Evaluation of Deep Learning Models for Prediction of Heart Rate Using Continuous Wave Radar Signals

WEICHUN WANG, Auckland University of Technology, New Zealand

Heart rate (HR) evaluation utilizing continuous wave radar is promising because it is non-invasive, non-contact, and privacy-protecting. Deep learning (DL) is transforming and driving innovation in various domains imaginable, with a radar-based HR estimation being indeed included. Recently, many DL models have been proposed for estimating HR based on the signal data generated by radar. However, in practical settings, radar-generated raw data frequently encompasses significant environmental noise. Due to this challenge, the performance metrics of many models are typically based on the premise of utilizing denoised signal data before training the model. Consequently, we believe that evaluating the performance of DL models that trained on the radar raw data holds considerable research significance. In this paper, we have evaluated seven different existing time series-based DL models to study the feasibility of utilizing the radar raw data for model training. Our results show that environmental noise substantially lowers model performance. Although the model can achieve better learning results during the training stage, the model's prediction results for new data are relatively poor during the cross-validation phase. Among the seven models evaluated in our study, the Bidirectional LSTM (BiLSTM) model emerged with the most favorable results, displaying an average Mean Squared Error (MSE) of 85.26 across 29 experimental subjects under cross-validation conditions. Furthermore, our analysis revealed that the environmental noise included in radar raw signals can significantly impact the overall performance of DL models in predicting human heart rates.

Additional Key Words and Phrases: Continuous Wave Radar, Environmental Noise, Radar Signal, Deep Learning, Time Series, Heart Rate, HR, Vital Signs, HR Prediction

Author's Contact Information: Weichun Wang, syt0722@autuni.ac.nz, Auckland University of Technology, Auckland, New Zealand.

1 INTRODUCTION

Heart rate (HR) is a vital physiological indicator for humans that provides valuable insights into the health and overall human body system functions [34]. This indicator is vital not only in diseases and medical fields but also in numerous applications, such as fitness and exercise regimens to optimize training intensity, stress management, and human general well-being [11]. Therefore, detecting HR has great value in a fundamental aspect of medical care, health monitoring and treatment, and even in the activities for search and rescue in natural disasters [9]. The traditional method of detecting HR relies on contact wearable devices, such as the photoplethysmograph (PPG) and electrocardiogram (ECG), which are considered the gold standard, but not suitable for long-term continuous monitoring and patients with skin allergies, skin burns, or infants [32]. Another method of detecting HR is using radar to measure chest displacement with signal processing techniques, which is a promising way of applying none-contact and non-invasive [16] [29] [31].

Recent studies have demonstrated that continuous wave radar combined with deep learning technology is an alternative promising method for continuous and long-term monitoring of vital signs such as heart rates with the advantages of non-contact and non-invasive [22, 28, 32]. Unlike the signal processing method that requires applying feature extraction algorithms, deep learning models can automatically learn and identify complex patterns and anomalies and extract relevant features related to predicting heart rates from raw radar signals. What's more, by learning from a large amount of training data and continuously learning and adapting from new datasets, deep learning models can automatically optimize the learned patterns to reduce the rate of false predicting, which can provide more reliable heart rate monitoring [20]. However, accurately estimating HR information from radar raw signals remains a significant challenge, due to the environmental noise and interference. Factors such as the individual's orientation, distance from the radar, lip movements while speech or other random body movements, and respiratory harmonics substantially contribute to the emanation of environmental noise within radar signals, leading to the challenge of HR information estimation. [16, 22].

Evaluating the performance of various DL models using radar raw data remains a valuable research topic. The reason is that, firstly, most of the existing studies first try to filter the noise from radar raw signal and then send the denoised data into DL models for training [26, 28, 32]. Secondly, different deep learning models with different architecture designs have varying capabilities in learning and identifying relevant patterns from radar raw data. Thirdly, By evaluating various DL models, it's possible to find those that can best adapt to radar raw signal, leading to more accurate and effective heart rate monitoring solutions. To evaluate the performance differences of the seven DL models, this research needs to address two research questions as below.

(1) How well do time series-based DL models perform using radar raw data without noise removal to train and predict heart rate?
(2) Among all evaluated seven time series-based DL models, which performs best in predicting heart rates using radar raw data?

In this paper, The major objective is evaluating the performance difference of several time series-based DL models for predicting HR using radar raw data without noise reduction. In this study, we will utilize a public raw dataset [23] without noise removal, collected from continuous wave radar and synchronized ECG data for the ground truth to train seven distinct time series-based DL models to assess and compare how the learning ability of each DL model and different performance achievements. We concluded our main contributions in this research, as follows.

- We evaluate the performance differences of seven time series-based DL models, including LSTM, BiLSTM, GRU, CNNLSTM, RNNED, TPALSTM, and N-BEATES, when predicting heart rate, utilizing radar raw data without environmental noise removal.

- We conclude that environmental noise included in the radar raw data significantly impact the performance of DL models when predicting heart rate.

The rest content of this study is managed as follows. we first review the related work in Section 2, and in Section 3 we introduce the research methodologies, including research design, data collection and processing, various DL model introductions, and our experimental setting. In Section 4 we present the results analysis and discussion of the key factors that impact the performance of DL models and some limitations. Lastly, in Section 5, we conclude our work and outline the possibilities of the future scope.

2 LITERATURE REVIEW

From a broad view, existing studies for detecting HR can be summarized into two categories: contact-based methods like the photoplethysmography (PPG) and electrocardiogram (ECG) are the most prevalent two devices that detect HR are derived from electric potential and light accordingly, and non-contact based methods such as radar based method that can provide the advantage of non-contact and non-invasive HR detection. The non-contact-based method can also be summarized into two sub-categories: signal processing-based and deep learning technique-based. This section mainly review the existing works of non-contact-based solutions for HR detection and we will also review several evaluation metrics for measuring the DL models' accuracy, reliability, and overall performance.

2.1 Signal Processing Methods

There are several algorithms exist based on signal-processing for heart rate extraction using radar. This literature [16] designed an algorithm to detect vital sign for range bin variation induced by heavy breathing and body movements and an algorithm to extract heartbeat signal for suppressing the harmonic of breathing and separating the heartbeat signal. The results of the experiments presented that the algorithm proposed in this literature enhances signal-to-noise ratio (SNR) and decrease harmonic noise. However, the proposed method can not be used for multi-subject vital signs monitoring. This literature [31] introduces a filtering technique utilizing polynomial fitting extraction which can greatly remove harmonic interference noise from respiration, contributing to the accuracy of heart rate detection improvement. The value of this method lies in the fact that by subtracting polynomial fitting waveforms from the received signal, it is easy to retrieve quasi-ideal matching pulses that approximate the actual heartbeat signal, providing high adaptability for individual subjects.

2.2 Deep Learning Methods

Deep learning, a branch of artificial intelligence (AI), has revolutionized numerous fields of science, technology, and industry. It enables computers to automatically learn highly complex representations from training data [20]. Many researchers have leveraged the strength of deep learning to develop innovative solutions for heart rate monitoring with continuous wave radar. This literature [8] proposed a multidimensional CNN-based model to extract vital signs using Impulse Radio Ultrawide-Band Radar, which demonstrated competitive performance in detecting long-term period vital signs in both static and dynamic states. This literature [27] also proposed a Doppler radar based solution for extracting heart rate under both stationary conditions and conditions of active exercise by using the combined model of recurrent neural network (RNN) and convolutional neural network (CNN), achieved the highest performance: 99% for HR detection, which has demonstrated the possibility of human vital signs prediction using non-contact radar and deep learning technology. Similarly, this literature [9] designed a multi-scale detection module that uses different convolution

kernel sizes and a heart rate feature attention module to address the multi-scale detection problem and distinguish the heart rate patterns from other unrelated body movements noise, and then these two models are combined to form a fusion convolutional network module for effectively extracting heart rate. After training and validation experiments, a value of MAE of HR extracting achieved 1.87 beats/min, and overall 97.51% relative estimation accuracy was achieved, which provided a valuable guideline for the application of DL technology in the domain of vital sign extracting from mmWave radar signal. In [15], a deep learning classifier, AlexNet, and a support vector machine (SVM) were employed for detecting the pattern of the heart rate feature using time series radar raw data with motion noise, rather than converting the signal into a frequency domain for the in-car monitoring of people. The proposed method was experimented with in a stationary car on five human subjects to show maximum performance of 97.4% accuracy.

