



---

**Research article**

## Edge-assisted multi-user millimeter-wave radar for non-contact blood pressure monitoring

**Xu Ji<sup>1</sup>, Fang Dong<sup>1,\*</sup>, Zhaowu Huang<sup>1</sup>, Xiaolin Guo<sup>1</sup>, Haopeng Zhu<sup>1</sup>, Baijun Chen<sup>1</sup> and Jun Shen<sup>2</sup>**

<sup>1</sup> School of Computer Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, China

<sup>2</sup> School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia

\* Correspondence: Email: [fdong@seu.edu.cn](mailto:fdong@seu.edu.cn).

Academic Editor: Pasi Fränti

**Abstract:** The non-contact blood pressure (BP) monitoring technology based on millimeter wave radar (mmWave) has been widely concerned for its advantages of non-invasive and real-time continuous monitoring. In recent years, studies have employed deep learning technologies to process mmWave radar, providing high-accuracy monitoring and high computing resource requirements. In this paper, we propose an edge-assisted framework for mmWave radar-based blood pressure monitoring to meet high accuracy and low latency application requirements because edge computing can provide a more powerful computing capability closer to users. However, it is non-trivial to effectively run such an edge-assisted mmWave radar-based blood pressure monitoring upon multiple users due to limited edge server resources. To solve this problem, we identify an opportunity to optimize the inference efficiency by adjusting key system parameters, such as sampling interval and input signal sequence length. This adjustment helps to reduce the inference latency and resource contention, especially in resource-constrained edge computing environments. By adaptively configuring these parameters for multiple users, we aim to strike a balance between a high accuracy and a low latency. First, we formulate the problem as an online learning problem and propose a deep reinforcement learning-based method to solve it. Finally, we implement a testbed to evaluate the performance of our method. Extensive experimental results show that our method outperforms the baselines, achieving a latency reduction of up to 70.3% and improving a reward by up to 29.7%, while maintaining an accuracy loss within 5%.

