Nigerian Journal of Technology (NIJOTECH)
Vol. 45, No. 1, March 2026, pp. 192- 203
www.nijotech.com

Print ISSN: 0331-8443
Electronic ISSN: 2467-8821
https://doi.org/10.4314/njt.2026.5675

EXTRACTION OF MULTIPLE PHYSIOLOGICAL PARAMETERS FROM A SINGLE-CHANNEL PHOTOPLETHYSMOGRAPHIC WAVEFORM USING MACHINE LEARNING

AUTHORS:
*A. A. Amusan, T. S. Ajani and A. L. Imoize

AFFILIATIONS:
Department of Electrical & Electronics Engineering, University of Lagos, NIGERIA,

***CORRESPONDING AUTHOR:**
Email: aamusan@unilag.edu.ng

ARTICLE HISTORY:
Received: September 27, 2025.
Revised: February 11, 2026.
Accepted: February 19, 2026.
Published: April 29, 2026.

KEYWORDS:
Cardiovascular health, k-fold cross-validation, Machine learning, Photoplethysmogram, Physiological parameters.

ARTICLE INCLUDES:
Peer review

DATA AVAILABILITY:
On request from author(s)

EDITORS:
Ozoemena Ani

FUNDING:
None

Abstract

In today's world, cardiovascular health challenges constitute the foremost causes of death due to a lack of timely medical attention. This concerns the present non-continuous, uncomfortable, and expensive clinical techniques for assessing cardiovascular health. Meanwhile, photoplethysmographic (PPG) signals contain valuable information concerning the human cardiovascular system. They may be exploited to extract physiological parameters, which can provide simple, convenient, and continuous monitoring of cardiovascular health. Although, some works have attempted to determine crucial physiological parameters from the photoplethysmogram, they do not extract these parameters simultaneously with high estimation accuracy. Consequently, this work performs a pulse wave analysis of the photoplethysmogram using four machine-learning (ML) techniques (XGBoost, feedforward ANN, RNN, and Bi-LSTM) to simultaneously estimate key physiological parameters (heart rate (HR) and blood pressure (BP)) to monitor the cardiovascular system. The ML models are developed and evaluated using a publicly available dataset, which contains a handcrafted 46-feature dataset extracted from a 3D input consisting of the photoplethysmogram (PPG), velocity photoplethysmogram (VPG), and acceleration photoplethysmogram (APG) signals. The performance evaluation of the machine learning models was implemented using k-fold cross-validation and typical error indices, such as mean absolute error (MAE), mean squared error (MSE), and root mean square error (RMSE) visualized through cross-validation plots, test set metric plots, and parity plots. This study revealed that the ML models achieved a mean absolute error (MAE) of less than 0.5 for heart rate, and less than 1 for diastolic, systolic, and arterial blood pressures, respectively, outperforming existing research.

1.0 INTRODUCTION

Statistics from the World Health Organization (WHO) reveal that cardiovascular diseases are among the foremost causes of high death rates globally [1]. Early detection of these diseases could inform appropriate and timely clinical intervention, which could have prevented the deaths of the victims. However, gold standard methods such as cuff-based blood pressure monitors and pulse oximeters, which may sometimes require the patient to be physically present at a hospital, make the continuous monitoring of the cardiovascular system difficult [2]. Moreover, cuff-based techniques are not simple, comfortable, and easy to implement by users outside a clinical setting [3]. Meanwhile, recent years have seen the emergence of biological signals (electrocardiograms (ECGs) and

HOW TO CITE:
Amusan, A. A., Ajani, T. S. and Imoize, A. L. "Extraction of Multiple Physiological Parameters From a Single-Channel Photoplethysmographic Waveform using Machine Learning", *Nigerian Journal of Technology*, 2026. 45(1), pp. 192- 203. https://doi.org/10.4314/njt.2026.5675

photoplethysmograms (PPGs)), which contain information about the cardiovascular system [4]. Photoplethysmography involves measuring blood volumetric changes induced by the cardiac cycle in the human body. To obtain the photoplethysmogram, the skin is illuminated with an optical source (often generated from a light-emitting diode), while small variations in the reflected or transmitted light intensity are measured using a photodetector. These variations relate to changes in blood volume, producing a waveform that may provide significant information concerning the human cardiovascular system. These signals reflect the physiological changes in the human body and may be analyzed to continuously observe the condition of the cardiovascular system, notifying the patient of any abnormality. Precisely, important features may be extracted that correspond to critical physiological parameters like heart rate, blood pressure, respiratory rate, oxygen saturation, and heart rate variability, which model the state of the cardiovascular system [5].

Although ECG-based methods accurately estimate physiological parameters, they only apply in clinical environments. Moreover, they are not easy and comfortable to implement since they require an electrode attached to the human body, which can cause irritation, drying, or discomfort over time [6]. Meanwhile, PPG-based methods are simple, convenient, easy, and economical when utilized and may be employed outside clinical environments [7].

Although the PPG waveform contains rich information to extract features to observe the physiological parameters for valuable insight into the human cardiovascular system [8], precise parameter extraction is crucial. PPG signal analysis can be impaired by noise and motion artifacts resulting to measurement errors. These errors can be mitigated by using a hybrid approach, such as machine learning for parameter extraction, due to the possibility of adaptive learning beyond handcrafted features. Further description of important PPG concept/terminology is given in Table 1.

