/Run	3.529×10^{-4}	1.733×10^{-2}	1.348×10^{-2}	-	-	-	-	-	-	-	-
Run/Walk	5.261×10^{-4}	2.154×10^{-2}	1.664×10^{-2}	137.46	46.37	38.79	0.77	68.73	25.00	22.24	0.86
Run/Sit	1.804×10^{-3}	3.924×10^{-2}	3.123×10^{-2}	763.56	181.01	182.17	0.38	447.25	113.01	112.09	0.56
Walk/Walk	4.271×10^{-4}	1.930×10^{-2}	1.503×10^{-2}	-	-	-	-	-	-	-	-
Walk/Run	4.057×10^{-4}	1.906×10^{-2}	1.483×10^{-2}	37.21	12.41	13.51	0.94	−14.13	−8.09	−8.86	1.11
Walk/Sit	1.344×10^{-3}	3.463×10^{-2}	2.704×10^{-2}	383.61	113.38	120.18	0.49	203.77	63.71	59.13	0.70
Sit/Sit	8.923×10^{-4}	2.801×10^{-2}	2.193×10^{-2}	-	-	-	-	-	-	-	-
Sit/Run	1.296×10^{-3}	3.051×10^{-2}	2.648×10^{-2}	82.82	26.88	44.01	0.82	−11.23	−9.74	−5.07	1.14
Sit/Walk	1.091×10^{-3}	2.944×10^{-2}	2.473×10^{-2}	65.94	23.07	31.63	0.87	−1.60	−5.71	−3.80	1.12
All Permutations Average	9.04×10^{-4}	2.66×10^{-2}	2.13×10^{-2}	245.10	67.19	71.72	0.71	115.46	29.70	29.29	0.92

TABLE 4. ARMA model prediction error and error change measures in cross-activity models with permutations of train and test data for group 1 (Non-Stationary) and groups 2 and 3 (Stationary) subjects.

Train/Test Activity Type	Prediction Error			Group 1				Groups 2 and 3			
	Average			Average Change (%)			Average	Average Change (%)			Average
	MSE	RMSE	MAE	MSE	RMSE	MAE	MASE	MSE	RMSE	MAE	MASE
Run/Run	3.577×10^{-4}	1.751×10^{-2}	1.370×10^{-2}	-	-	-	-	-	-	-	-
Run/Walk	5.595×10^{-4}	2.220×10^{-2}	1.724×10^{-2}	141.58	48.95	40.87	0.76	67.10	24.67	21.72	0.86
Run/Sit	1.866×10^{-3}	4.007×10^{-2}	3.187×10^{-2}	780.59	185.37	184.81	0.37	437.58	110.88	108.15	0.57
Walk/Walk	4.441×10^{-4}	1.963×10^{-2}	1.535×10^{-2}	-	-	-	-	-	-	-	-
Walk/Run	4.227×10^{-4}	1.947×10^{-2}	1.511×10^{-2}	36.38	12.64	12.45	0.95	−11.45	−6.62	−7.81	1.10
Walk/Sit	1.418×10^{-3}	3.547×10^{-2}	2.772×10^{-2}	386.64	114.69	120.26	0.49	202.27	64.36	59.24	0.69
Sit/Sit	8.995×10^{-4}	2.814×10^{-2}	2.214×10^{-2}	-	-	-	-	-	-	-	-
Sit/Run	1.258×10^{-3}	3.137×10^{-2}	2.730×10^{-2}	77.79	26.63	41.76	0.82	−3.13	−4.53	1.04	1.05
Sit/Walk	1.056×10^{-3}	2.982×10^{-2}	2.512×10^{-2}	57.44	21.65	28.65	0.86	4.33	−2.37	0.16	1.06
All Permutations Average	9.202×10^{-4}	2.708×10^{-2}	2.173×10^{-2}	246.74	68.32	71.47	0.71	116.12	31.06	30.42	0.89

TABLE 5. GLM IG model prediction error and error change measures in cross-activity models with permutations of train and test data for group 1 (Non-Stationary) and groups 2 and 3 (Stationary) subjects.