**Keywords:** millimeter-wave radar; blood pressure monitoring; edge computing; latency-sensitive; deep learning

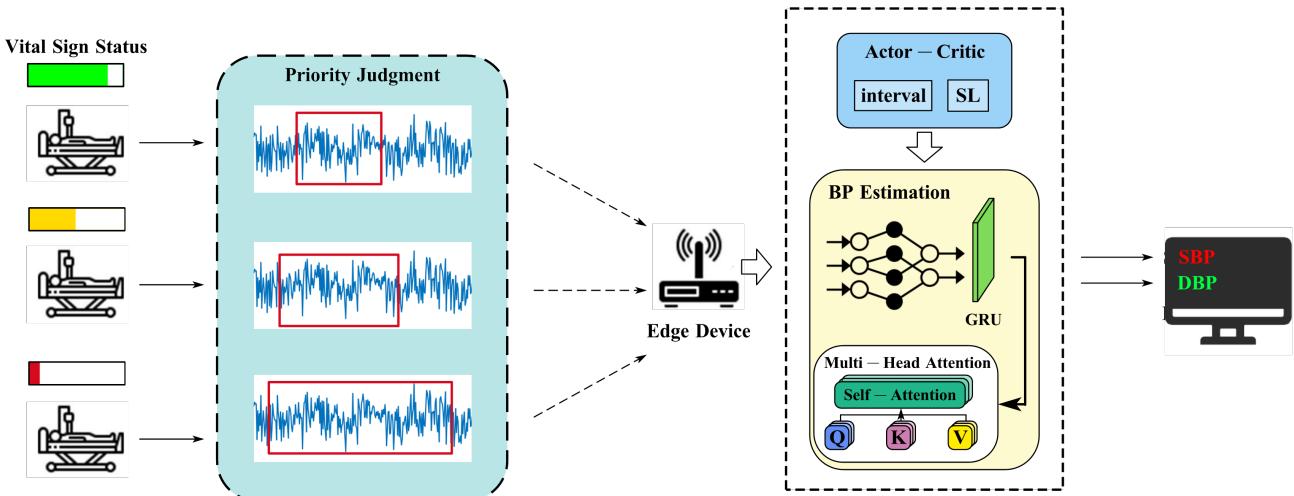
---

## 1. Introduction

Blood pressure (BP) is a critical physiological parameter to evaluate human health and assess cardiovascular conditions [1]. Accurate and continuous BP monitoring is indispensable for early hypertension detection, treatment efficacy evaluation, and the prevention of cardiovascular diseases. However, BP is inherently dynamic, and is influenced by factors such as physical activity, emotional stress, and circadian rhythms [2]. Traditional intermittent BP measurements frequently fail to capture essential fluctuations, such as morning and nocturnal hypertension, which have been strongly associated with cardiac complications [3] and an increased risk of stroke, heart failure, and kidney dysfunction [4]. The lack of real-time, continuous BP data can lead to impaired clinical decision-making and delayed treatments, particularly for hypertensive patients and individuals in critical care.

Over the past few years, millimeter-wave (mmWave) radar technology has shown significant promise in non-contact BP monitoring. Early work using conventional machine learning algorithms (e.g., regression models) validated the feasibility of extracting BP-related features from mmWave signals [5, 6]. Subsequently, deep learning techniques—such as convolutional neural networks (CNNs) [7, 8] and recurrent neural networks (RNNs) [9]—have enhanced the measurement capabilities by modeling complex spatial and temporal features in radar data. More recently, transformer-based architectures have demonstrated notable success in capturing long-range dependencies and improving the predictive accuracy [10].

Although mmWave radar technology based on deep learning enables high-precision, non-invasive BP monitoring, it imposes significant computational demands, posing challenges for continuous high-accuracy BP monitoring on resource-limited devices. Therefore, we seek an edge-assisted approach to achieve high-accuracy, low-latency reasoning for mmWave radar-based BP monitoring. In edge-assisted mmWave radar-based BP monitoring, the device collects the millimeter wave signal and transmits this information to the edge server, which performs deep learning reasoning, as shown in Figure 1. This approach is feasible due to the widespread adoption of edge computing, which enables high-precision and low-latency deep learning reasoning.



**Figure 1.** System design.

However, it is non-trivial to effectively run such an edge-assisted mmWave radar-based BP

---

monitoring upon multiple users. There is an increased computing latency because the edge server has limited computing resources and a large number of users submitting inference requests at the same time will cause resource contention.

We see an opportunity to optimizew the inference efficiency by adjusting key system parameters, such as the sampling interval and the input signal sequence length (SL), which can play a new role in efficient mmWave radar-based BP monitoring. We refer to a combination of specific parameter values as a configuration. Specifically, we can use a non-uniform sampling interval and the model sequence length to reduce the contention of a large number of users for the computing resources. However, we find that naively setting up such configurations, including the sampling interval and the sequence lengths, is inefficient. According to the pre-experimental results (shown in Section 2), a higher sampling frequency and a longer sequence length can improve the inference accuracy; however, it can bring a longer inference delay at the same time. It is a non-trivial problem to set up the proper configuration to balance the accuracy and the latency, which encounters two challenges: (1) the configuration space is huge, and using a brute force search will bring considerable time overhead; and (2) the priority level of each user is different. If the sampling interval and the input signal sequence length are consistent, then some users with severe conditions may not receive timely and adequate BP monitoring, thus delaying critical treatment or intervention opportunities and increasing the risk of disease deterioration. Therefore, the methodology to select the appropriate configuration in the edge computing environment to meet the reasoning needs of multiple users with different priorities is a difficult problem.

To address these challenges, we propose a novel edge-assisted mmWave radar framework for non-contact BP monitoring (EMMRBP). The overall design is shown in Figure 1. The framework implements adaptive configuration settings for multiple users. We formulate the problem as an online learning problem to balance high accuracy and low latency by selecting the sampling interval and the sequence length. We analyze the difficulty of the problem and propose a method based on deep reinforcement learning to solve it. Finally, we implement a testbed to evaluate the performance of EMMRBP. Extensive experimental results show that EMMRBP outperforms the baselines, achieving a significant enhancement in real-time BP estimation compared to the existing methods.