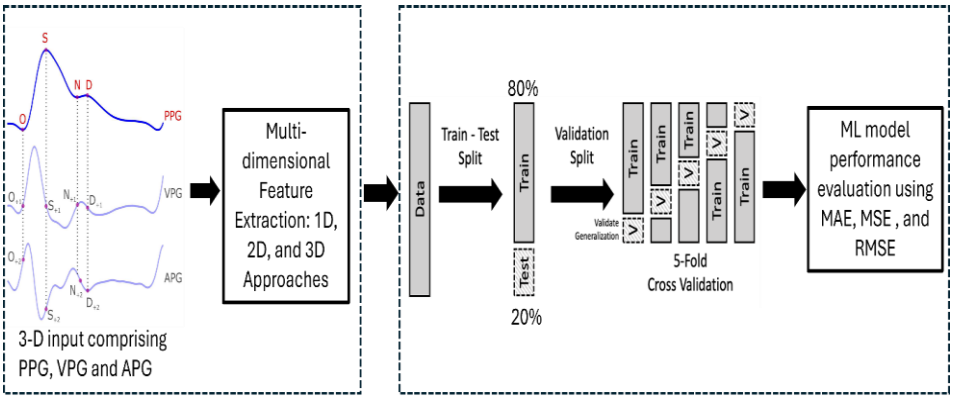

Figure 1: Block diagram showing the different stages of the methodology adopted in this work

We explored existing research works that have understudied the PPG waveform and used it to extract significant physiological parameters, with the contributions and limitations of each work. For example, the authors in [11] developed a compact device to measure multiple physiological parameters (hematocrit, SpO_2, pulse, and respiration rate) using PPG, and blood pressure using oscillometric methods. They employed multiple linear regression and digital filtering techniques. However, the device was not wearable, supported only one-time measurement, and used a cuff for BP, not PPG. Another work in [12] used a support vector machine (SVM) to estimate blood pressure from PPG data (University of Queensland dataset). Preprocessing involved low-pass filtering, with 75% of the data used for training. It only estimated blood pressure and did not incorporate other physiological parameters. Furthermore, the authors in [13] measured heart rate and respiratory rate using bandpass filtering and DFT-based pulse wave analysis. Specific frequency ranges were used for filtering. Blood pressure was not measured. Existing works mainly focused on extracting a single physiological parameter, such as cardiac output, heart rate, blood pressure, and glucose monitoring from the PPG waveform [14] - [17]. However, multiple physiological parameters may be extracted simultaneously to provide robust insight into cardiovascular health, thereby detecting possible abnormalities [9], [18]. Thus, this research focuses on the pulse wave analysis of the PPG waveform to simultaneously estimate critical physiological parameters and monitor cardiovascular health.

https://doi.org/10.4314/njt.2026.5675

Consequently, this research simultaneously extracts four parameters from the Photoplethysmogram using ML methods. In addition, this study improves the work performed in [9], which is the baseline study. To do this, a robust dataset containing handcrafted features was obtained from a 3-D input comprising the PPG, VPG, and APG, as recorded in the publicly available MIMIC-III dataset. Then, the dataset was prepared using adequate data preprocessing techniques involving data reduction and normalization. The data reduction step comprises the computation of the root mean squared value of each waveform extracted over a period of 8 seconds to reduce the dataset complexity, while the data normalization step involves the downscaling of the dataset to reduce computation complexity. Subsequently, we employed 80% of the prepared dataset to develop and train selected machine learning models which are extreme gradient boosting (XGBoost), feedforward artificial neural network (ANN), recurrent neural network (RNN), and bidirectional long short-term memory (Bi-LSTM), to compare the performance of the shallow model regressor (XGBoost) with the deep model regressors (ANN, RNN and Bi-LSTM). Then, the trained models were tested by deploying them to simultaneously estimate selected parameters such as the heart rate (HR), arterial blood pressure (ABP), systolic blood pressure (SBP), and diastolic blood pressure (DBP) using the remaining 20% of the prepared dataset. Moreover, the prediction performance of the developed models was evaluated on each physiological parameter using k-fold cross-validation and test set metrics, including mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE), with results visualized through cross-validation plots, test set metric plots, and parity plots.

Table 1: A comprehensive description of important concepts on photoplethysmography

Important Concepts	Description
Measuring Site	The PPG signal quality highly depends on the site from which the measurement is taken in the body [9]. Particularly, the signal quality, shape, and robustness to motion artifacts are influenced by the measuring location in the body. Some known body sites from which PPG measurement may be taken include the fingertip, wrist, earlobe, forehead, eyes, etc. Among these body sites, PPG signals measured from the fingertip provide the most analyzable waveforms [10]. Consequently, this study focuses on the pulse wave analysis of PPG signals measured from the fingertip.
Modes of Measurement	As mentioned earlier, the Photoplethysmographic waveform is produced by the variations of the light intensity moving from the source (LED) to the photodetector. The path of light from the emitter to the receiver determines the mode of operation. In relation to how the emitter and receiver are placed, there are two approaches to measuring PPG signals from the human body: the transmissive and reflective modes [1].
The PPG Waveform	The photoplethysmogram (PPG) waveform can be segmented into two main sections, namely, anacrotic and catacrotic phase/section. The anacrotic region corresponds to the rising edge of the waveform, whereas the catacrotic region represents the falling edge. Precisely, the first section (anacrotic segment) often models the systolic phase, while the second section (catacrotic segment) describes the cardiac cycle diastole. In the systolic phase, the heart contracts pushing oxygenated blood through the blood vessels to the peripheral parts of the body, while in the diastolic phase, the heart retracts to suck in deoxygenated blood. Certain information features may be observed and identified from the PPG waveform, such as systolic amplitude, pulse width, pulse area, peak-to-peak interval, pulse interval, etc. In addition, the first- and second-time derivatives of the PPG signal – VPG and APG, have been discovered to provide good features for accurately estimating blood pressure.
Physiological Parameters	heart rate, heart rate variability, respiratory rate, blood pressure, and blood oxygen saturation.

In summary, this work extracts four physiological parameters from the PPG signal using a machine learning technique and subsequently compares the performance of selected shallow and deep learning models. The remaining section of the manuscript is organized as follows. Section 2 describes the methodology used, while the results obtained are

https://doi.org/10.4314/njt.2026.5675
http://creativecommons.org/licenses/by-nc-nd/4.0/

presented and discussed in Section 3. The conclusion and recommendation are presented in Section 4.