Train/Test Activity Type	Prediction Error			Group 1				Groups 2 and 3			
	Average			Average Change (%)			Average	Average Change (%)			Average
	MSE	RMSE	MAE	MSE	RMSE	MAE	MASE	MSE	RMSE	MAE	MASE
Run/Run	3.534×10^{-4}	1.735×10^{-2}	1.348×10^{-2}	-	-	-	-	-	-	-	-
Run/Walk	5.047×10^{-4}	2.099×10^{-2}	1.638×10^{-2}	108.55	38.82	35.70	0.79	54.02	20.68	20.06	0.87
Run/Sit	1.683×10^{-3}	3.789×10^{-2}	3.042×10^{-2}	685.50	169.21	174.64	0.39	416.48	106.70	106.84	0.58
Walk/Walk	4.273×10^{-4}	1.930×10^{-2}	1.502×10^{-2}	-	-	-	-	-	-	-	-
Walk/Run	4.007×10^{-4}	1.893×10^{-2}	1.474×10^{-2}	36.77	12.06	13.18	0.95	−16.59	−9.36	−9.69	1.12
Walk/Sit	1.255×10^{-3}	3.350×10^{-2}	2.645×10^{-2}	354.97	106.54	116.87	0.50	181.05	58.00	54.73	0.72
Sit/Sit	8.798×10^{-4}	2.785×10^{-2}	2.184×10^{-2}	-	-	-	-	-	-	-	-
Sit/Run	1.250×10^{-3}	2.977×10^{-2}	2.574×10^{-2}	74.85	23.70	39.57	0.85	−12.61	−10.88	−6.71	1.18
Sit/Walk	1.044×10^{-3}	2.846×10^{-2}	2.405×10^{-2}	45.26	14.81	27.11	0.90	−4.50	−7.76	−5.36	1.16
All Permutations Average	8.664×10^{-4}	2.600×10^{-2}	2.090×10^{-2}	217.65	60.86	67.85	0.73	102.97	26.23	26.64	0.94

TABLE 6. KNN model prediction error and error change measures in cross-activity models with permutations of train and test data for group 1 (Non-Stationary) and groups 2 and 3 (Stationary) subjects.

Train/Test Activity Type	Prediction Error			Group 1				Groups 2 and 3			
	Average			Average Change (%)			Average	Average Change (%)			Average
	MSE	RMSE	MAE	MSE	RMSE	MAE	MASE	MSE	RMSE	MAE	MASE
Run/Run	3.683×10^{-4}	1.780×10^{-2}	1.391×10^{-2}	-	-	-	-	-	-	-	-
Run/Walk	1.106×10^{-3}	2.921×10^{-2}	2.290×10^{-2}	763.78	135.29	129.17	0.60	136.00	41.25	33.74	0.80
Run/Sit	1.104×10^{-2}	9.287×10^{-2}	7.974×10^{-2}	10385.04	771.64	896.34	0.15	2306.29	304.23	304.19	0.37
Walk/Walk	4.841×10^{-4}	2.055×10^{-2}	1.606×10^{-2}	-	-	-	-	-	-	-	-
Walk/Run	5.920×10^{-4}	2.263×10^{-2}	1.779×10^{-2}	256.38	56.89	57.45	0.84	−12.36	−7.15	−5.79	1.07
Walk/Sit	6.642×10^{-3}	7.333×10^{-2}	6.106×10^{-2}	3133.29	414.58	463.25	0.23	685.91	155.75	154.68	0.49
Sit/Sit	9.235×10^{-4}	2.876×10^{-2}	2.264×10^{-2}	-	-	-	-	-	-	-	-
Sit/Run	4.577×10^{-5}	6.736×10^{-5}	5.168×10^{-2}	1646.55	230.26	298.12	0.44	110.79	35.33	43.26	0.87
Sit/Walk	3.007×10^{-3}	4.783×10^{-2}	4.109×10^{-2}	515.47	128.26	152.34	0.50	80.28	24.31	26.74	0.93
All Permutations Average	3.193×10^{-3}	4.337×10^{-2}	3.632×10^{-2}	2783.42	289.49	332.78	0.46	551.15	92.29	92.80	0.76

TABLE 7. GB model prediction error and error change measures in cross-activity models with permutation of train and test data for group 1 (Non-Stationary) and groups 2 and 3 (Stationary) subjects.