We summarize our contributions as follows:

- To the best of our knowledge, we are the first to investigate edge-assisted mmWave radar-based BP monitoring for multiple users.
- We propose a DRL-based online learning method to select the appropriate configuration.
- We implement a real-world test bed and conduct intensive experiments to evaluate our method. The results show that the EMMRBP significantly outperforms the baselines.

## 2. Preliminary and motivation

### 2.1. Preliminary

In this section, we present the precise definitions and clarifications of key concepts essential to the paper, with the aim of enhancing the conceptual clarity for those less acquainted with deep learning terminology.

**Sampling Interval:** The sampling interval refers to the time difference between consecutive data points collected from the continuous mmWave signal captured by the radar devices. In our study, it

---

determines the frequency at which observations are recorded from the radar signal, which is crucial for the accuracy and timeliness of BP estimation. Typically, the sampling interval in our system ranges from 0 to 100 to ensure a balance between the data resolution and the computational efficiency.

**Sequence Length:** The sequence length refers to the number of time steps or elements in a temporal sequence, representing the length of the radar signal segment used for BP estimation. In our study, it determines how much historical data is considered in each prediction. The sequence length is a critical parameter for models such as transformer, as it impacts both the accuracy and the computational load of the BP estimation process. In our case, the sequence length typically ranges from 0 to 1000, depending on the computational capacity of the edge server and the desired trade-off between the accuracy and the latency.

**Multi-Head Attention (MHA):** MHA is a mechanism used in transformer architectures to allow the model to simultaneously focus on different parts of the input sequence [11]. It consists of multiple attention heads, each capturing distinct patterns in the data. MHA enables the model to capture long-range dependencies and intricate temporal relationships in the mmWave radar signals, which are crucial for accurate BP estimation.

**Actor-Critic:** Actor-critic is a reinforcement learning algorithm that combines two components: the actor, which decides actions, and the critic, which evaluates the actions taken. This approach aims to balance exploration and exploitation in decision-making.

**Reward/Advantage Functions:** In reinforcement learning, the reward function provides feedback on the success of an action, while the advantage function measures how much better a specification is compared to the average action in a given state. These functions are essential for the learning process.

## 2.2. Motivation

We developed a testbed to analyze the impact of interval and sequence length selection on the performance of BP estimation using the mmWave radar signal. The primary purpose of the preliminary experiments was to evaluate the feasibility of using sequence length and interval configurations to optimize the trade-off between accuracy and latency in BP estimation tasks. For edge computing, we employed an NVIDIA Jetson TX2 as the edge server, equipped with a Dual-core NVIDIA Denver™ 2 64-bit CPU, a quad-core Arm® Cortex®-A57 MPCore processor, and a 256-core NVIDIA Pascal™ architecture GPU, with a computational capability of 1.33 TFLOPS. The NVIDIA Jetson TX2 platform, developed by NVIDIA Corporation, was used for the experiments. More details can be found at \*. The test dataset was PPG-BP Database, and the detection model utilized was based on multi-head attention. The dataset “PPG-BP Database” is publicly available on Figshare at †.

Our investigation primarily focused on two aspects: first, the influence of the sequence length on the performance of a transformer-based model for BP estimation; and second, the effect of interval selection. Notably, since transformer models traditionally require fixed-length input during training, we trained separate models for each sequence length to ensure a compatibility with our pre-experiments and subsequent comprehensive experiments. The choice of using fixed-length inputs is motivated by the need to maintain a consistent data structure and computational efficiency during training. While modern transformer variants, such as Longformer or Transformer-XL, can handle variable-length sequences, we chose a fixed-length approach in our study to simplify the model implementation