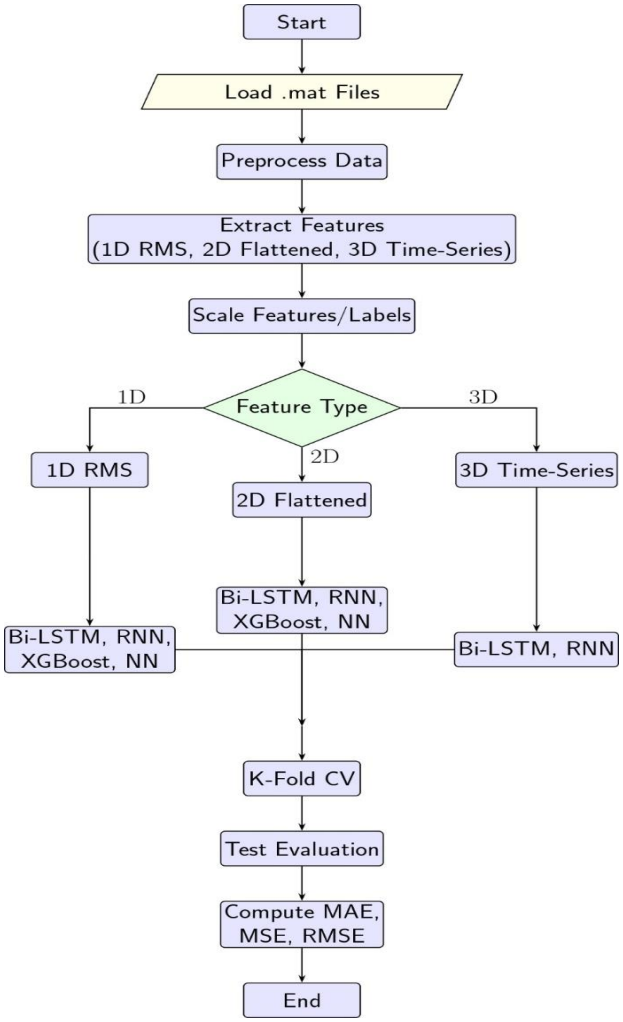

Figure 2: Flow chart showing the different stages of the methodology adopted in this work

2.0 METHODOLOGY

The methodology adopted in this work is illustrated in the block diagram in Figure 1 and the flowchart in Figure 2. This comprises four stages: dataset collection, dataset preparation, model training and testing, and model performance evaluation. Dataset collection involves loading the PPG signals and corresponding labels (HR, ABP, SBP, DBP) from .mat files, with caching to optimize processing. Dataset preparation includes computing handcrafted features, 1D root mean square (RMS), 2D flattened features for XGBoost and ANN, while retaining 3D time-series data for RNN and Bi-LSTM models, followed by scaling using StandardScaler. Model training and testing were performed using XGBoost and ANN with a MultiOutputRegressor for 1D handcrafted features or 2D data, and RNN and Bi-LSTM models for 1D, 2D or 3D data, using 80% of the dataset for training and 20% for testing to predict four-dimensional vectors (HR, ABP, SBP, DBP). Model performance evaluation assesses all models using k-fold cross-validation and test set metrics (MAE, MSE, RMSE), with results visualized through cross-validation plots, test set metric plots, and parity plots. Figure 1 shows the pictorial description of this methodological approach, illustrating dataset collection, dataset preparation, machine learning model training, testing, and performance evaluation.

2.1 Database

The database employed in this study was collected from an existing work by Kanoga et al. [9]. This dataset is publicly available and can be accessed via the Google link provided by the author in reference [19]. The dataset contains cleaned, segmented, and preprocessed data from the MIMIC-III waveform database matched subset database series, which contains thousands of recordings of multiple biosignals and time series of vital signs. Specifically, a handcrafted 46-feature dataset of 1811 subjects extracted from a three-dimensional (3D) data input containing photoplethysmograms (PPGs), velocity photoplethysmograms (VPGs), and acceleration photoplethysmograms (APGs) with a length of 1000 values were employed. These handcrafted features are mapped to data labels including HR, ABP, SBP, and DBP, averaged over an 8-second window to obtain numeric records. The dataset is organized into three categories or cases – 1st category (subject independent), 2nd category (subject dependent), and 3rd category (complete subject dependent). In this work, we adopted the 3rd category owing to the reduced size because of the available training samples in the dataset (620,382) compared to case 1 (1,514,633) and case 2 (2,135,015). Specifically, we divide the dataset into two segments – the training dataset (80% | 1453 subjects) and the testing dataset (20% | 358 subjects).

2.2 Machine Learning Techniques and Feature Engineering

In this study, the impact of feature engineering on the ML model performance is investigated for the simultaneous estimation of cardiovascular variables (HR, BP). Two classes of models are employed: shallow regressors and deep learning architecture. The shallow model used is XGBoost, while the deep learning models include ANN, RNN, and Bi-LSTM networks. A key focus of this work is a systematic exploration of different feature representations, aimed at understanding how the choice of features

https://doi.org/10.4314/njt.2026.5675

influences model performance. Three feature formats are considered:

1. 1D root mean square (RMS) features: These summarize the signal energy in each channel and serve as compact statistical descriptors. This was performed by computing the RMS of each waveform as opposed to the average value in the baseline method [15]
2. 2D flattened features: Represent spatial and temporal patterns in a flattened matrix format, preserving more local variability than RMS features.
3. 3D structured features: Retain the full temporal and spatial structure of the original signals, allowing models like RNN and Bi-LSTM to learn from sequential dependencies and contextual information.

The XGBoost and Feedforward Neural Network models were evaluated using 1D and 2D features, while the RNN and Bi-LSTM models were additionally assessed using 3D features to exploit their temporal modeling capabilities. While complex feature engineering can enhance performance, its effectiveness depends on the alignment between the feature structure and the model's inductive bias; hence, by exploring different feature representations and dimensional reduction, we can draw some insights into how we can use AI effectively to represent some of the physics in humans.