2.3 Evaluation Metrics for Deep Learning Models

The metrics for DL model evaluation are crucial. On one hand, they can accurately measure model performance, identify bias and variance, and guide the ability of model improvement. On the other hand, they can enable objective comparison between different models, facilitating the selected and best-performing one and validating that the model can perform well with other unseen data, ensuring model robustness and reliability [3] [5]. Different evaluation metrics are based on different measurements and show different perspectives of model performance. According to the calculation measurements, this literature [13] classified and summarized three different kinds of evaluation metrics. The first category is based on squared errors or absolute errors, which are named scale-dependent metrics, compromising Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The characteristic of scale-dependent metrics is that they have the same range of scale as the reference data and calculate errors within the same unit range. However, the disadvantage is that comparing series on different ranges of scales or with different units might be difficult. The second category is based on percentage errors, which are called percentage-dependent metrics, comprising the errors of symmetric mean absolute percentage (SMAPE) and mean absolute percentage (MAPE). Percentage-dependent metrics are unit-free means no limitation on units, they measure the error size in percentage terms and provide insightful assessments of prediction quality. However, the disadvantage of percentage-dependent metrics is that they are scale-sensitive, and they are easier to get extreme values if the reference value tends to be very small. The third category is correlation-based, including coefficient of determination (R^2, also called R-squared) and correlation coefficient (R). The correlation coefficient (R) ranges from -1 to +1, in which -1 is inverse correlation and +1 is the positive correlation. R^2 is the correlation square between the reference value and the models' prediction value, and the value of R^2 ranges from $-\infty$ to +1, where $-\infty$ indicates the model performs worst, and +1 indicates the model performs best. Note that, if the value of R^2 is 0, the DL model does not learn effective patterns around its mean of reference data [5].

3 RESEARCH METHODOLOGY

3.1 Research Design

The major objective of this paper is to evaluate the performance difference of seven time series-based DL models for predicting HR using radar raw data without noise reduction. To address the research questions effectively, we conducted an overall plan as our research blueprint for the entire research process that is shown as follow Figure 1:

Our research process contains four main steps which are data collection, data process, DL models implementation, and results analysis and evaluation. In Step 1, we utilize the public dataset that

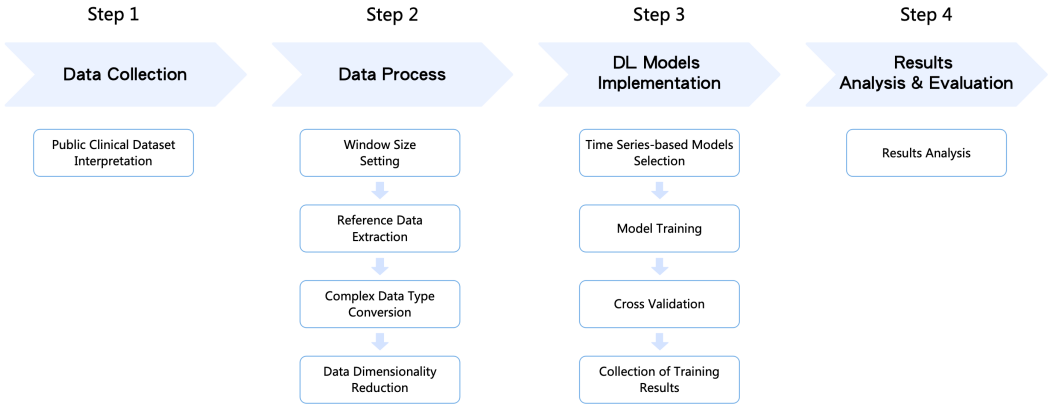

Fig. 1. The Designed Research Process.

includes synchronized data from a continuous wave radar and an ECG device for reference [23]. In Step 2, a fixed window size of 5 seconds is selected for processing the datasets obtained from radar and electrocardiogram (ECG) devices, respectively. Subsequently, heart rate (HR) is extracted from the ECG-collected dataset using the fixed 5-second window as the reference data. Given that radar raw signals are presented in the form of In-Phase and Quadrature signals, a conversion process is implemented to transform the complex data type into a three-dimensional float data type. Ultimately, a Principal Component Analysis (PCA) technique is employed to decrease the dimension of the three-dimensional float data to two-dimensional float data. During the process of Step 3, we trained seven distinct DL models based on time-series data, as elaborated upon in subsequent sections. Each model underwent cross-validation procedures to assess performance across 29 individual experimental subjects. In the final Step 4, we employ metrics such as MSE, MAE, and RMSE to evaluate the performance of each DL model.

3.2 Data Set

In this paper, we utilize a public database of radar-recorded vital signs, from 30 healthy participants. These individuals underwent stimulation using various methods, including the Valsalva maneuver (VM), breath-holding, and the tilt table test. Synchronously, Task Force Monitor (TFM) was taken to measure ECG signals as reference data when extracting HR using the radar signal [23].

All measurements were conducted by medical professionals at the University Hospital Erlangen following a predefined protocol. The placement of the radar system and ECG electrodes can be seen in Figure 2.

Among the 30 participants, there are 14 males and 16 females with 30.7±9.9 years average age, 175.7±10.5cm average height, 72.2±14.0kg average weight, and 23.2±3.3kg/m2 average BMI. Each participant was allocated a distinct identifier for pseudonymization purposes. Also, all participants provided written consent, permitting the sharing of pseudonymized data.

Five different scenarios were conducted, including Resting, Valsalva, Apnea, TiltUp, and TiltDown, with oversight by a minimum of two individuals following a standardized protocol. This study specifically focuses on the dataset collected during the Resting scenario for our research experiments. Figure 3 depicts an exemplary extraction of ECG and radar raw signals obtained during a resting scenario.

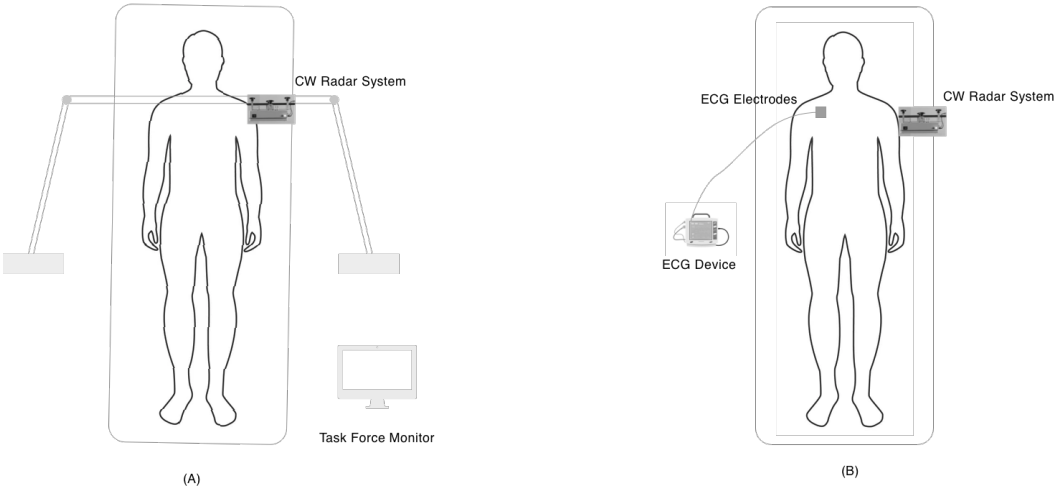

Fig. 2. The Placement of the Radar System and ECG Electrodes.

3.3 Data Pre-processing

In light of the collected dataset, for this study, we set a fixed window size of 5 seconds. Subsequently, the window is shifted by 1 second incrementally to generate data samples, as visually depicted in Figure 3. Following this method, the radar raw data alongside the corresponding electrocardiogram (ECG) data undergo processing to generate the training dataset and the ground truth dataset. Also, Python is taken as the programming language to simulate these signals and handle other data processing, and Pytorch is the deep learning framework to train various DL models.

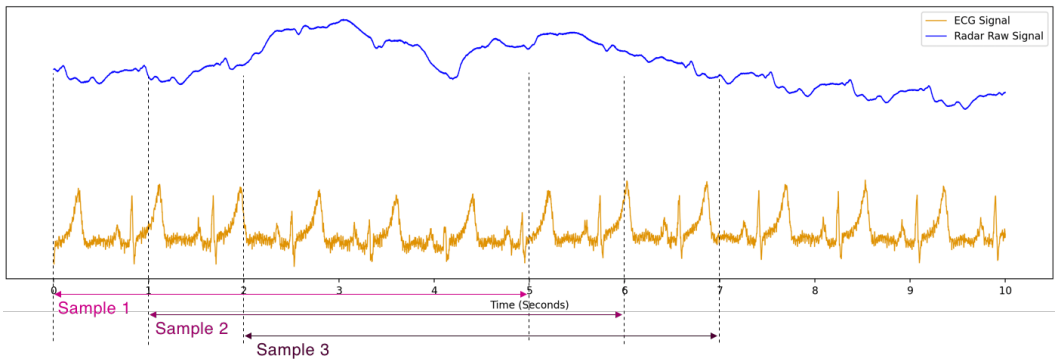

Fig. 3. The Method of Generating Data Samples.