---

\*<https://developer.nvidia.com/embedded/jetson-tx2>

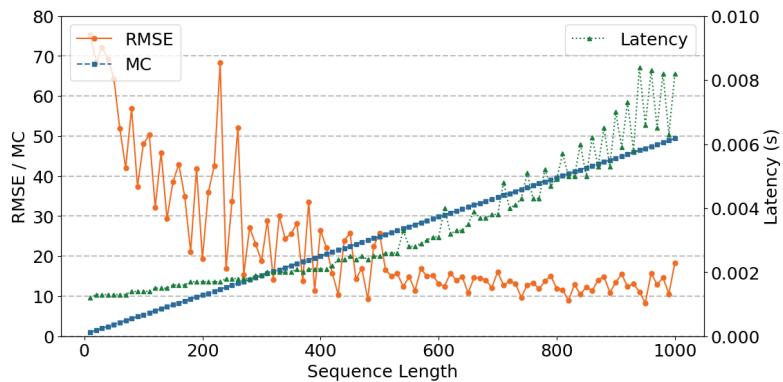
†[https://figshare.com/articles/dataset/PPG-BP\\_Database\\_zip/5459299](https://figshare.com/articles/dataset/PPG-BP_Database_zip/5459299)

and ensure that all input sequences were uniformly processed, thus reducing the complexity and the potential model instability.

This study provides valuable insights into optimizing the performance of transformer-based models for biomedical signal processing and highlights the potential of mmWave radar and edge computing in advancing BP estimation methodologies.

### 2.2.1. The impact of sequence length on accuracy and latency in edge biomedical signal processing

In Figure 2, we vary the sequence length of input detection from 0 to 1000 to explore its impact on the accuracy, inference time, and memory usage. Our findings reveal that as the sequence length increases, the root mean square error (RMSE), which is used as a metric for accuracy, gradually decreases (lower is better). However, this improvement comes at the cost of an increased latency, which can reach up to  $6.91 \times$  the original value, along with a linear increase in the memory consumption. This is due to the fact that an increase in the sequence length results in more parameters being processed by the model, thus leading to a higher computational load and a greater overhead.



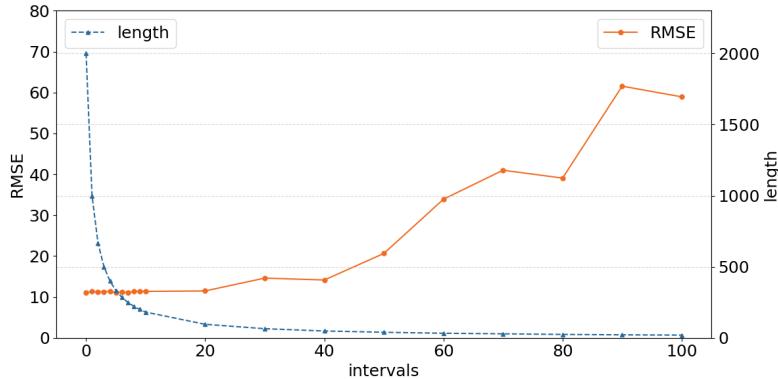
**Figure 2.** Influence of sequence length on RMSE, memory consumption and latency.

Furthermore, we observe that while memory consumption (MC) linearly grows with the sequence length; the changes in the RMSE and the inference time do not follow a linear pattern. Instead, these metrics exhibit distinct trends, with their variations becoming more gradual as the sequence length either increases or decreases. For the RMSE, when the sequence length exceeds approximately 500, the fluctuations in the accuracy become minimal, and further improvements plateau. In contrast, for the inference latency, when the sequence length drops below approximately 500, the variations stabilize and the fluctuations become less pronounced. Therefore, selecting an appropriate sequence length to balance the accuracy and latency, while ensuring that the overall computational overhead of the algorithm remains within the processing capabilities of the edge, is one of the key research objectives of this study.

### 2.2.2. The impact of sampling interval on accuracy and latency in edge biomedical signal processing

In Figure 3, we vary the interval size from 0 to 100, while taking values from the set  $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$  for the calculation. The interval represents the sampling interval for a sequence; for example, for a sequence of length 2000, an interval of 0 corresponds to a sampled sequence length of 2000, while an interval of 100 corresponds to a sampled

sequence length of 20.