Train/Test Activity Type	Prediction Error			Group 1				Groups 2 and 3			
	Average			Average Change (%)			Average	Average Change (%)			Average
	MSE	RMSE	MAE	MSE	RMSE	MAE	MASE	MSE	RMSE	MAE	MASE
Run/Run	3.997×10^{-4}	1.864×10^{-2}	1.460×10^{-2}	-	-	-	-	-	-	-	-
Run/Walk	1.059×10^{-3}	2.863×10^{-2}	2.231×10^{-2}	561.96	110.50	104.67	0.66	119.07	35.20	27.13	0.85
Run/Sit	1.053×10^{-2}	9.035×10^{-2}	7.726×10^{-2}	8327.60	687.91	797.86	0.17	2034.11	279.15	275.67	0.40
Walk/Walk	5.231×10^{-4}	2.127×10^{-2}	1.648×10^{-2}	-	-	-	-	-	-	-	-
Walk/Run	6.272×10^{-4}	2.312×10^{-2}	1.819×10^{-2}	279.65	58.49	63.20	0.85	−14.73	−8.76	−7.57	1.10
Walk/Sit	6.686×10^{-3}	7.340×10^{-2}	6.112×10^{-2}	3002.50	398.90	453.52	0.24	688.93	152.72	153.72	0.52
Sit/Sit	1.047×10^{-3}	3.079×10^{-2}	2.399×10^{-2}	-	-	-	-	-	-	-	-
Sit/Run	4.441×10^{-5}	5.636×10^{-2}	4.978×10^{-2}	1518.73	191.21	247.35	0.53	146.00	42.93	51.02	0.81
Sit/Walk	3.041×10^{-3}	4.751×10^{-2}	4.004×10^{-2}	441.59	105.69	123.30	0.61	109.41	27.65	30.74	0.93
All Permutations Average	3.150×10^{-3}	4.334×10^{-2}	3.597×10^{-2}	2355.34	258.78	298.32	0.51	513.80	88.15	88.45	0.77

TABLE 8. ANN model prediction error and error change measures in cross-activity models with permutations of train and test data for group 1 (Non-Stationary) and groups 2 and 3 (Stationary) subjects.

Train/Test Activity Type	Prediction Error			Group 1				Groups 2 and 3			
	Average			Average Change (%)			Average	Average Change (%)			Average
	MSE	RMSE	MAE	MSE	RMSE	MAE	MASE	MSE	RMSE	MAE	MASE
Run/Run	5.108×10^{-4}	2.103×10^{-2}	1.661×10^{-2}	-	-	-	-	-	-	-	-
Run/Walk	7.985×10^{-4}	2.693×10^{-2}	2.119×10^{-2}	167.54	54.79	51.38	0.72	80.32	26.34	21.61	0.88
Run/Sit	2.421×10^{-3}	4.654×10^{-2}	3.715×10^{-2}	863.41	187.18	188.62	0.40	516.72	116.55	114.81	0.59
Walk/Walk	6.777×10^{-4}	2.436×10^{-2}	1.885×10^{-2}	-	-	-	-	-	-	-	-
Walk/Run	5.411×10^{-4}	2.223×10^{-2}	1.758×10^{-2}	19.59	4.25	7.76	1.01	−22.86	−13.06	−11.99	1.15
Walk/Sit	2.112×10^{-3}	4.371×10^{-2}	3.479×10^{-2}	372.91	110.55	117.19	0.50	230.33	71.30	73.62	0.64
Sit/Sit	1.231×10^{-3}	3.332×10^{-2}	2.657×10^{-2}	-	-	-	-	-	-	-	-
Sit/Run	1.283×10^{-3}	3.287×10^{-2}	2.849×10^{-2}	39.97	9.86	24.40	0.95	−16.73	−11.85	−9.05	1.17
Sit/Walk	1.210×10^{-3}	3.258×10^{-2}	2.762×10^{-2}	28.94	7.32	18.43	0.99	−9.57	−9.00	−8.29	1.16
All Permutations Average	1.198×10^{-3}	3.151×10^{-2}	2.543×10^{-2}	248.73	62.32	67.96	0.76	129.70	30.05	30.12	0.93

TABLE 9. CNN model prediction error and error change measures in cross-activity models with permutations of train and test data for group 1 (Non-Stationary) and groups 2 and 3 (Stationary) subjects.