2.3 Materials and Simulation Environment

All computational experiments were performed on the Stampede3 supercomputer at the Texas Advanced Computing Center (TACC) (details shown in Table 2). We utilized two compute nodes, each equipped with an Intel Xeon Phi 7250 (Knights Landing) processor containing 68 physical cores, supporting up to 4 hardware threads per core. This configuration provided a total of 224 logical. Each node offered 96 GB of high-bandwidth MCDRAM and 96 GB of DDR4 RAM, totaling 384 GB of available memory across the two nodes. Nodes were interconnected via Intel Omni-Path Architecture (OPA) to ensure high-speed, low-latency communication for parallel computations. Jobs were submitted and executed using the Slurm Parallel Runtime (SPR) environment, leveraging MPI-based parallelization for efficient scaling of compute-intensive simulations. The program code used in this research was written in the Python programming language, and development was carried out using Visual Studio Code (VS Code), an integrated development environment (IDE) developed by Microsoft. All compute resources were accessed remotely via secure connections through VS Code's Remote SSH extension, allowing seamless code development and execution on Stampede3 nodes.

Table 2: Computational environment and simulation setup

Category	**Details**
Computing Platform	Stampede3 Supercomputer, Texas Advanced Computing Center (TACC)
Compute Nodes Used	2 nodes
Processor Type	Intel Xeon Phi 7250 (Knights Landing)
Cores per Node	68 physical cores (64 used for computation)
Hardware Threads per Core	Up to 4
Total Logical Processors	224 (2 nodes × 64 cores × 2 threads)
Memory per Node	96 GB MCDRAM + 96 GB DDR4 RAM
Total Available Memory	384 GB (192 GB per node × 2 nodes)
Interconnect	Intel Omni-Path Architecture (OPA)
Job Scheduler	Slurm Parallel Runtime (SPR)
Parallelization Method	MPI-based
Programming Language	Python
Development Environment	Visual Studio Code (VS Code)
Simulation & ML Environment	Jupyter Notebook
Machine Learning Libraries	- numpy (numerical computation) - pandas (data manipulation) - h5py (HDF5 access) - xgboost (gradient boosting) - scikit-learn (classical ML) - tensorflow (neural networks)

3.0 RESULTS AND DISCUSSION

https://doi.org/10.4314/njt.2026.5675

The results obtained are presented highlighting comparisons with existing approaches and a baseline reference. The performance was quantitatively assessed and benchmarked through established metrics such as MAE, MSE, and RMSE. The analysis highlights the improvements achieved by our approach relative to prior studies and demonstrates its superior predictive accuracy and robustness.

3.1 Extreme Gradient Boosting (XGBoost) Model

The shallow model that was trained and deployed is the XGBoost regressor. XGBoost is a robust gradient-boosting model based on a sequential ensemble of decision trees to minimize prediction errors and with inbuilt regularization to prevent overfitting. The model is configured for regression tasks with a squared error loss function (`*reg:squarederror*`), 100 boosting rounds, and RMSE as the evaluation metric. It was wrapped in `*MultiOutputRegressor*` to extend its function to handle multi-target regression by fitting different regressors for each target variable. The model was trained using scaled input features and target variables, ensuring consistent feature scaling for improved performance and accuracy (see Table 3 for details).

Table 3: Hyperparameters and configuration values used for the development of the XG-boost model

Hyperparameters/Configuration	Value
Learning Rate	0.1
Maximum Depth	6
Number of Estimators	100
Subsample	0.8
Colsample by Tree	0.8
Regularization (λ)	1.0
Regularization (α)	0.0
Objective	Squared Error
Booster	Gradient Boosted Tree
Minimum Child Weight	1

The hyperparameters of the XGBoost model were selected based on a combination of cross-validation and computational considerations. We employed k-fold cross-validation to assess model stability and generalization performance. Given the large size of the dataset and the substantial training time required, extensive hyperparameter optimization (e.g., grid or Bayesian search) was not pursued. We started from the default configuration and tuned minimally such as lowering the learning rate to 0.1, as well as subsample and column subsample to 0.8, to improve randomness, prevent overfitting and ultimately improve the model stability and generalization. This initial model configuration already demonstrated robust and stable performance across folds, with strong predictive accuracy and minimal variance. Further hyperparameter tuning did not yield meaningful performance gains relative to the increased computational cost.

3.2 Feedforward Artificial Neural Network (ANN)

The feedforward ANN is a brain-inspired model where data flows in one direction from the input layer through one or more hidden layers to the output layer without any feedback connections.

The feedforward ANN model was deployed using TensorFlow and Keras, while the model was built using the `Sequential` class, allowing stacking layers sequentially. The network has multiple dense layers, with each layer followed by batch normalization and dropout. The first dense layer has 128 neurons and inputs shaped like the scaled input of `X`, where X denotes the training data, which contains the input feature matrix. This is followed by two dense layers with 64 and 32 neurons, respectively. The hidden (dense) layers applied a rectified linear activation function (ReLU) and L2 regularization with a penalty factor of 0.01 and batch normalization for stabilizing the training. Dropout was applied at a rate of 0.3 to avoid overfitting. The output layer consists of 4 neurons with a linear activation function, which is suitable for regression tasks with continuous output values. The model was then compiled using the Adam optimizer and mean squared error as the loss function, which is typical for regression problems (See Table 4 for more details). An EarlyStopping

https://doi.org/10.4314/njt.2026.5675

callback is implemented to monitor the validation loss and stop training if it does not improve for three consecutive epochs (`patience=5`), helping to prevent overfitting. The model is then trained using the `fit` method, with the scaled input and scaled target values. The training is split into a validation set using 20% of the data. The training runs for a maximum of 50 epochs, with a batch size of 32. The early stopping ensures that the best model, based on validation performance, is retained.