**Figure 3.** Influence of interval on RMSE and sequence length.

Based on the results shown in the figure, it is evident that as the interval increases, the sampled sequence length decreases, while the RMSE rises. Furthermore, within the interval range of 0–20, the RMSE remains almost unchanged, whereas the sampled sequence length significantly decreases.

Therefore, selecting an appropriately sized interval to dramatically reduce the sequence length required for inference while minimizing the accuracy loss is one of the critical research problems addressed in this study.

**Conclusions:** In the two preliminary experiments, we investigated the impact of interval and sequence length selection on the performance of BP estimation using mmWave radar data. It is important to note that the choice of interval directly affects the initial sequence length. Consequently, once the interval is determined, the selected sequence length cannot exceed the length of the adjusted sequence.

Ultimately, the objective of this study is to collaboratively optimize both the interval and the sequence length to achieve a balance between the latency and accuracy of BP estimation using mmWave radar data, while ensuring that the computational overhead does not exceed the device's processing capacity.

### 3. Model design and system architecture

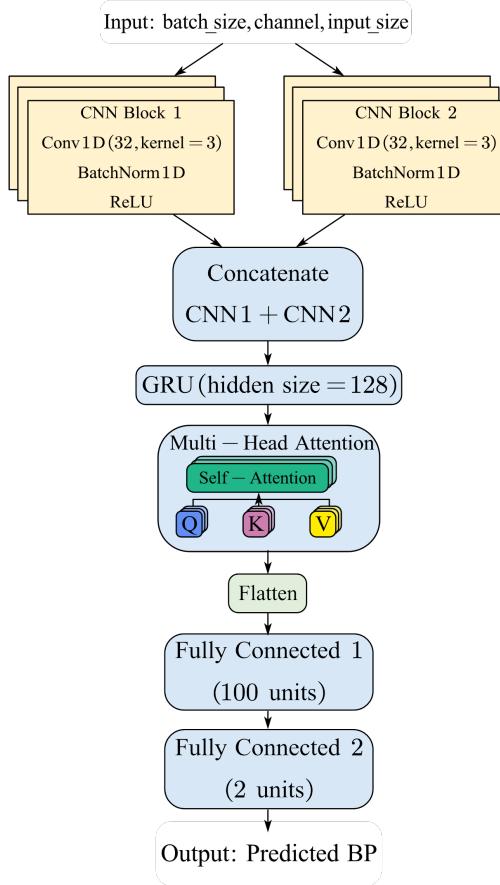
#### 3.1. Network architecture for BP prediction

In this work, we propose a neural network architecture based on the multi-head attention mechanism for our BP estimation task. As shown in Figure 4, the model includes a CNN for feature extraction, gated recurrent units for temporal modeling, multi-head attention to capture long-range dependencies, and a fully connected layer for the final predictions.

The input can be represented as  $X \in \mathbb{R}^{B \times C \times T}$ , where  $T$  is the length of the time-series sequence. The sequence represents a pulse wave captured over time, with each value in the sequence representing a particular point in the signal. We apply two convolutional layers to capture both the local and global temporal patterns in the signal followed by Eqs (1) and (2):

$$Y_1 = \text{Conv1d}(X, 3), \quad (1)$$

$$Y_2 = \text{Conv1d}(X, 5). \quad (2)$$



**Figure 4.** The architecture of transformer-based blood pressure estimation network.

The kernel sizes are 3 and 5, respectively. We use a gated recurrent unit (GRU) to capture the sequential patterns in our data. The GRU takes the concatenated feature maps from the two convolutional layers as input and outputs a sequence of hidden states. The GRU update is defined by Eqs (3)–(6):

$$r_t = \sigma(W_r X_t + U_r h_{t-1}), \quad (3)$$

$$z_t = \sigma(W_z X_t + U_z h_{t-1}), \quad (4)$$

$$\tilde{h}_t = \tanh(W_h X_t + U_h(r_t \odot h_{t-1})), \quad (5)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \quad (6)$$