Train/Test Activity Type	Prediction Error			Group 1				Groups 2 and 3			
	Average			Average Change (%)			Average	Average Change (%)			Average
	MSE	RMSE	MAE	MSE	RMSE	MAE	MASE	MSE	RMSE	MAE	MASE
Run/Run	4.038×10^{-4}	1.864×10^{-2}	1.476×10^{-2}	-	-	-	-	-	-	-	-
Run/Walk	5.275×10^{-4}	2.183×10^{-2}	1.711×10^{-2}	133.82	41.52	36.85	0.80	51.11	18.97	16.15	0.90
Run/Sit	1.418×10^{-3}	3.552×10^{-2}	2.810×10^{-2}	678.13	157.94	158.75	0.44	298.73	80.03	75.61	0.71
Walk/Walk	5.373×10^{-4}	2.168×10^{-2}	1.702×10^{-2}	-	-	-	-	-	-	-	-
Walk/Run	4.828×10^{-4}	2.070×10^{-2}	1.626×10^{-2}	22.31	5.97	8.28	1.00	−17.30	−9.91	−11.29	1.15
Walk/Sit	1.268×10^{-3}	3.374×10^{-2}	2.649×10^{-2}	204.44	70.48	74.19	0.61	162.41	53.80	50.89	0.72
Sit/Sit	1.195×10^{-3}	3.250×10^{-2}	2.601×10^{-2}	-	-	-	-	-	-	-	-
Sit/Run	8.304×10^{-4}	2.639×10^{-2}	2.218×10^{-2}	−7.53	−9.10	−0.33	1.21	−33.37	−21.58	−20.79	1.36
Sit/Walk	8.502×10^{-4}	2.712×10^{-2}	2.248×10^{-2}	−8.91	−9.39	−1.93	1.22	−23.68	−16.04	−16.49	1.27
All Permutations Average	8.348×10^{-4}	2.646×10^{-2}	2.116×10^{-2}	170.38	42.90	45.97	0.88	72.98	17.54	15.68	1.02

TABLE 10. LSTM model prediction error and error change measures in cross-activity models with permutations of train and test data for group 1 (Non-Stationary) and groups 2 and 3 (Stationary) subjects.

Train/Test Activity Type	Prediction Error			Group 1				Groups 2 and 3			
	Average			Average Change (%)			Average	Average Change (%)			Average
	MSE	RMSE	MAE	MSE	RMSE	MAE	MASE	MSE	RMSE	MAE	MASE
Run/Run	4.728×10^{-4}	2.013×10^{-2}	1.584×10^{-2}	-	-	-	-	-	-	-	-
Run/Walk	9.065×10^{-4}	2.720×10^{-2}	2.185×10^{-2}	185.38	58.78	59.60	0.74	82.07	28.04	26.29	0.85
Run/Sit	4.011×10^{-5}	5.478×10^{-2}	4.592×10^{-2}	1450.19	242.86	267.14	0.35	651.96	147.27	151.21	0.49
Walk/Walk	5.483×10^{-4}	2.180×10^{-2}	1.719×10^{-2}	-	-	-	-	-	-	-	-
Walk/Run	6.084×10^{-4}	2.274×10^{-2}	1.846×10^{-2}	85.41	26.19	31.36	0.89	−13.91	−7.89	−7.87	1.10
Walk/Sit	2.683×10^{-3}	4.724×10^{-2}	3.829×10^{-2}	563.22	146.34	156.67	0.46	311.25	92.25	91.94	0.60
Sit/Sit	1.294×10^{-3}	3.317×10^{-2}	2.628×10^{-2}	-	-	-	-	-	-	-	-
Sit/Run	3.699×10^{-4}	4.743×10^{-2}	4.295×10^{-2}	274.23	69.12	97.66	0.70	34.52	7.38	14.96	1.09
Sit/Walk	2.745×10^{-3}	4.312×10^{-2}	3.821×10^{-2}	185.79	49.08	71.34	0.78	27.06	5.04	9.63	1.07
All Permutations Average	1.885×10^{-3}	3.529×10^{-2}	2.944×10^{-2}	457.37	98.73	113.96	0.65	182.16	45.35	47.69	0.87

to consider whether different intensities of PA have resulted in varying heart rate levels, and the training process should be structured appropriately to reflect these variations.