Table 4: Hyperparameters and configurations for ANN, RNN, and Bi-LSTM model

Hyperparameters/Configuration	ANN	RNN	Bi-LSTM
Number of Hidden Layers	3	3	3
Hidden Units per Layer	128, 64, 32	64, 32, 16	64, 32, 16
Input Shape	(3,)	(1,3)	(1,3)
Activation Function	ReLU	ReLU	Tanh
Learning Rate	0.001	0.001	0.001
Optimizer	Adam	Adam	Adam
Batch Size	32	32	32
Epochs	100	50	50
Dropout Rate	0.3	0.2	0.2
Loss Function	Mean Squared Error	Mean Squared Error	Mean Squared Error
Initializer	He Normal	Nil	Nil
Recurrent Unit Type	Nil	Nil	Bi-LSTM
Merge Mode	Nil	Nil	Concatenate

3.3 Recurrent Neural Networks (RNN)

Unlike the feedforward ANN, the RNN has a feedback path from the hidden layer to the inputs, so that the output depends not only on the current inputs but also on previous hidden states. To facilitate a more comprehensive and robust comparison across machine learning models, a standard RNN is employed to train and predict the four key physiological parameters. The RNN architecture, characterized by its recurrent connections, allows information to persist over time steps, making it suitable for modeling sequential data (See Table 4). Unlike more advanced variants such as LSTM or Bi-LSTM, the standard RNN operates with a simpler structure and shorter memory, serving as a useful baseline. Prior to model training, the temporal signal data is compressed by computing the RMS value over time, effectively reducing the dataset to a single representative value per signal. The predictive performance of the RNN is evaluated using MAE, MSE, and RMSE, ensuring consistent performance assessment across all model types.

3.4 Bidirectional Long Short-Term Memory (Bi-LSTM)

The Bi-LSTM network is a specialized RNN architecture that propagates the input sequences in both forward and backward directions, allowing the network to model contextual dependencies from both past and future steps. This bidirectional structure enhances the model's ability to learn temporal patterns in physiological signals, particularly when current states are influenced by both prior and subsequent values. The predictive performance of the Bi-LSTM model is quantitatively assessed using standard regression metrics: MAE, MSE, and RMSE, providing a detailed measurement of its accuracy and robustness across different signal representations.

3.5 Performance Evaluation using MAE, MSE, and RMSE Metrics

This work utilizes performance evaluation metrics such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE), to evaluate the developed machine learning models. These evaluation metrics provide an intuitive comparison between the predicted value and the true value. The MAE is computed as the average of the absolute errors between the predicted and true value, while the MSE is calculated as the average of the squared difference between the predicted and true value. The RMSE is computed by taking the square root of MSE. These metrics further allow for direct comparison of the model's performance with that of the baseline. To ensure the generalizability of our model, we did a five-fold cross-validation with shuffling (random_state=42) to evaluate the model's performance on the training data. We also ensure that

https://doi.org/10.4314/njt.2026.5675

the distribution of samples within the training and test datasets accurately represents the physiological space of interest, as seen in Figure 3.

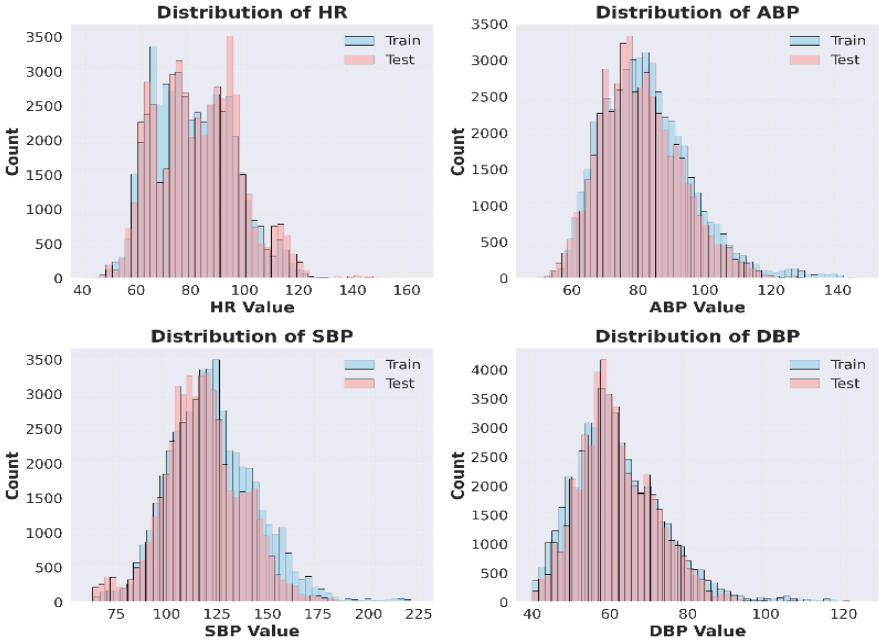

Figure 3: Distribution of data across the training and testing sets, demonstrating coverage of the relevant physiological target space to ensure representative and unbiased model evaluation.

The five-fold cross-validation is a method in which the original dataset is randomly divided into five approximately evenly sized subsets or folds. The model training and evaluation process is then repeated five times, at every instance with a unique data subset for validation and the remaining four data subsets for training. This systematic rotation guarantees that each data point is used exactly once for validation and four times for training, minimizing both bias and variance associated with a single random train-test split. Low variance in metrics (MAE, MSE, RMSE) across folds (indicated by small standard deviations) suggests that the model is stable and generalizes well. High variance may indicate overfitting or sensitivity to data splits. We also evaluated the performance of Bi-LSTM and RNN models using different feature representations across 1D, 2D, and 3D formats. Our results indicate that, while deeper models generally benefit from richer feature representations, particularly those that preserve temporal structure, the performance gains are not always substantial in comparison to the additional computational cost associated with complexity of the feature space, as shown in Figure 4 and 5.