Then, we incorporate a Multi-Head Attention (MHA) mechanism, thus allowing the model to focus on important features in the time-series sequence. The GRU output  $H$  is linearly transformed into queries  $Q$ , keys  $K$ , and values  $V$ :

$$Q, K, V = \text{Linear}(H). \quad (7)$$

The attention weights can be calculated by  $A = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)$ , where  $d$  is the dimension of  $Q$  and  $K$ .  $O$  represents the weighted sum of  $A$ , and  $O = A \times V$ . The final output can be represented as Eq (8):

$$O_{\text{final}} = \text{Concat}(O_1, O_2, \dots, O_h)W_O. \quad (8)$$

### 3.2. Preprocessing steps

For an accurate BP estimation using transformer models, the following standard preprocessing steps are typically applied:

**Token Embedding:** The radar signals are first segmented into time steps or windows, each representing a distinct temporal feature. Then, these segments are converted into embeddings, which capture the temporal patterns within the radar data.

**Positional Encoding:** As transformer models do not have an intrinsic understanding of the sequential order of the data, positional encoding is added to the token embeddings. This encoding allows the model to learn the relative positions of tokens in the sequence, which is crucial for capturing the temporal dependencies in radar signals that are essential for an accurate BP estimation.

**Normalization:** To ensure that the input features are on a comparable scale, we applied a normalization technique such as z-score normalization. This step helps to enhance the stability and convergence of the model during training by preventing any feature from dominating the learning process due to differing scales.

These preprocessing steps are critical for preparing the radar signal data, thereby enabling the transformer model to effectively capture the temporal dependencies and variations that underpin an accurate BP estimation.

### 3.3. System model

We consider an edge-assisted mmWave radar BP estimation system that consists of  $N$  mmWave radar end devices and an edge server. These radar end devices are connected to the edge server via access points over a network. Each radar end device is responsible for collecting signal from a corresponding patient to analyze their BP. On the edge server, we deploy the transformer model proposed in Figure 4 to measure the patient's BP based on the collected signal. The computing capacity of the edge server is denoted by  $C$ . Since the transmission time of the mmWave signal is extremely short and negligible compared to the inference time of the Transformer model, in this paper, we ignore the transmission time between the end devices and the edge server.

For each patient  $i \in I$ , the collected mmWave signal is denoted as  $\text{signal}_i$ , and the priority level of the patient is represented as  $p_i \in P$ . For instance, patients are categorized into three levels: normal, moderate, and critical, thus indicating different requirements for BP estimation accuracy and latency. When using the transformer model for BP estimation, we need to determine the signal interval  $\text{interval}_i(t)$  and the sequence length  $\text{length}_i(t)$  input to the transformer model at each time step  $t \in T$ . In our transformer model, given a fixed edge server computing capacity  $C$ , a longer sequence length increases the inference latency but can potentially improve the measurement accuracy. Conversely, a shorter sequence length reduces the latency but may lead to an inaccurate BP estimation. We use the functions  $\text{Acc}(\text{length}_i(t))$  and  $f(\text{length}_i(t))$  to calculate the measurement accuracy and latency of the transformer model for a sequence length  $\text{length}_i(t)$ . Note that  $\text{Acc}(\cdot)$  and  $f(\cdot)$  can be determined through empirical measurements on the edge server.

**Problem Definition.** Considering the priority level of the patients, different patients have varying requirements for the latency and accuracy of BP estimation. To balance the accuracy and latency of BP estimation, we need to carefully decide on 1) the interval of the signal and 2) the sequence length input to the Transformer model. In the edge-assisted mmWave BP monitoring system, the BP

estimation delay consists of the signal interval and the Transformer inference delay, which is calculated as follows:

$$d_i(t) = \text{interval}_i(t) + f(\text{length}_i(t)). \quad (9)$$

Since the computing resources available on the edge server are dynamic, we can reduce the sequence length of the input signal to the Transformer model to prevent an excessive Transformer inference delay, thereby lowering the inference delay. The sequence length must satisfy the following:

$$0 \leq \text{length}_i(t) \leq \text{interval}_i(t). \quad (10)$$

Our optimization goal is to balance the cumulative accuracy and latency of BP estimation across multiple patients by deciding on the interval and sequence length of the Transformer model input. Our objective is to provide patients with a higher accuracy and lower latency in BP estimation, which can be formulated as follows:

$$\max \sum_{i \in I} \sum_{t \in T} [\text{Acc}(\text{length}_i(t)) - \alpha d_i(t)] p_i, \quad (11)$$

$$\text{s.t. } 0 \leq \text{length}_i(t) \leq \text{interval}_i(t), \quad (12)$$

$$p_i \in P, i \in I, \quad (13)$$

where  $\alpha$  is a tunable parameter used to balance the trade-off between accuracy and latency in BP estimation. Constraint (12) shows that the input sequence length cannot exceed the sampling interval. Constraint (13) ensures that the priority level of the patients must belong to the set of priority levels and that the patients belong to the patient set. This is a challenging problem due to the limited computing resources available on the edge server, and the inference delay function is unknown and difficult to formalize. Furthermore, the accuracy function is also unknown, and the relationship between the signal length and the inference accuracy is difficult to establish analytically.

#### 4. Deep reinforcement learning-based decision method

Deep reinforcement learning (DRL) has shown significant advantages in solving problems where explicit objective functions are unavailable by learning strategies through trial and error. Therefore, our problem can potentially be addressed using a DRL-based approach [12]. In our problem, DRL learns the decision-making process for the interval and sequence lengths by enabling an agent to interact with the environment. The details of the DRL framework are as follows.

##### 4.1. State

In DRL, the state  $s_t$  describes the condition of the environment at each time step  $t \in T$ . The agent needs to acquire information about the environment, which must contain all the essential information needed for decision-making. For our problem, the state  $s_t = (\vec{c}_t, \vec{bp}_t, \vec{p}_t, \vec{l}_t, \vec{g}_t)$  is defined as follows:

- $\vec{c}_t = \{c_{t-m}, \dots, c_{t-1}\}$ : The edge server's available computing capacity over the past  $m$  time steps, which directly affects the Transformer model's inference delay.

---

- $\vec{bp}_t = \{bp_{t-m}, \dots, bp_{t-1}\}$ : BP data measured for patients over the past  $m$  time steps. More historical BP information helps guide decisions on the interval and sequence lengths at time  $t$ .
- $\vec{p}_t = \{p_{t-m}, \dots, p_{t-1}\}$ : The priority levels of  $I$  patients. Different priority levels may require varying BP estimation accuracies and inference latencies, influencing interval and sequence length decisions.
- $\vec{l}_t = \{l_{t-m}, \dots, l_{t-1}\}$  and  $\vec{g}_t = \{g_{t-m}, \dots, g_{t-1}\}$ : The sequence length and interval over the past  $m$  time steps, serving as guidance for decision-making at time  $t$ .

Considering the past  $m$  time steps allows the DRL agent to capture hidden patterns and dynamics in environmental changes, thereby enhancing the feature extraction.

#### 4.2. Action and policy

At each time step  $t \in T$ , the agent decides which action  $a_t$  to take (i.e., determining the interval and sequence length to balance BP estimation accuracy and latency). A policy  $\pi$  maps states to actions and specifies the probability distribution of all possible actions under a specific state. Given the high dimensionality of the state space in our problem, we design a deep neural network as the policy network to handle the complexity. For the input types of these time sequences, we employ a 1D CNN layer to extract the features along the temporal dimension. Subsequently, we flatten the processed outcomes and pass them to a fully connected layer, where the relationships among the extracted features are learned. Ultimately, these features are directed to a softmax layer to compute the probability distribution across possible actions. By adjusting the neural network parameters  $\theta$ , the policy is optimized. Once trained, the deep neural network generates appropriate interval and sequence lengths based on the environmental state at each time step  $t$ .