VII. FUTURE WORK

In future studies, it would be valuable to explore the benefits of utilizing piecewise stationary sections of data and ML-based prediction models for RR interval correction necessary for heart rate variability (HRV) analysis. The efficacy of the ML-based RR interval correction method can be evaluated by comparing it with traditional RR correction methods such as interpolation, imputation, and exclusion of IBIs.

In addition, it would be beneficial to examine the generalizability of this analysis by conducting experiments on IBI recorded from additional activities, such as standing and lying down, as different activities result in distinct IBI patterns.

Furthermore, it would be valuable to investigate the impact of introducing additional physiological parameters as input features to ML models to evaluate the accuracy

TABLE 11. ADF test results for the subject in the pulse transit time PPG dataset.

Subject	Order	p-value	Test Statistic	Critical Value (1%)
1	25	0.4263	−1.7093	−3.434
2	26	0.4827	−1.6018	−3.432
3	26	0.0678	−2.7375	−3.434
4	12	0	−5.2536	−3.433
5	14	0.1190	−2.4857	−3.433
6	21	0.1497	−2.3722	−3.433
7	21	0.5434	−1.4800	−3.434
8	17	0	−5.5090	−3.433
9	11	0	−5.1518	−3.434
10	18	0.1391	−2.4095	−3.433
11	12	0.0013	−4.0151	−3.434
12	17	0.1759	−2.2879	−3.434
13	10	0	−4.9207	−3.434
14	21	0.6423	−1.2711	−3.435
15	11	0.0001	−4.6004	−3.433
16	15	0.0623	−2.7729	−3.434
17	24	0.4766	−1.6126	−3.433
18	16	0.3158	−1.9350	−3.433
19	26	0.0072	−3.5324	−3.433
20	12	0.0001	−4.5833	−3.433
21	18	0.0389	−2.9588	−3.433
22	10	0.0001	−4.7367	−3.433

of HRV prediction. Parameters such as respiration rate, body temperature, and electrodermal activity offer insights into the physiological state. It could prove advantageous to incorporate these parameters as additional features to enhance the prediction accuracy and provide better insight into the analysis of the autonomic nervous system.

APPENDIX TABLES

See Tables 3–11.