The radar raw signals, called In-Phase and Quadrature components, are produced concurrently utilizing the 24-bit analog-to-digital converter (ADC) ADS1298 manufactured by Texas Instruments, operating at 2000Hz sampling rate. In this study, we preprocess the complex data type into a one-dimensional floating-point format before training DL models. Each data sample within the training dataset comprises a 5-second sequence containing 2000 one-dimensional floating-point values per second, thereby constituting a 4-dimensional matrix. Consequently, we employ the Principal Component Analysis (PCA) algorithm to diminish the dimension of the training dataset from 4 dimensions to 3 dimensions.

PCA, a dimensionality decrease algorithm, is designed to convert high-dimensional data into a lower-dimensional space while endeavoring to maintain maximal original variance [25]. Mathematically, the PCA algorithm can be divided as below sub steps. Let X be an $n \times d$ matrix representing the dataset, in which n is the number of data samples and d is the number of features. In our study, the value of n is the sequence length, that is 5 seconds, and the value of d is 2000 one-dimensional with float values.

- `Mean Centering`: PCA initiates by subtracting each feature's mean from the dataset X, thus centering the data around the origin. This preprocessing step facilitates the computation of the covariance matrix.

$$X_{centered} = X - X_{mean} \tag{1}$$

  where, $X_{centered}$ is the centered dataset, and X_{mean} is the mean of each feature in the dataset.
- `Covariance Matrix Computing`: Calculate the covariance matrix (Σ) of the centered data to measure the relationship between each pair of features.

$$\Sigma = \frac{1}{n} X_{centered}^{T} X_{centered} \tag{2}$$

  where, $X_{centered}^{T}$ denotes the transpose of the centered data matrix.
- `Eigenvalue Decomposition`: Performs eigenvalue decomposition on the covariance matrix (Σ) to derive the eigenvectors and eigenvalues.

$$\Sigma V = V \Lambda \tag{3}$$

  where, V denotes the eigenvectors' matrix, and Λ denotes the diagonal eigenvalues' matrix.
- `Selecting Principal Components`: Rank the eigenvectors in descending order according to their corresponding eigenvalues. The eigenvector associated with the largest eigenvalue is the first principal component (PC1), followed by subsequent components in descending order of eigenvalues, represented as PC2, PC3, and so on.
- `Projection`: Project the centered data onto the subspace defined by the chosen principal components.

$$X^{'} = X_{centered} V_k \tag{4}$$

  where, $X^{'}$ denotes the reduced-dimensional dataset, and V_k denotes the matrix that is consist of the k eigenvectors corresponding to the top k eigenvalues.

3.4 Deep Learning Models

Deep learning has achieved significant success in various domains such as health care [30], robotics [14], and autonomous vehicles [19]. It has also been effective in modeling temporal sequences [21] [20]. This section will introduce the architecture and mathematical formulas of the seven time series-based DL models selected in this study which are Long Short-Term Memory (LSTM) Network, Gated Recurrent Unit (GRU) Network, Bidirectional LSTM (BiLSTM)Network, RNN Encoder–Decoder (RNNED), A Fusion of CNN and LSTM (CNN-LSTM) Networks, Temporal Pattern Attention with LSTM (TPA-LSTM) Network, and Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS).

3.4.1 Long Short-Term Memory (LSTM) Network [12]. The Long Short-Term Memory (LSTM) Network, commonly referred to as LSTM, is a specialized variant RNN. It adeptly captures long-term dependencies and addresses issues like vanishing and exploding gradients inherent in conventional RNN architectures. Initially introduced in [12], LSTM has demonstrated notable efficacy across diverse domains such as image recognition, speech recognition, and natural language processing

(NLP). Its widespread adoption in deep learning underscores its utility in tackling real-world challenges.

LSTM, like the standard RNN, is formed by a chain of repeating cells of a neural network. The critical part of LSTM is the cell state, which comprises three gates with sigmoid layers that output numbers from zero to one, to decide which information should be forgotten and which should be memorized. In the first step of LSTM, a sigmoid layer also named as a "forget gate layer" determines which information should be discarded from the cell state. This layer examines the input from the preceding hidden state (h_{t-1}) and the current input vector (x_t), producing an output value ranging from 0 to 1 for each element within the cell state (C_{t-1}). The value of 1 means the complete retention of information, and the value of 0 means complete elimination of information. Mathematically, it can be expressed as:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{5}$$

where, W_f and b_f refer to the weight and the bias which are automatically adjusted during the process of training. h_{t-1} and x_t is the preceding hidden state and the current input.

In the second step of LSTM, a sigmoid layer also named a “input gate layer” determines which new information should be memorized in the cell state, and then a vector of new candidate values ($\tilde{C}_t$) which is produced by a tanh layer is added to the state. Finally, the new candidate values ($\tilde{C}_t$), and the previous cell state (C_{t-1}) were combined, as shown in Equation 6.

$$\begin{aligned} \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \end{aligned} \tag{6}$$

where, W_i, W_C, b_i and b_C refer to the weight matrices and the bias.

The ultimate step is to determine what information should be output based on our cell state along with the manipulation of sigmoid and tanh layers, as shown in Equation 7.

$$\begin{aligned} O_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= O_t * \tanh(C_t) \end{aligned} \tag{7}$$

where, W_o and b_o refer to the weight matrices and the bias.

3.4.2 Bidirectional LSTM (BiLSTM) Network [10]. In a standard LSTM network, data propagation occurs unidirectionally along the temporal axis, progressing only from the past to the future. During computation at the current time t, when the LSTM cell unit computes the hidden layer state h_t and output O_t, it exclusively considers the current input vector V_t and the preceding hidden layer state h_{t-1}. However, in certain practical scenarios, the output O_t at the current time t may not only be influenced by past states but could also be related to the future states.

Bidirectional LSTM [1] incorporates two independent LSTM chains that process information in both directions. Each direction has its own hidden layer, and the outputs from these layers are merged and propagated to a common output layer. The first direction is a forward-propagation LSTM chain, which processes the sequence in the order of time from past to future ($t = 1$ to $t = T$), capturing past context information and generating a forward hidden state h_t^f. Conversely, the second direction is a backward-propagation LSTM chain, which processes the sequence in reverse order, capturing future context information from $t = T$ to $t = 1$ and generating a backward hidden state h_t^b. This approach enables the model, at each time step t, to integrate information not only

from the past but also from the future. The combined hidden state h_t can be represented at the current time t as:

$$h_t = h_t^f + h_t^b \tag{8}$$

When both LSTM chains are combined together, information is propagated in a bidirectional manner.

3.4.3 Gated Recurrent Unit (GRU) Network [6]. The Gated Recurrent Unit (GRU) Network, introduced by [6] in 2014, is a variant of RNN. Similar to the LSTM, it was also designed to solve the problems of gradient explosion and vanishing gradients in a long-term memory and backpropagation. But GRU has a simpler architecture, usually, it can provide a comparable performance as LSTM, and the computational speed is faster. In the architecture of GRU, it consists of two gates: one is reset gate R_t and other one is update gate Z_t. Reset gate handles how the new inputs are combined with preceding memory H_{t-1}. Update gate handles what and how the preceding state needs to be kept. Assuming x_t is the input vector at the time step t and H_{t-1} is the state of the hidden layer at the time step $t-1$, which contains relevant information from the previous node, the mathematical expressions for reset gate R_t and update gate Z_t are as below:

$$\begin{aligned} R_t &= \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \\ Z_t &= \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \end{aligned} \tag{9}$$

where, W_{xr} and W_{xz} are the weights corresponding to the input vector, W_{hr} and W_{hz} are the weights corresponding to the hidden layer, and b_r and b_z are bias. The hidden layer output H_t at the time step t is expressed as:

$$H_t = Z_t \bigodot \tilde{H}_t + (1 - Z_t) \bigodot H_{t-1} \tag{10}$$

where, $\tilde{H}_t$ dnotes the current candidate hidden state at the time step t, and the symbol $\bigodot$ is the $\tilde{H}_t$ product operator. We have already known the range of the update gate Z_t is [0,1]. When Z_t is closer to 0, more past data has been `memorized` from the preceding hidden state H_{t-1}; the closer Z_t is to 1, it means more past data has been `forgotten`. The candidate hidden state $\tilde{H}_t$ is expressed as:

$$\tilde{H}_t = \tanh(X_t W_{hx} + R_t \bigodot H_{t-1} Whh + b_h) \tag{11}$$

where, W_{hh} and W_{hx} are the weights, and b_h is the bias. When the value of reset gate R_t approaches 0, the model discards past hidden information and leaves only the current input information. When R_t approaches 1, the useful previous information will be put into the current information.