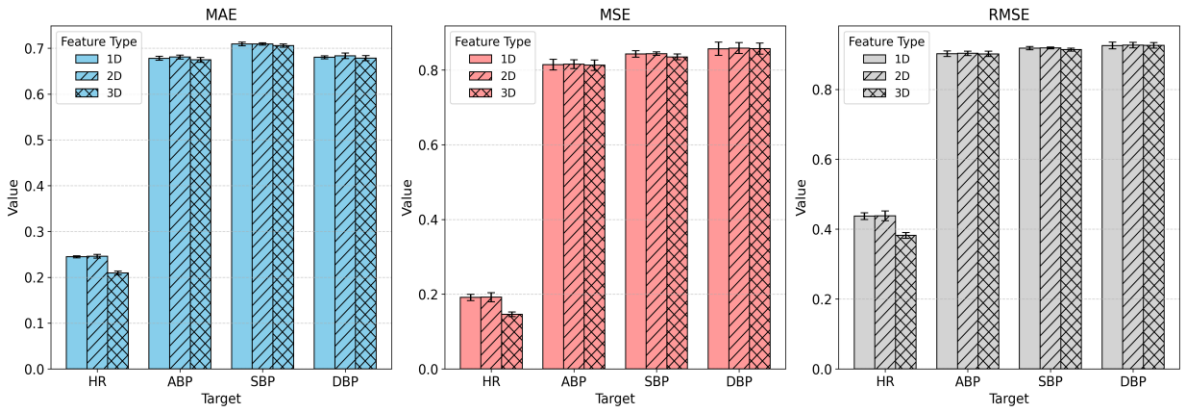

Figure 4: Performance of Bi-LSTM in predicting physiological parameters using different feature types.

https://doi.org/10.4314/njt.2026.5675
http://creativecommons.org/licenses/by-nc-nd/4.0/

Given that model performance does not significantly improve with more complex feature spaces, we present the performance metrics based on 1D features in Table 5. This also enables a fair comparison with existing models in the literature. Table 5 presents the mean performance across the five cross-validation folds, while Table 6 reports the standard deviation, highlighting the variability in performance across different models. Comparing the MAE of this work to the baseline study, we find that the XGBoost model in this study outperforms that in existing baseline research.

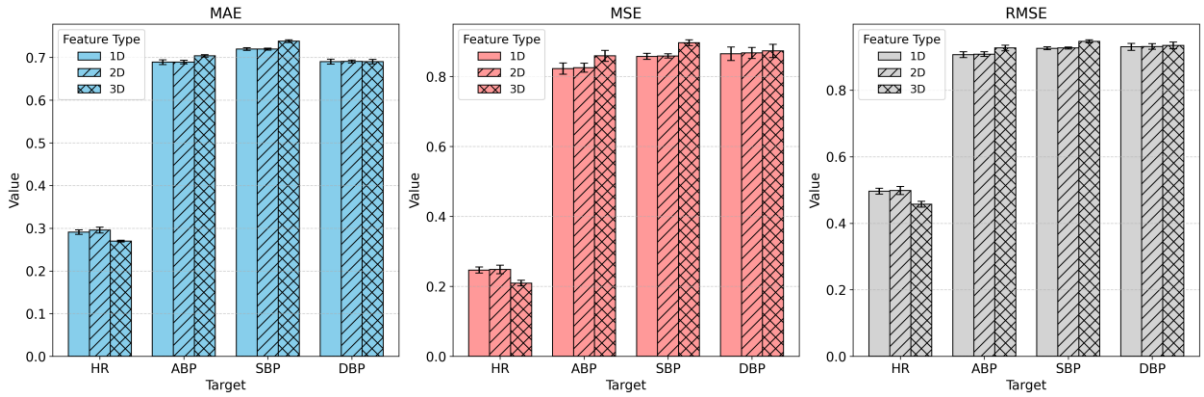

Figure 5: Performance of RNN in predicting physiological parameters using different feature types.

Table 5: Summary of training performance across different machine learning algorithms, evaluated using cross-validation.

Model	HR			ABP			SBP			DBP		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
XGBoost	0.30	0.22	0.47	0.67	0.79	0.89	0.71	0.85	0.91	0.68	0.85	0.92
ANN	0.29	0.23	0.48	0.73	0.90	0.95	0.75	0.92	0.96	0.71	0.91	0.95
RNN	0.27	0.21	0.46	0.70	0.86	0.93	0.74	0.90	0.95	0.69	0.87	0.93
Bi-LSTM	0.25	0.19	0.44	0.68	0.81	0.90	0.71	0.84	0.92	0.86	0.92	0.93

Table 6: Statistical variation of training results (represented as the standard deviation across five cross folds validation)

Model	HR			ABP			SBP			DBP		
	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE
XGBoost	0.0030	0.0068	0.0072	0.0038	0.011	0.0064	0.0022	0.0069	0.0038	0.0041	0.015	0.0080
ANN	0.0084	0.014	0.015	0.0046	0.014	0.0076	0.0051	0.0093	0.0048	0.0047	0.0019	
RNN	0.0020	0.0081	0.0088	0.0028	0.016	0.0086	0.0031	0.0083	0.0044	0.0053	0.019	0.010
Bi-LSTM	0.0023	0.0084	0.0096	0.0043	0.014	0.0080	0.0040	0.0088	0.0048	0.0034	0.018	0.0095

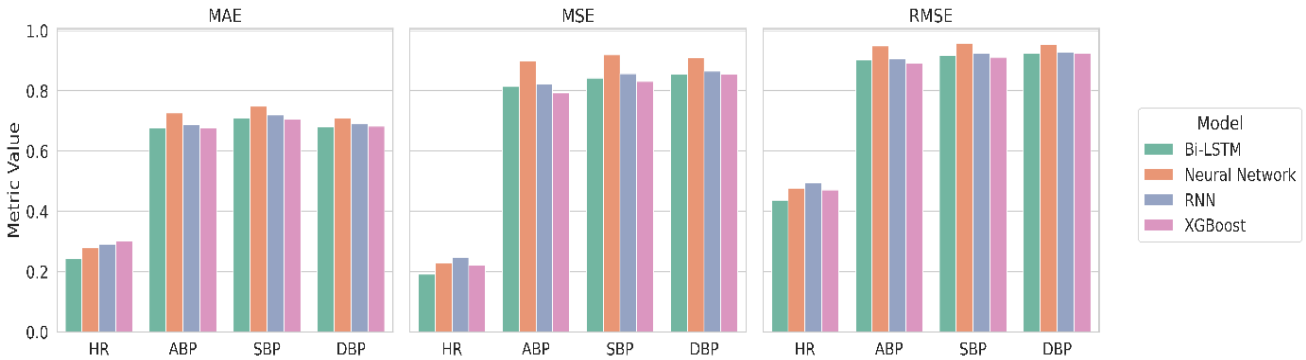

Figure 6: Comparison of the cross-fold validation mean training errors across different models