REFERENCES

[1] P. A. Iaizzo, *Handbook of Cardiac Anatomy, Physiology, and Devices*. Berlin, Germany: Springer, 2010.
[2] M. D. Roberts, *Mastering the 12-Lead EKG*. Berlin, Germany: Springer, 2019.
[3] M. L. Simoons and P. G. Hugenholtz, "Gradual changes of ECG waveform during and after exercise in normal subjects," *Circulation*, vol. 52, no. 4, pp. 570–577, Oct. 1975.
[4] R. Gordan, J. K. Gwathmey, and L.-H. Xie, "Autonomic and endocrine control of cardiovascular function," *World J. Cardiol.*, vol. 7, no. 4, p. 204, 2015.
[5] L. K. McCorry, "Physiology of the autonomic nervous system," *Amer. J. Pharmaceutical Educ.*, vol. 71, no. 4, pp. 3–7, 2007.
[6] N. J. Fortuin and J. L. Weiss, "Exercise stress testing," *Circulation*, vol. 56, no. 5, pp. 699–712, 1977.
[7] M. A. Syed, S. V. Raman, and O. P. Simonetti, *Basic Principles of Cardiovascular MRI: Physics and Imaging Techniques*. Berlin, Germany: Springer, 2015.
[8] R. Barbieri, E. C. Matten, A. A. Alabi, and E. N. Brown, "A point-process model of human heartbeat intervals: New definitions of heart rate and heart rate variability," *Amer. J. Physiol.-Heart Circulatory Physiol.*, vol. 288, no. 1, pp. 424–435, Jan. 2005.
[9] M. Benchekroun, B. Chevallier, V. Zalc, D. Istrate, D. Lenne, and N. Vera, "The impact of missing data on heart rate variability features: A comparative study of interpolation methods for ambulatory health monitoring," *IRBM*, vol. 44, no. 4, Aug. 2023, Art. no. 100776.
[10] M. Oyeleye, T. Chen, S. Titarenko, and G. Antoniou, "A predictive analysis of heart rates using machine learning techniques," *Int. J. Environ. Res. Public Health*, vol. 19, no. 4, p. 2417, Feb. 2022.
[11] H. Qin and G. Liu, "A dual-model deep learning method for sleep apnea detection based on representation learning and temporal dependence," *Neurocomputing*, vol. 473, pp. 24–36, Feb. 2022.
[12] J. Svane, T. Wiktorski, S. Ørn, and T. C. Eftestøl, "Recurrent neural networks for artifact correction in HRV data during physical exercise," *Nordic Mach. Intell.*, vol. 3, no. 1, p. 11, May 2023.
[13] M. Luo and K. Wu, "Heart rate prediction model based on neural network," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, vol. 715, 2020, p. 012060.
[14] A. Alharbi, W. Alosaimi, R. Sahal, and H. Saleh, "Real-time system prediction for heart rate using deep learning and stream processing platforms," *Complexity*, vol. 2021, pp. 1–9, Feb. 2021.
[15] A. Staffini, T. Svensson, U.-I. Chung, and A. K. Svensson, "Heart rate modeling and prediction using autoregressive models and deep learning," *Sensors*, vol. 22, no. 1, p. 34, Dec. 2021.
[16] S. Masum, J. P. Chiverton, Y. Liu, and B. Vuksanovic, "Investigation of machine learning techniques in forecasting of blood pressure time series data," in *Artificial Intelligence XXXVI*. Cambridge, U.K.: Springer, 2019, pp. 269–282.
[17] A. Reiss, I. Indlekofer, P. Schmidt, and K. Van Laerhoven, "Deep PPG: Large-scale heart rate estimation with convolutional neural networks," *Sensors*, vol. 19, no. 14, p. 3079, Jul. 2019.
[18] D. Biswas, L. Everson, M. Liu, M. Panwar, B.-E. Verhoef, S. Patki, C. H. Kim, A. Acharyya, C. Van Hoof, M. Konijnenburg, and N. Van Helleputte, "CorNET: Deep learning framework for PPG-based heart rate estimation and biometric identification in ambulant environment," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 2, pp. 282–291, Apr. 2019.
[19] M. Yuchi and J. Jo, "Heart rate prediction based on physical activity using feedforwad neural network," in *Proc. Int. Conf. Converg. Hybrid Inf. Technol.*, 2008, pp. 344–350.
[20] F. Xiao, Y. Chen, M. Yuchi, M. Ding, and J. Jo, "Heart rate prediction model based on physical activities using evolutionary neural network," in *Proc. 4th Int. Conf. Genetic Evol. Comput.*, Dec. 2010, pp. 198–201.
[21] K. Mutijarsa, M. Ichwan, and D. B. Utami, "Heart rate prediction based on cycling cadence using feedforward neural network," in *Proc. Int. Conf. Comput., Control, Informat. Appl. (ICINA)*, Oct. 2016, pp. 72–76.
[22] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, Oct. 2006.
[23] P. Mehrgardt, M. Khushi, S. Poon, and A. Withana, "Pulse transit time PPG dataset," *PhysioNet*, vol. 10, pp. 215–220, Jan. 2022.
[24] P. PhysioBank, "PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. 215–220, 2000.
[25] D. Lane, D. Scott, M. Hebl, R. Guerra, D. Osherson, and H. Zimmer, *Introduction to Statistics*. 2003. [Online]. Available: http://onlinestatbook.com/
[26] A. Becker, "The benefits of single-subject research designs and multi-methodological approaches for neuroscience research," *Frontiers Hum. Neurosci.*, vol. 17, Oct. 2023, Art. no. 1190412.
[27] P. Mehrgardt, personal communication, Apr. 2023.
[28] M. P. Tarvainen, J.-P. Niskanen, J. A. Lipponen, P. O. Ranta-Aho, and P. A. Karjalainen, "Kubios HRV—Heart rate variability analysis software," *Comput. Methods Programs Biomed.*, vol. 113, no. 1, pp. 210–220, 2014.
[29] J. D. Cryer and K.-S. Chan, *Time Series Analysis: With Applications in R*, vol. 2. Berlin, Germany: Springer, 2008.
[30] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *J. Amer. Stat. Assoc.*, vol. 74, no. 366, pp. 427–431, Jun. 1979.
[31] J. G. MacKinnon, "Critical values for cointegration tests," Queen's Econ. Dept. Working Paper, Queen's Univ. Kingston, ON, Canada, Tech. Rep. 1227, 2010.
[32] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.
[33] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with Python," in *Proc. 9th Python Sci. Conf.*, 2010, pp. 92–96.
[34] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. Berlin, Germany: Springer, 2002.
[35] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 2020.
[36] T. G. Smith, "Pmdarima: Arima estimators for Python," Tech. Rep., 2017. [Online]. Available: https://alkaline-ml.com/pmdarima/
[37] J. W. Hardin, J. W. Hardin, J. M. Hilbe, and J. Hilbe, *Generalized Linear Models and Extensions*. TX, USA: Stata Press, 2007.
[38] P. J. Green, "Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives," *J. Roy. Stat. Soc., Ser. B*, vol. 46, no. 2, pp. 149–170, Jan. 1984.
[39] F. Martínez, M. P. Frías, M. D. Pérez, and A. J. Rivera, "A methodology for applying k-nearest neighbor to time series forecasting," *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 2019–2037, Oct. 2019.
[40] F. Martínez, M. Frías, F. Charte, and A. Rivera, "Time series forecasting with KNN in R: The tsfknn package," *R J.*, vol. 11, no. 2, p. 229, 2019.