3.4.4 A Fusion of CNN and LSTM (CNN-LSTM) Networks [17]. The CNN-LSTM model integrates Convolutional Neural Networks (CNNs) and LSTM, constituting a hybrid deep learning architecture. It is widely favored for tasks characterized by time series-based data with temporal and spatial dependencies. The model proposed in this paper [17] initially employs CNN for feature extraction from the dataset, followed by LSTM for predicting stock prices using the extracted features. The experimental results demonstrate that the CNN-LSTM model is reliable and accurate in predicting stock price, offering valuable insights for scholars investigating financial time series data. The CNN-LSTM model consists of followed steps, firstly, the sequential data undergoes feature extraction via a one-dimensional convolutional layer followed by *ReLU*, the nonlinear activation function, secondly, a max pooling layer is applied to reduce the dimensions of the extracted features. Subsequently, the data passes through an LSTM layer to capture short and long term dependencies inherent in the time series, and ultimately, a fully connected linear output layer is utilized to produce prediction results.

3.4.5 RNN Encoder–Decoder (RNNED) [7]. This paper [7] designed a neural network architecture using GRU units to form encoder and decoder components that learn to encode sequential input $X = \{x_1, x_2, ..., x_T\}$ into the vector with a fixed-length and then to decode the given vector back into a output sequence with a targeted length $Y = \{y_1, y_2, ..., y_{T'}\}$, where it should be noted that the input sequence length T and the output sequence length $T^{'}$ might differ. In this study, we set $T = 5$ and $T^{'} = 1$ as each 5 seconds radar raw wave can be extracted one value of heart rate.

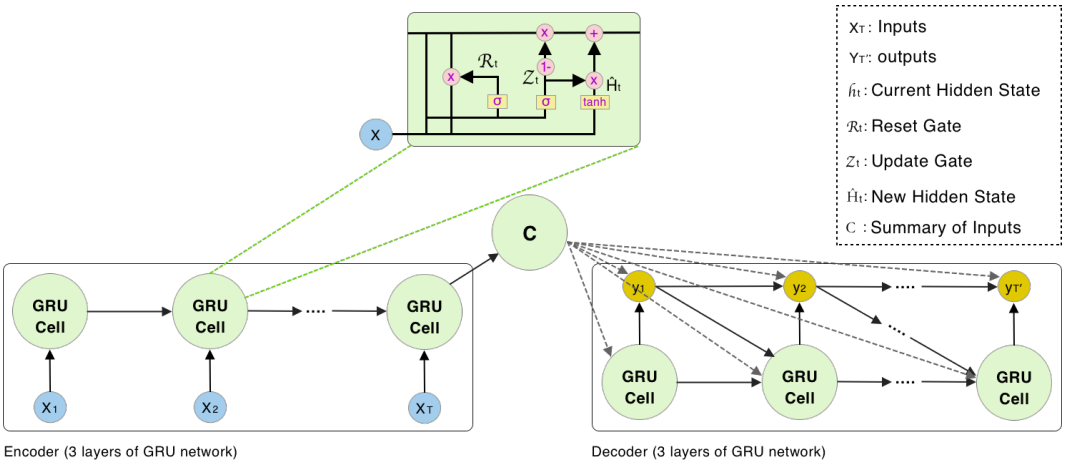

Fig. 4. Architecture of GRU Encoder-Decoder Network.

Figure 4 is the depiction of the GRU Encoder-Decoder model architecture. The encoder comprises a 3-layer GRU network that processes each input vector of the input sequence x sequentially. As it processes each input vector, the hidden state updates according to Equation 10. Once the entire sequence is read, the hidden state serves as a summary C of the entire input sequence. The decoder component is another 3-layer GRU network designed to generate the output sequence. It predicts y_t based on the hidden state H_t, the preceding output y_{t-1}, and the summary C of the input sequence. Thus, the hidden state of the decoder component at the time t is expressed by:

$$H_t = f(H_{t-1}, y_{t-1}, C) \tag{12}$$

where f denotes a non-linear activation function.

3.4.6 Temporal Pattern Attention with LSTM (TPA-LSTM) Network [24]. Predicting multivariate time series data poses challenges due to intricate and nonlinear interdependencies between different time steps and series. In this paper [24], it has built a model for a long-term dependency task in time series data to obtain accurate prediction which was achieved by employing LSTM augmented with an attention mechanism and a convolutional neural network. The attention mechanism captures temporal patterns across time series, and the CNN extracts temporal pattern information from each individual variable.

As illustrated in Figure 5, the proposed model comprises several key steps. Initially, time series data expressed as $X = \{x_1, x_2, ..., x_t\}$ undergoes processing through LSTM networks, where h_t is the hidden state of the LSTM at the time step t. Subsequently, k $1 - D$ CNN filters with a length w convolve over m features of the hidden states, yielding a matrix H^C with m rows and k columns. Within the attention mechanism, each row of H^C is assigned a weight by the scoring function, based on its comparison with the current hidden state h_t. These weights are then normalized,

and the resulting weighted sum of the rows of H^C produces the output V_t. Finally, V_t and h_t are concatenated, and a matrix multiplication operation is performed to generate $h_t^{'}$, which constitutes the final forecast value. Formally, the CNN operation is expressed by

$$H_{i,j}^C = \sum_{l=1}^{w} H_i, (t - w - 1 + l) \times C_{j,T-w+l} \tag{13}$$

where, T denotes the maximum length w, and $H_{i,j}^C$ denotes the convolutional value of the i_{th} row vector and the j_{th} filter. The attention mechanism is expressed by

$$\begin{aligned} f(H_i^C, h_t) &= (H_i^C)^T W_a h_t \\ a_i &= \sigma(f(H_i^C, h_t)) \\ v_t &= \sum_{i=1}^{n} a_i H_i^C \\ h_t^{'} &= W_h h_t + W_v v_t \\ y_{t-1} + \triangle &= W_{h'} h_t^{'} \end{aligned} \tag{14}$$

where, $f(H_i^C, h_t)$ is the scoring function, H_i^C is the i_{th} row of H^C, $W_a \in R^{k \times m}$, a_i is the attention weight, v_t denotes the context vector which is a weighted sum of row vectors of H^C, $W_h \in R^{m \times m}$, $W_v \in R^{m \times k}$, $W_{h'} \in R^{n \times m}$, $h_t^{'} \in R^m$, and $y_{t-1} + \triangle$ is the final forecast value.

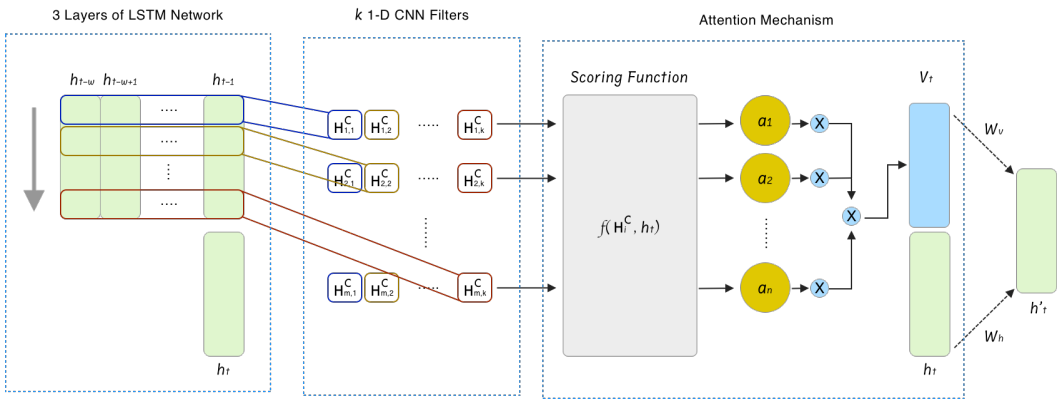

Fig. 5. Architecture of TPA-LSTM Network.