3.6. Discussion

https://doi.org/10.4314/njt.2026.5675
http://creativecommons.org/licenses/by-nc-nd/4.0/

The MAE for the test dataset is presented in Table 7, comparing the models in this study with the baseline study. The results reveal that our approach gave lower MAE values across all physiological parameters (HR, ABP, SBP, DBP), outperforming the baseline. This improvement stems from two key techniques. First, unlike the baseline study, which averages feature vectors to reduce the dataset to a single value per instance, our method computes the root mean square (RMS) of PPG, VPG, and APG signals. This RMS-based feature extraction, which captures signal energy more effectively, enhances model estimation accuracy. Second, we applied standard scaling to both input features and target variables using standard scaling, ensuring zero mean and unit variance. This consistent scaling improves model performance by normalizing data distribution. Additionally, Table 7 shows that the XGBoost model achieves lower MAE values compared to the deep learning models. This is attributed to the dataset size and computational constraints. Deep learning models typically excel with large datasets, whereas shallow models like XGBoost perform better with smaller, reduced datasets. By using a reduced dataset to lower data complexity and computational demand, our approach targets edge devices with limited resources, making it well-suited for practical deployment.

Table 7: Comparison of the MAE obtained from the test dataset between this work and the baseline study

Model	HR	ABP	SBP	DBP
XGBoost (This study)	0.37	0.61	0.68	0.58
ANN (This study)	0.32	0.65	0.73	0.58
RNN (This study)	0.36	0.64	0.67	0.57
Bi-LSTM (This study)	0.28	0.62	0.68	0.56
XGBoost (Baseline Study) [9]	2.42	-	8.58	4.62
Random Forest (RF) [9]	2.42	-	8.81	4.73
MLR [9]	4.22	-	15.46	7.84
Generalized Regression Neural Network [20]	-	-	3.96	2.39
Temporal Convolutional Network [10]	-	-	2.38	1.23
Modified Long-term Recurrent Convolutional Network [12]	1.137	-	2.942	1.747
Adaptive Boosting (AdaBoost) [13]	-	-	11.17	5.35
Transfer Learning [21]	-	-	3.52	2.20

3.7 Limitations

The dataset used in this study was intentionally designed to be diverse, encompassing signals from subjects with varying skin tones, ages, genders, and other demographic characteristics. While this diversity enhances the generalizability of the models, it also introduces variability in the quality of PPG, VPG, and APG signals. This variability stems from physiological differences such as skin pigmentation and vascular properties, which can affect the signal-to-noise ratio and, in turn, the reliability of extracted features. These factors may influence model performance across different target variables (HR, ABP, SBP, DBP). Future work could investigate hybrid approaches that combine lightweight feature extraction with end-to-end deep learning, aiming to optimize performance on edge devices while leveraging larger, demographically rich datasets.

CONCLUSION

In this work, the simultaneous estimation of physiological parameters from a single-channel PPG signal using ML techniques has been presented. Physiological parameters, including HR, ABP, SBP, and DBP were extracted using shallow models (XGBoost) and deep learning models (feedforward ANN, RNN, and Bi-LSTM). To achieve this, a clean, segmented, and preprocessed dataset containing 46 handcrafted features from 1811 subjects, recorded over an 8-second window was collected. It comprises several thousand sessions of multiple bio signals and vital sign time series. The dataset's complexity was reduced by generating a single value for the handcrafted 46-feature vector using the root mean square computation for each instance. The results reveal that the developed models gave lower MAE, values across all physiological parameters (HR, ABP, SBP, DBP), outperforming the baseline. Specifically, both the shallow (XGBoost) and deep learning models (ANN, RNN and BiLSTM) achieved a mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE) of less than 0.5 for HR, and less than 1 for ABP, SBP, and DBP, respectively, outperforming existing research. This

https://doi.org/10.4314/njt.2026.5675

improvement can be mainly attributed to the computation of the root mean square (RMS) of feature vectors in the PPG, VPG, and APG signal for dataset reduction as compared to averaging in the previous work. Also, the use of scaling to input features and target variables improves model performance. Models were also developed for 2D and 3D feature spaces, but no drastic change in performance was observed as the input is scaled to 2D and 3D. In addition, the XGBoost model generally slightly outperforms the deep learning models owing to the nature of the dataset employed, since shallow models are known to typically excel when using a smaller dataset compared to deep learning models.

Future work involves developing a wearable device capable of simultaneously estimating key physiological parameters (e.g., HR, BP, etc.) from real-time PPG signals using machine learning models. To achieve this, specialized hardware with sufficient computational resources to process ML models in real time needs to be designed and developed. In addition, this will enable the development of personalized models tailored to individual needs, using real-time PPG data for optimal performance.

REFERENCES

[1] Elgendi, M. *PPG Signal Analysis: An Introduction Using MATLAB*, Boca Raton, FL: CRC Press, 2021.

[2] Kumar, S., Yadav, S., and Kumar, A. "Blood pressure measurement techniques, standards, technologies, and the latest futuristic wearable cuff-less know-how," *Sensors and Diagnostics*, 3 (2), pp. 181–202, 2024. doi: 10.1039/d3sd00201b.

[3] Ding, X.-R., Zhang, Y.-T., Liu, J., and Dai, W.-X "Continuous cuffless blood pressure estimation using pulse transit time and photoplethysmogram intensity ratio," *IEEE Transactions on Biomedical Engineering*, 63 (5), pp. 964–972, 2016. doi: 10.1109/TBME.2015.2480679.