[41] T. Lee and T. B. M. J. Ouarda, "Identification of model order and number of neighbors for k-nearest neighbor resampling," *J. Hydrol.*, vol. 404, nos. 3–4, pp. 136–145, Jul. 2011.
[42] A. G. Karegowda, M. A. Jayaram, and A. S. Manjunath, "Combining Akaike's information criterion (AIC) and the golden-section search technique to find optimal numbers of K-nearest neighbors," *Int. J. Comput. Appl.*, vol. 2, no. 1, pp. 80–87, May 2010.
[43] B. Kp and D. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, vol. 65. New York, NY, USA: Springer, 2002, p. 2335.
[44] B. P. Roe, H.-J. Yang, J. Zhu, Y. Liu, I. Stancu, and G. McGregor, "Boosted decision trees as an alternative to artificial neural networks for particle identification," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 543, nos. 2–3, pp. 577–584, May 2005.
[45] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
[46] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1–9.
[47] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Inf. Sci.*, vol. 191, pp. 192–213, May 2012.
[48] J. Racine, "Consistent cross-validatory model-selection for dependent data: HV-block cross-validation," *J. Econ.*, vol. 99, no. 1, pp. 39–61, Nov. 2000.
[49] N. D. C. Lewis, *Neural Networks for Time Series Forecasting With R: An Intuitive Step by Step Blueprint for Beginners*. SC, USA: CreateSpace Independent Publishing Platform, 2017.
[50] M. Peixeiro, *Time Series Forecasting in Python*. Shelter Island, NY, USA: Manning, 2022.
[51] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*.
[52] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
[53] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[54] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Sci. Rep.*, vol. 8, no. 1, p. 6085, Apr. 2018.
[55] Y. Chen, K. Zhong, J. Zhang, Q. Sun, and X. Zhao, "LSTM networks for mobile human activity recognition," in *Proc. Int. Conf. Artif. Intell., Technol. Appl.* Dordrecht, The Netherlands: Atlantis Press, 2016, pp. 50–53.
[56] G. B. Stanley, K. Poolla, and R. A. Siegel, "Threshold modeling of autonomic control of heart rate variability," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 9, pp. 1147–1153, 2000.
[57] L. Citi, E. N. Brown, and R. Barbieri, "A real-time automated point-process method for the detection and correction of erroneous and ectopic heartbeats," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 10, pp. 2828–2837, Oct. 2012.
[58] R. Wang, Z. Chen, W. Zhang, and Q. Zhu, "Prediction for time series with CNN and LSTM," in *Proc. 11th Int. Conf. Model., Identificat. Control* (Lecture Notes in Electrical Engineering), vol. 582. Singapore: Springer, 2020, pp. 631–641.
[59] P. Mehrgardt, M. Khushi, S. Poon, and A. Withana, "Deep learning fused wearable pressure and PPG data for accurate heart rate monitoring," *IEEE Sensors J.*, vol. 21, no. 23, pp. 27106–27115, Dec. 2021.