3.4.7 Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS) [21]. In this paper [21], a deep neural architecture named N-BEATS is proposed, incorporating backward and forward residual links, along with a deep stack of fully connected layers. The proposed neural architecture emphasizes its interpretability and applicability across diverse target domains without modification to a wide array, all while facilitating rapid training. As shown in Figure 6, the overall architecture comprises two primary stacks: a trend stack and the seasonality stack. Each stack consists of basic building blocks and two residual branches, one traversing the backcast prediction of each layer and another one traversing the forecast branch of each layer. The residual operation is expressed by the following equations:

$$x_l = x_{l-1} - \hat{x}_{l-1}$$
$$\hat{y} = \sum_l \hat{y}_l \tag{15}$$

where, x_l denotes the backcast residual branch which is to analyze the sequential of the input signal, $\hat{x}_{l-1}$ denotes the block's best estimate of x_{l-1}, $\hat{y}_l$ denotes the partial forecast of outputs of the each block, and the final forecast, denoted as $\hat{y}$ is the aggregate of all partial forecasts.

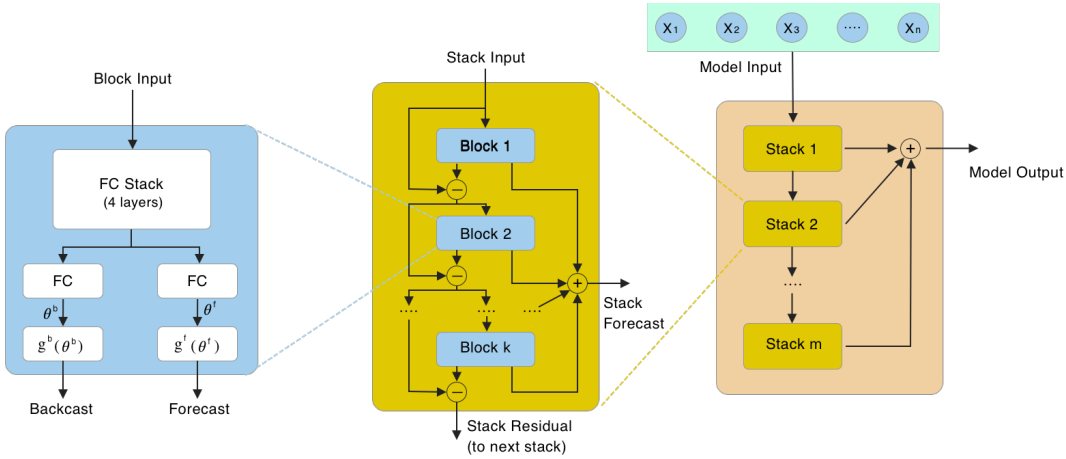

Fig. 6. Architecture of N-BEATS Network.

Within the internal architecture, the basic building block comprises two components. The first one is a fully connected layer responsible for producing both the forward predictor θ_l^f and the backward predictor θ_l^b of expansion coefficients. The second part encompasses the backward basis layer g_l^b and the forward basis layer g_l^f, which accept the corresponding forward θ_l^f and backward θ_l^b expansion coefficients. Internally, these layers project the inputs onto a group of basic functions and produce the backcast $\hat{x}_l$ and the forecast outputs $\hat{y}_l$.

In a generic DL model context, incorporating arbitrary g_l^b and g_l^f configurations for each layer fosters transparency to gradient flows within the network. In a special instance, deliberate structure constraints are enforced on g_l^b and g_l^f shared across a stack. The design of this architecture plays a vital role in facilitating interpretability by the aggregating meaningful partial forecasts.

3.5 Experiments Setup

The primary aim of this study is to evaluate the performance of seven DL models while predicting human heart rate utilizing radar raw signals. To achieve this objective, we carry out numerous experiments within diverse combinations of hyperparameters and observe the performance difference. Table 1 gives the details of hyperparameters for different DL models that have achieved the best performance among the all experiments we carried out. Due to the varying convergence speeds of different models, we use a maximum time of 1000 epochs for the N-BEATS model and 500 epochs for other models.

To enhance the objectivity of model performance evaluation and improve the model's generalization ability, we employ the cross-validation technique to organize and train DL models, which will

Table 1. Hyperparameters Configuration of DL Models

Models	Layers	lr	Droprate	Optimizer	Comments
LSTM	3	0.001	0.01	ASGD	Hidden layer = 1024 nodes
Bi-LSTM	3	0.001	0.01	ASGD	Hidden layer = 1024 nodes
GRU	3	0.001	0.01	ASGD	Hidden layer = 1024 nodes
CNN-LSTM	3	0.001	0.01	Adamax	1-D CNN with ReLu.
RNNED	2	0.001	0.01	ASGD	Hidden layer = 1024 nodes
TPA-LSTM	3	0.001	0.01	Adamax	1-D filters k is 3, and length w is 5.
N-BEATS	4	0.001	/	Adamax	Stack size = 4, Blocks = 2

be introduced in the subsequent paragraph. Throughout the model training phase, we use average stochastic gradient descent (ASGD) and Adamax optimizer to train the deep learning model, shown in Table 1. Furthermore, we use the loss function with MSE to calculate the discrepancy between the model's predictions and the reference values. Subsequently, In the evaluation phase, four metrics: MSE, RMSE, MAE, and R^2, are utilized to evaluate the performance of the trained models, as shown in Equations 16.

$$
\begin{aligned}
MSE &= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \\
RMSE &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \\
MAE &= \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \\
R^2 &= 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}
\end{aligned}
\tag{16}
$$

where, N denotes the size of the sample data, x_i denotes the input vector, y_i denotes the observed ground truth value, $\bar{y}_i$ denotes the mean of the reference value y_i, and $\hat{y}_i$ denotes the prediction value of the trained model on the input vector x_i.

3.5.1 Cross-validation. Cross-validation is a statistical method employed in machine learning to evaluate the model performance by repeatedly partitioning the dataset into training and validation sets, training the model on one partition, and evaluating it on another [2]. This process is crucial for estimating the model's ability to generalization on unseen data, detecting overfitting or underfitting issues, and tuning hyperparameters. By providing a more dependable model performance estimation than a single train-test dataset, cross-validation aids in model selection, feature selection, and hyperparameter tuning, ultimately leading to better-performing machine learning models with improved generalization capabilities.

K-fold cross-validation stands out as one of the most popular methods for cross-validation, it ensures that every data point is used for both training and validation exactly once leads to more efficient use of the available data, and reduces bias in the performance estimation. In K-fold

cross-validation, the learning set is divided into k separate subsets of roughly same proportions, with $k-1$ subsets designated for the dataset of training and the remaining subset reserved for the validation set, the cross-validated performance is determined by calculating the mean of the performance obtained from the k validation sets. The cross-validated performance $\hat{\epsilon}_{cv}$, is then given as:

$$\hat{\epsilon}_{cv} = \frac{1}{k}\sum_{j=1}^{k} L_j(y_i - f_{-j}(x_i)) \tag{17}$$

where, f_{-j} represents the model trained on all subsets of the learning set except the j_{th} subset, y_i denotes the observed ground truth value, $f_{-j}(x_i)$ is the predicted value, and $L_j(y_i - f_{-j}(x_i))$ denotes j_{th} performance measurements error.

In this study, we set the k value to 29, which means that the data of 29 participants out of 30 in the public dataset was used as the learning dataset to form a 29-fold (the ECG data of the third participant cannot extract heart rate with a 5-second window size and is therefore discarded).