[4] Fuadah, Y. N., and Lim, K. M. "Advances in cardiovascular signal analysis with future directions: a review of ML and DL models for cardiovascular disease classification based on ECG, PCG, and PPG signals," *Biomedical Engineering Letters*, 15 (4), pp. 619-660, 2025. doi: 10.1007/s13534-025-00473-9.

[5] Shuzan, M. N. I., Chowdhury, M. H., Chowdhury, M. E., Murugappan, M., Bhuiyan, E. H., Ayari, M. A., and Khandakar, A. "Machine learning – based respiration rate and blood oxygen saturation estimation using photoplethysmogram signals," *Bioengineering*, 10 (2), article 167, pp. 1–15, 2023. doi: 10.3390/bioengineering10020167.

[6] Kang, T. W., Lee, J., Kwon, Y., Kim, J., Lee, H., Kim, S., Sim, J. Y., and Yeo, W.-H. "Recent progress in the development of flexible wearable electrodes for electrocardiogram monitoring during exercise," *Advanced NanoBiomed Research*, 4 (8), 2300169, pp. 1–25, 2024. doi: 10.1002/anbr.202300169.

[7] Allen, J. "Photoplethysmography and its application in clinical physiological measurement," *Physiological Measurement*, 28 (3), pp. 1–39, 2007. doi: 10.1088/0967-3334/28/3/R01.

[8] Jayroop, R., Zahra, S., Raafat, A., and Assim, S. "Atrial fibrillation classification with smart wearables using short-term heart rate variability and deep convolutional neural networks," *Sensors*, 21(21), 7233, pp. 1–24, 2021. doi: 10.3390/s21217233.

[9] Kanoga, S., Hoshino, T., Kamei, S., Kobayashi, T., Ohmori, T., Uchiyama, M., and Tada, M. "Comparison of seven shallow and deep regressors in continuous blood pressure and heart rate estimation using single-channel photoplethysmograms under three evaluation cases," *Biomedical Signal Processing and Control*, 85, article 105029, pp. 1–18, 2023. doi: 10.1016/j.bspc.2023.105029.

[10] Monalisa, S. R., Rajarshi, G., and Kaushik, D. S. "BePCon: A photoplethysmography-based quality-aware continuous beat-to-beat blood pressure measurement technique using deep learning," *IEEE Transactions on Instrumentation and Measurement*, 71, Art no. 2519709, pp. 1-9, 2022. doi: 10.1109/TIM.2022.3212750.

[11] Yoon, G., Lee, J. Y., Jeon, K. J., Park, K. K., and Kim, H. S. "Development of a compact home health monitor for telemedicine," *Telemedicine Journal and e-Health*, 11 (6), pp. 660–667, 2005. doi: 10.1089/tmj.2005.11.660.

[12] Yen, C., and Liao, C. "Blood pressure and heart rate measurements using photoplethysmography with modified LRCN," *Computers, Materials & Continua*, 71 (1), pp. 1973–1986, 2022. doi:

10.32604/cmc.2022.022679.

[13] Slapničar, G., Mlakar, N., and Lustrek, M. "Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network," *Sensors*, 19 (15), 3420, pp. 1–17, 2019. doi: 10.3390/s19153420.

[14] Kim, M., Sung, M. D., Jung, J., Cho, S. P., Park, J., Soh, S., Joo, H. C., and Chung, K. S. "Wearable ECG-PPG Deep Learning Model for Cardiac Index-Based Noninvasive Cardiac Output Estimation in Cardiac Surgery Patients," *Sensors (Basel, Switzerland)*, 26 (2), 735, pp. 1–20, 2026. doi: 10.3390/s26020735.

[15] Xing, J., Fang, X., Bai, J., Cui, L., Zhang, F., and Xu, Y. "Study on Multimodal Sensor Fusion for Heart Rate Estimation Using BCG and PPG Signals," *Sensors*, 26 (2), 548, pp. 1–17, 2026. doi: 10.3390/s26020548.

[16] Sunwoo, J., Einalou, Z., Dadgostar, M., Renna, M., Wu, K. C., Otic, N., Martin, A., Starkweather, Z., Oh, Y., Qu, J. Z., and Franceschini, M. A. "Continuous, noninvasive blood pressure estimation using forehead NIRS-PPG and LSTM-CNN deep learning," *Biomedical Signal Processing and Control*, 113 (B), 108943, pp. 1–X, 2026. doi: 10.1016/j.bspc.2025.108943.

[17] Zeynali, M., Alipour, K., Tarvirdizadeh, B., and Ghamari, M. "Non-invasive blood glucose monitoring using PPG signals with various deep learning models and implementation using TinyML," *Scientific Reports*, 15 (1), 581, pp. 1–23, 2025. doi: 10.1038/s41598-024-84265-8.

[18] Panwar, M., Gautam, A., Biswas, D., and Acharyya, A. "PP-Net: A deep learning framework for PPG-based blood pressure and heart rate estimation," *IEEE Sensors Journal*, 20 (17), pp. 10000–10011, 2020. doi: 10.1109/JSEN.2020.2990864.

[19] Kanoga, S., Hoshino, T., Kamei, S., Kobayashi, T., Ohmori, T., Uchiyama, M., and Tada, M., "Preprocessed MIMIC III waveform database matched subset," 2023, [Online]. Available: https://drive.google.com/drive/folders/182xF0Y8NPiDGownNcxUfE4D-BPwv-bvO

[20] Li, Z., and He, W. "A continuous blood pressure estimation method using photoplethysmography by GRNN-based model," *Sensors*, 21 (21), 7207, pp. 1–15, 2021. doi: 10.3390/s21217207.

[21] Leitner, J., Chiang, P., and Dey, S. "Personalized blood pressure estimation using photoplethysmography: A transfer learning approach," *IEEE Journal of Biomedical and Health Informatics*, 25 (9), pp. 2194–2204, 2021, doi: 10.1109/JBHI.2021.3085526.