HAMED MOJTAHED (Student Member, IEEE) received the B.S. degree (Hons.) in computer engineering from Purdue University, IN, USA, in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of California San Diego and San Diego State University. He has been a Research Assistant with the University of California San Diego and San Diego State University, since 2021. He was a recipient of grants from the NSF and the state of Indiana. His research interests include signal and image processing, machine learning, deep learning, time series analysis, and biomedical engineering.

RAMESH R. RAO (Fellow, IEEE) received the bachelor's degree (Hons.) in electrical and electronics engineering from the University of Madras, India, in 1980, and the M.S. and Ph.D. degrees from the University of Maryland, College Park, MD, USA, in 1982 and 1984, respectively. Since 1984, he has been with the Faculty of the Department of Electrical and Computer Engineering, University of California San Diego, where he is currently a Professor and the Director of the Qualcomm Institute. His research interests include architectures, protocols, and performance analysis of wireless, wireline, and photonic networks for integrated multimedia services. He was elected to the IEEE Information Theory Society Board of Governors, in 1997 and 2000. He regularly serves as a member for the technical program committees of IEEE conferences and as a Reviewer for agencies, such as the National Science Foundation. He has served as the Editor for the Information Theory Society Newsletter, from 1993 to 1995. He is the Founding Web Editor of the Information Theory Society website. He is an Editor of packet multiple access of IEEE TRANSACTIONS ON COMMUNICATIONS. He is an Editorial Board Member of the *ACM/Baltzer Wireless Networks* and *IEEE Network Magazine*. He has been a guest editor of special issues of several ACM and IEEE journals.

CHRISTOPHER PAOLINI (Member, IEEE) received the B.S. and M.S. degrees in computer science and the Ph.D. degree in computational science from SDSU, in 1991, 1998, and 2007, respectively. He is an Assistant Professor with the Department of Electrical and Computer Engineering, SDSU, where he advises graduate students. He was a recipient of grants from the Department of Energy and NASA and five NSF Office of CyberInfrastructure Awards. His current research interests include the Internet of Things device development, machine learning, embedded systems, cloud computing, big data analytics, deep learning, software engineering, numerical chemical thermodynamics, numerical chemical kinetics, numerical geochemistry, high-performance computing, scientific computing and numerical modeling, high speed (100gbps) networking, cyberinfrastructure development, and cybersecurity.

MAHASWETA SARKAR (Member, IEEE) received the Ph.D. degree from the University of California San Diego, in January 2006. She is a Professor with the Department of Electrical and Computer Engineering, San Diego State University (SDSU). After a brief stint as a Senior Scientist with the SPAWAR Laboratory, Point Loma, she joined SDSU, as a tenure-track Faculty Member, in August 2006. She is currently the Senior Associate Dean with the Global Campus (formerly the College of Extended Studies), SDSU. Her research interests include wireless networks, especially developing artificial intelligence infused MAC layer algorithms for applications ranging from precision agriculture and healthcare to brain–computer interfaces. She has published over 100 research articles (with several best-paper awards) and one patent. She was a recipient of the President's Leadership Award at SDSU, in 2010; the Faculty Excellence Award in Diversity and Inclusion, in 2021; and the Distinguished Faculty Award, in 2023–2024. She received the fellowship under the Fulbright Specialist Program, from 2021 to 2024.