4 RESULTS ANALYSIS AND DISCUSSION

To answer the research questions in this paper, we trained and evaluated 7 time series-based DL models for a maximum of 1000 epochs and 16 batch sizes using the Pytorch framework based on the radar raw data generated by continuous wave radar sources. We trained our models using the ASGD and Adamax optimizer with 1024 hidden neuron nodes and a learning rate 0.001. We also set a dropout of 0.01 to mitigate overfitting tendencies. In this way, dropout deactivates a subset of neurons during training, with a 1% probability, thereby preventing the models from overly relying on specific features and promoting generalization. Figure 7 shows the average of training and

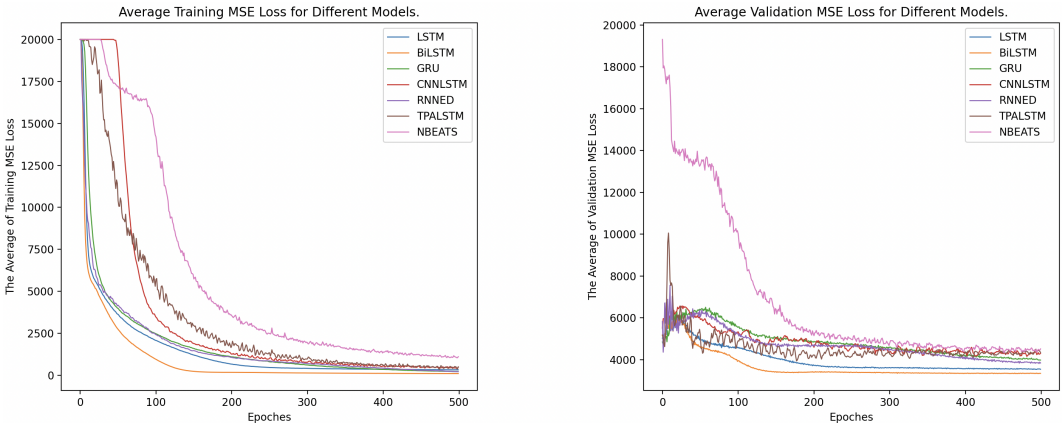

Fig. 7. The Average of Training and Validation Loss for Different Models. (For a more intuitive display, we set the value of loss greater than 20000 to 20000.)

validation loss for 7 DL models with 500 epochs (For a more intuitive display, we set the value of loss greater than 20000 to 20000). The results show that the BiLSTM model exhibits the quickest convergence rate during the training phase, whereas the N-BEATS model demonstrates the slowest convergence. during the validation process, all models almost no longer continue to converge after converging to a certain extent. After completing the training of 7 DL models, we employed these models along with cross-validation to predict and evaluate the four key indicators, including

MSE, RMSE, MAE, and R2 for the corresponding 29 participants. The results are shown in Figure 8, and the aggregation details of predictions are shown in Table 2. As we can observe from Figure 8, different DL models tend to have consistent volatility in their prediction results for each participant, among which the NBEATS model presents the highest degree of fluctuation in predicting results for each participant. Figure 9 presents the produced 29 evaluation values for each model which

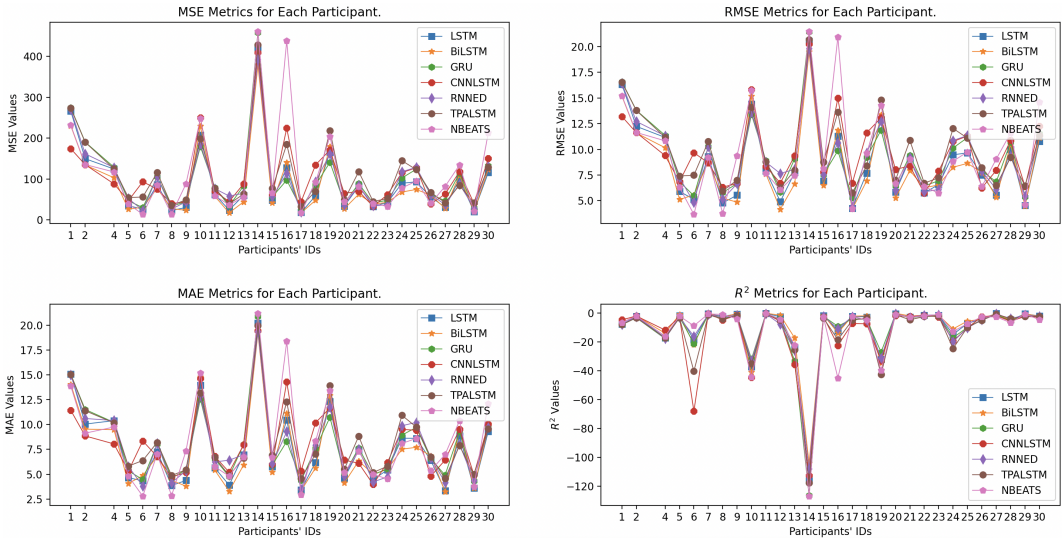

Fig. 8. The Prediction and Evaluation of Four Key Indicators of MSE, RMSE, MAE, and R_2 for 29 Participants (the learning dataset of the third participant was discarded due to the inability to extract heart rate for the ground truth data with a 5-second window size).

evaluates against four metrics. In these results, the height of the box reflects the dispersion of the prediction results, and the shorter the box, the more stable the prediction results are. The bottom of the box indicates the lowest quartile or first quartile (Q1), which represents 25% of the lowest data, and the top of the box indicates the highest quartile or third quartile (Q3), which represents 25% of the highest data. The lines extending from the top and bottom edges of the box, named whiskers, represent the maximum and minimum prediction values. The orange line on the box represents the respective average values of four evaluation metrics across all participants. The points plotted outside the whiskers indicate outliers, exhibiting exceptional data points that deviate from the overall prediction values. As we can observe, the BiLSTM model outperforms all other 7 DL models in all four metrics in terms of the average value of the four evaluation metrics, followed closely by the performance of the LSTM model, while the N-BEATS model holding the interpretable characteristic presents the poorest performance and the TPALSTM model ranks second to last. GRU model, that is similar to the LSTM model but has a simpler architecture, however, its overall performance is lower than the LSTM model. CNN-LSTM model, which represents a fusion of CNN and LSTM networks, achieves poorer performance than the GUR and LSTM models. RNNED model, which is consists of the encoder and decoder components, the overall performance is better than the GRU model but lower than the LSTM model.

Thus far, we have been able to answer Research Question 2 which is that the BiLSTM model performs the best across other evaluated models. By observing Figure 7, during the training phase, most of the models were able to converge effectively and quickly. However, during the validation

Table 2. The Aggregation Details of the Prediction Value of Four Key Indicators of MSE, RMSE, MAE, and R2 for Participants 1-30.
"Mean" indicates the respective average values of four evaluation metrics across all participants;
"Min (8)" indicates Participant 8 has the smallest values for most of the four evaluation metrics;
"Max (14)" indicates Participant 14 has the greatest values for most of the four evaluation metrics.
Note that: the maximum values are indicated as exception values in Figure 9.

	Metrics	LSTM	BiLSTM	GRU	RNNED	CNNLSTM	TPALSTM	NBEATS
Min (8)	MSE	23.1498	27.5179	33.9495	26.5831	40.2068	35.5651	13.9739
	RMSE	4.8114	5.2458	5.8266	5.1559	6.3409	5.9637	3.7382
	MAE	3.8925	4.3746	4.6792	4.127	4.7801	4.8648	2.8436
	R^2	-2.3854	-3.0242	-3.9647	-2.8875	-4.8798	-4.201	-1.0435
Max (14)	MSE	421.8848	380.4825	462.4107	392.7846	405.7246	428.0234	461.0484
	RMSE	20.5398	19.506	20.6887	21.5037	20.1426	19.8188	21.472
	MAE	20.2087	19.2738	20.9416	19.3623	19.3573	v19.9399	21.1987
	R^2	-116.1309	-104.6361	-127.3824	-108.0516	-111.6442	-117.8352	-127.0042
Mean	MSE	91.0739	**85.2598**	101.0226	97.3041	106.1107	109.9003	112.7853
	RMSE	8.7571	**8.4379**	9.3815	9.211	9.7505	9.8569	9.6
	MAE	7.5785	**7.3492**	7.9712	7.8569	8.1825	8.3818	8.2184
	R^2	-11.1975	**-10.5369**	-12.2298	-11.43	-14.5948	-13.7049	-13.1269

phase, the model slowly converged to a certain extent and then stopped continuing to converge. This is a typical overfitting phenomenon where the models perform so well during the training phase but badly on unseen validation data. To observe this phenomenon more intuitively, we selected data from four participants as the validation dataset during the cross-validation phase and we employed the BiLSTM model, which demonstrated the best overall performance in terms of the average value of the four evaluation metrics to predict the corresponding heart rates. As illustrated in Figure 10, the prediction results for Participant 14 significantly deviate from the reference heart rate, with a mean square error of 380.4825 and R-squared score (R^2) of -104.6361 (as detailed in Table 2). The reason why the predicted results differ significantly from the reference heart rate is mostly because of the influence of environmental noise. According to this paper [4], environmental noise can introduce variability and errors into the dataset and distort the true signal, leading the model to misinterpret noise as valid signal patterns, resulting in learning incorrect relationships. As illustrated in Figure 9 and detailed in Table 2, the BiLSTM model, which demonstrated the best overall performance, still yields the average value of R-squared score (R^2) of -10.5369 which is far less than 0, indicating that the model might incorrectly learn relationships from noise data to form predictive patterns. Consequently, the predicted results are even worse than those obtained by directly utilizing the average of the reference heart rate as the prediction results (Generally, the value range of the R-squared metric should be $[-\infty, 1]$. The larger the R-squared, the better the model performance. According to Formula 16, when $R^2 = 0$, the numerator is equal to the denominator, and each predicted value of the sample data is equal to the mean of the sample data [5]). In this study, we utilized radar raw data to train multiple DL models, which means the radar raw data may comprise environmental noise, such as breathing harmonics, hand gestures, lip movements while speech, or other body movements, leading to the difficulty to separate the feature associated with the human heart rate from environmental noise [15]. Researchers have proposed numerous approaches to detect HR using continuous wave radar [16] [8] [31] [27]. However, these methods can obtain satisfactory estimation results only when radar raw data's signal-to-noise

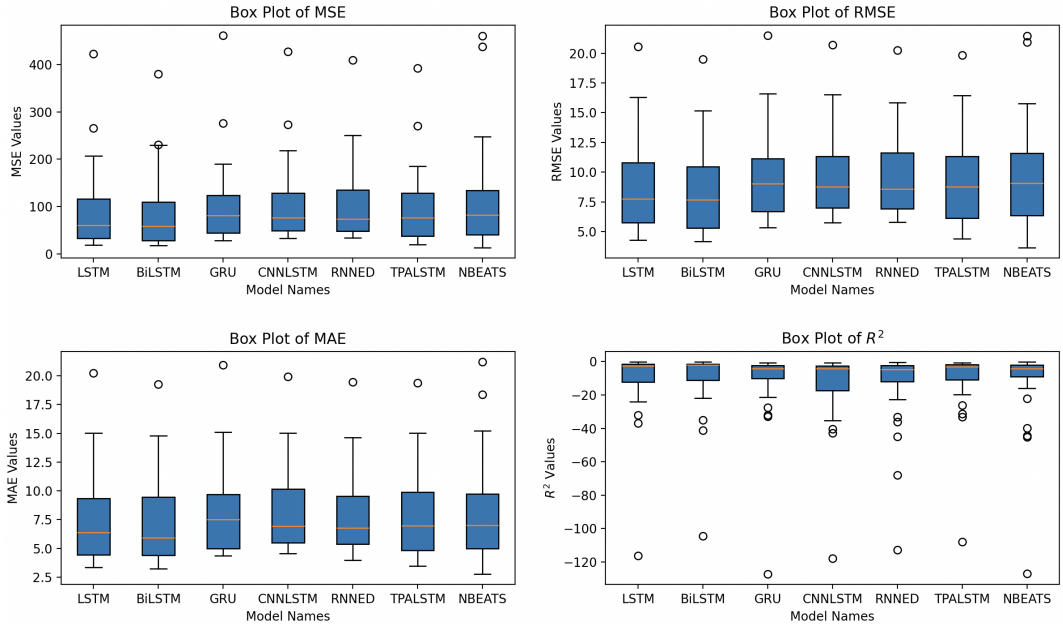

Fig. 9. The Box Plot of Four Key Indicators of MSE, RMSE, MAE, and R^2 for 7 DL models.
During the cross-validation process, we employed seven DL models to train the data across $k = 29$ folds, resulting in 29 trained models per DL model. In the validation phase, we used four key evaluation metrics to predict and assess the data of the corresponding 29 participants. Each model, evaluated against four metrics, produced 29 evaluation values. Finally, we utilized a box plot to visualize the results.

ratio (SNR) is sufficiently high, if it is in conditions of severe noise, the accuracy of most existing HR estimation methods significantly decreases [33]. The reason is that the chest wall movements induced by heartbeats are quite small and the noise can be as substantial or even more substantial than the weak heartbeat signal. Additionally, these estimation methods rely on sensing small physiological movements, any random body movement can produce a significant source of noise further complicating accurate HR estimation [18]. This limitation is a key factor restricting the broad application of radar-based sensors for non-contact heartbeat detection to some extent. In view of this, a feasible and effective solution to this problem is to implement some appropriate noise reduction algorithms to the dataset before training DL models. Reducing or eliminating noise is crucial, as clean, noise-free data significantly enhances model training performance. By minimizing environmental noise, DL models can learn from high-quality, relevant features, leading to better generalization and improved accuracy [4]. Based on the above analysis and discussion, we can now address Research Question 1: Utilizing time series-based deep learning models to learn and predict heart rate from radar raw data exhibits inferior performance, even worse than simply using the average of the reference heart rate as the prediction.

We would like to note that despite we have conducted numerous experiments and found out the main factors that affect the performance of DL models, there are still some limitations to the evaluation in this study. First of all, in the choice of setting parameters for DL models, we only chose a range of values of the experiment parameters based on our knowledge and expertise, and we do not know whether these parameters can represent the optimal settings for predicting

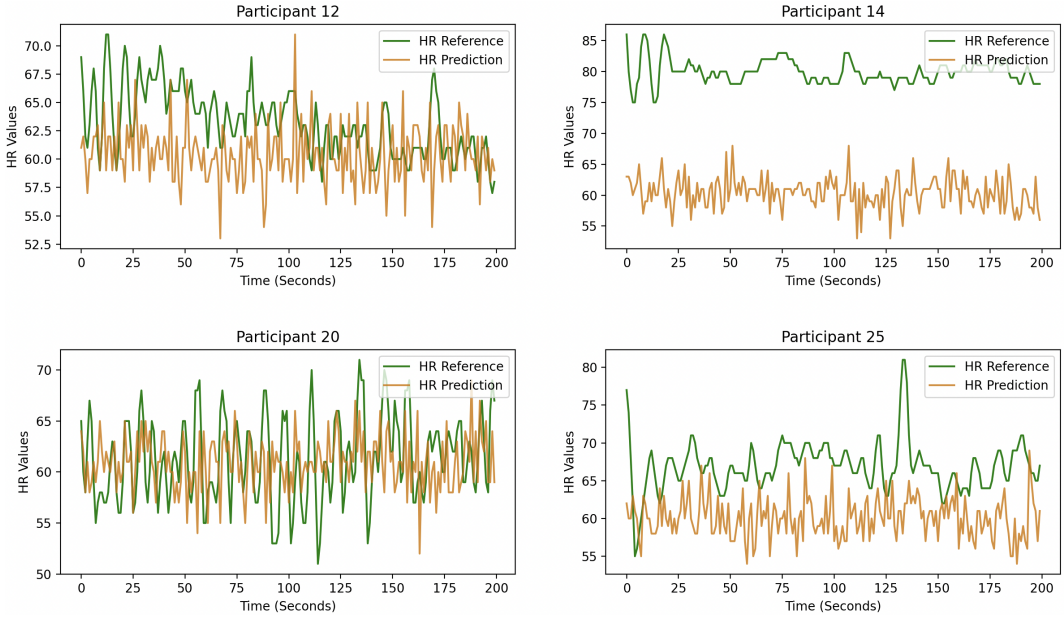

Fig. 10. The prediction results for participants 12, 14, 20, and 25 using the BiLSTM model (Participants 12, 14, 20, and 25 were manually selected as the validation dataset during the cross-validation phase to represent the overfitting phenomenon of the overall dataset).

heart rate. Secondly, in the selection of window size, we chose 5 seconds as the window size to process datasets, a longer period of window size might also potentially impact the overall model performance. What is more, in the selection of DL models, we only selected 7 time series-based DL models for evaluation, and there might be better DL models that are more effective at identifying heart rates from radar raw data.

5 CONCLUSION AND FUTURE WORK

In this paper, we evaluate the performance of serval time series-based DL models, including LSTM, BiLSTM, GRU, RNNED, CnnLSTM, TPALSTM, and N-BEATES, to predict heart rate, utilizing a public database of radar-recorded vital signs, and we conduct the dataset from resting scenario for the training and validation dataset. We applied the cross-validation technique across our experiments. In the evaluation phase, we evaluate the overall performance of DL models using four metrics: MSE, RMSE, MAE, and R^2. The overall results address the Research Questions, indicating that the BiLSTM model outperforms other DL models. However, all 7 evaluated DL models exhibit very poor performance due to the impact of the environmental noise. Therefore, we suggest that appropriate noise reduction algorithms should be applied to mitigate the impact of environmental noise before training DL models for predicting HR based on radar-recorded data.

As part of future work, we will propose employing different noise reduction algorithms, such as band-pass filters and wavelet denoising, to reduce the environmental noise before training DL models to increase the overall model performance. The proposed future work is as follows:

(1) Integrate noise reduction algorithms into the process of data preprocessing for the training and validation dataset before training DL models.
(2) Evaluate the performance of various DL models after applying noise reduction algorithms.

(3) Assess the performance metrics of each DL model with and without applying noise reduction algorithms to quantify the impact of the noise reduction.

Also, if we can get satisfactory performance results, we will aim to investigate the potential applications of these DL models across various domains, including smart home systems for elderly care and medical applications.

REFERENCES

[1] Khaled A Althelaya, El-Sayed M El-Alfy, and Salahadin Mohammed. 2018. Evaluation of bidirectional LSTM for short-and long-term stock market prediction. In *2018 9th international conference on information and communication systems (ICICS)*. IEEE, 151–156.

[2] Daniel Berrar et al. 2019. Cross-validation.

[3] Omar Boursalie, Reza Samavi, and Thomas E Doyle. 2021. Evaluation metrics for deep learning imputation models. In *International Workshop on Health Intelligence*. Springer, 309–322.

[4] Vicente Carrasco, Jorge P. Arenas, Pablo Huijse, Diego Espejo, Victor Vargas, Rhoddy Viveros-Muñoz, Victor Poblete, Matthieu Vernier, and Enrique Suárez. 2023. Application of Deep Learning to Enforce Environmental Noise Regulation in an Urban Setting. *Sustainability* 15, 4 (2023). https://doi.org/10.3390/su15043528

[5] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *Peerj computer science* 7 (2021), e623.

[6] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[8] Sang Ho Choi and Heenam Yoon. 2023. Convolutional Neural Networks for the Real-Time Monitoring of Vital Signs Based on Impulse Radio Ultrawide-Band Radar during Sleep. *Sensors* 23, 6 (2023). https://doi.org/10.3390/s23063116

[9] Lingdong Feng and Yubin Miao. 2023. Intelligent Heart Rate Extraction Method Based on Millimeter Wave Radar. *Journal of Shanghai Jiaotong University (Science)* TBD, TBD (28 October 28 2023), TBD. https://doi.org/10.1007/s12204-023-2656-1

[10] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* 18, 5-6 (2005), 602–610.

[11] Khushi Gupta, Srinivas M. B., Soumya J, Om Jee Pandey, and Linga Reddy Cenkeramaddi. 2022. Automatic Contact-Less Monitoring of Breathing Rate and Heart Rate Utilizing the Fusion of mmWave Radar and Camera Steering System. *IEEE Sensors Journal* 22, 22 (2022), 22179–22191. https://doi.org/10.1109/JSEN.2022.3210256

[12] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[13] Alipujiang Jierula, Shuhong Wang, Tae-Min Oh, and Pengyu Wang. 2021. Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. *Applied Sciences* 11, 5 (2021), 2314.

[14] Artúr István Károly, Péter Galambos, József Kuti, and Imre J Rudas. 2020. Deep learning in robotics: Survey on model structures and training strategies. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51, 1 (2020), 266–279.

[15] Faheem Khan, Stéphane Azou, Roua Youssef, Pascal Morel, and Emanuel Radoi. 2022. IR-UWB radar-based robust heart rate detection using a deep learning technique intended for vehicular applications. *Electronics* 11, 16 (2022), 2505.

[16] Zongquan Ling, Weinan Zhou, Yuhao Ren, Jiping Wang, and Liquan Guo. 2022. Non-Contact Heart Rate Monitoring Based on Millimeter Wave Radar. *IEEE Access* 10 (2022), 74033–74044. https://doi.org/10.1109/ACCESS.2022.3190355

[17] Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun, and Jingyang Wang. 2020. A CNN-LSTM-based model to forecast stock prices. *Complexity* 2020 (2020), 1–10.

[18] Marco Mercuri, Ilde Rosa Lorato, Yao-Hong Liu, Fokko Wieringa, Chris Van Hoof, and Tom Torfs. 2019. Vital-sign monitoring and spatial tracking of multiple people using a contactless radar-based sensor. *Nature Electronics* 2, 6 (2019), 252–262.

[19] Sajjad Mozaffari, Omar Y Al-Jarrah, Mehrdad Dianati, Paul Jennings, and Alexandros Mouzakitis. 2020. Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Transactions on Intelligent Transportation Systems* 23, 1 (2020), 33–47.

[20] Rafid Umayer Murshed, Md Abrar Istiak, Md Toufiqur Rahman, Zulqarnain Bin Ashraf, Md Saheed Ullah, and Mohammad Saquib. 2023. A CNN based multifaceted signal processing framework for heart rate proctoring using millimeter wave radar ballistocardiography. *Array* 20 (2023), 100327.

[21] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437* (2019).

[22] Justin Saluja, Joaquin Casanova, and Jenshan Lin. 2020. A Supervised Machine Learning Algorithm for Heart-Rate Detection Using Doppler Motion-Sensing Radar. *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology* 4, 1 (2020), 45–51. https://doi.org/10.1109/JERM.2019.2923673

[23] Stefan Schellenberger, Kefan Shi, Thomas Steigleder, et al. 2020. A dataset of clinically recorded radar vital signs with synchronised reference sensor signals. *Scientific Data* 7, 1 (2020), 291. https://doi.org/10.1038/s41597-020-00629-5

[24] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. 2019. Temporal pattern attention for multivariate time series forecasting. *Machine Learning* 108 (2019), 1421–1441.

[25] Lindsay I Smith. 2002. A tutorial on Principal Components Analysis. (2002).

[26] Yu-Chiao Tsai, Shih-Hsuan Lai, Ching-Ju Ho, Fang-Ming Wu, Lindor Henrickson, Chia-Chien Wei, Irwin Chen, Vincent Wu, and Jyehong Chen. 2020. High accuracy respiration and heart rate detection based on artificial neural network regression. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 232–235.

[27] Yu-Chiao Tsai, Shih-Hsuan Lai, Ching-Ju Ho, Fang-Ming Wu, Lindor Henrickson, Chia-Chien Wei, Irwin Chen, Vincent Wu, and Jyehong Chen. 2020. High Accuracy Respiration and Heart Rate Detection Based on Artificial Neural Network Regression. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*. 232–235. https://doi.org/10.1109/EMBC44109.2020.9175161

[28] Haili Wang, Fuchuan Du, Hao Zhu, Zhuangzhuang Zhang, Yizhao Wang, Qixin Cao, and Xiaoxiao Zhu. 2023. HeRe: Heartbeat Signal Reconstruction for Low-Power Millimeter-Wave Radar Based on Deep Learning. *IEEE Transactions on Instrumentation and Measurement* 72 (2023), 1–15. https://doi.org/10.1109/TIM.2023.3267348

[29] Hongqiang Xu, Malikeh P Ebrahim, Kareeb Hasan, Fatemeh Heydari, Paul Howley, and Mehmet Rasit Yuce. 2021. Accurate heart rate and respiration rate detection based on a higher-order harmonics peak selection method using radar non-contact sensors. *Sensors* 22, 1 (2021), 83.

[30] Sijie Yang, Fei Zhu, Xinghong Ling, Quan Liu, and Peiyao Zhao. 2021. Intelligent health care: Applications of deep learning in computational medicine. *Frontiers in Genetics* 12 (2021), 607471.

[31] Zi-Kai Yang, Sheng Zhao, Xiang-Dong Huang, and Wei Lu. 2020. Accurate Doppler radar-based heart rate measurement using matched filter. *IEICE Electronics Express* 17, 8 (2020), 20200062–20200062.

[32] Haoyu Zhang. 2020. Heartbeat monitoring with an mm-wave radar based on deep learning: a novel approach for training and classifying heterogeneous signals. *Remote Sensing Letters* 11, 11 (2020), 993–1001. https://doi.org/10.1080/2150704X.2020.1809735 arXiv:https://doi.org/10.1080/2150704X.2020.1809735

[33] Wenyu Zhang, Gang Li, Zetao Wang, and Hao Wu. 2022. Non-contact monitoring of human heartbeat signals using mm-wave frequency-modulated continuous-wave radar under low signal-to-noise ratio conditions. *IET Radar, Sonar & Navigation* 16, 3 (2022), 456–469.

[34] Peijun Zhao, Chris Xiaoxuan Lu, Bing Wang, Changhao Chen, Linhai Xie, Mengyu Wang, Niki Trigoni, and Andrew Markham. 2020. Heart Rate Sensing with a Robot Mounted mmWave Radar. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2812–2818. https://doi.org/10.1109/ICRA40945.2020.9197437