#### 4.3. Reward

In the DRL framework, the agent receives a reward from the environment for each action taken at time  $t$ . The training process in DRL aims to maximize the cumulative reward. To align the reward with our optimization objective, we design the reward function as follows:

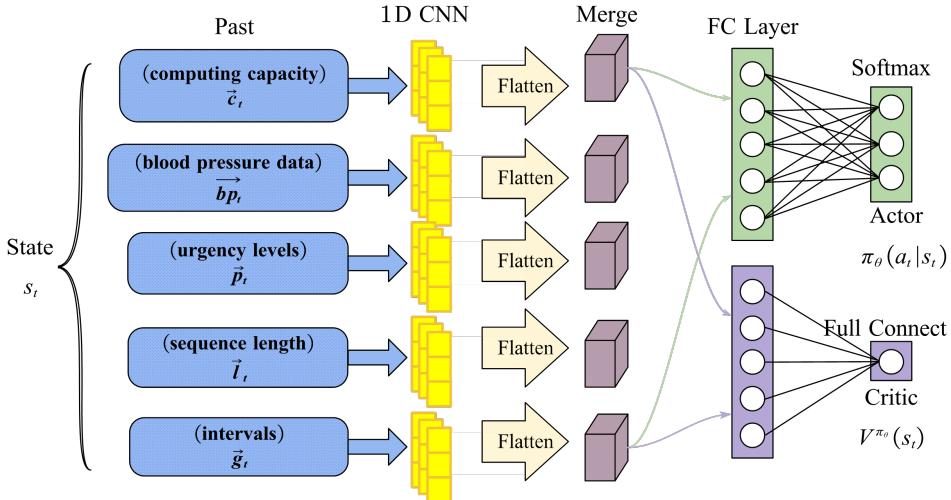
$$R_t = \sum_{i \in I} (\text{Acc}(\text{length}_i(t)) - \alpha d_i(t)) p_i. \quad (14)$$

Here,  $\alpha$  is a parameter that can be adjusted to manage the balance between accuracy and latency.

#### 4.4. DRL framework

We adopt an Actor-Critic framework in Figure 5, which combines the advantages of the policy network (Actor) and the value network (Critic).

- Actor: Responsible for generating actions and interacting with the environment. It represents the policy function and aims to maximize the expected cumulative reward.
- Critic: Evaluates the performance of the Actor by estimating the value function of the current policy. It provides feedback to help the Actor improve its policy.



**Figure 5.** Actor-Critic framework.

In Actor-Critic algorithms, the Critic helps stabilize training by enabling single-step parameter updates without waiting for an episode to end. The Actor updates its policy network parameters using the value information provided by the Critic to adjust toward higher rewards. Simultaneously, the Critic updates its value network parameters based on environmental feedback and new states to improve its value estimations. To address the instability caused by large policy updates in traditional policy gradient methods, we employ the proximal policy optimization (PPO) algorithm. PPO ensures stable training by constraining the policy updates within a predefined range.

We implement the Actor and Critic networks, which are jointly optimized through a cooperative rather than an adversarial approach. The Actor (policy network) and Critic (value network) share feature extraction layers but maintain separate output heads, thus enabling a parameter efficiency while learning distinct objectives. During training, both components are simultaneously updated using a combined loss function:

The Actor is updated via a policy gradient using advantage - weighted log probabilities ( $L_{\text{actor}} = -\log \pi(a|s) \cdot \hat{A}$ ), where the advantage estimate  $\hat{A} = r + \gamma V(s') - V(s)$  is detached from gradient computation to prevent conflicting updates.

The Critic is optimized through temporal difference learning ( $L_{\text{critic}} = \hat{A}^2$ ), thus minimizing the mean squared error of value predictions.

This synchronous update scheme allows the critic to provide stabilized advantage estimates for policy improvement while avoiding adversarial competition between the components. The shared base network facilitates coordinated the feature learning, with gradients from both losses flowing through the common layers. We employ a single Adam optimizer (learning rate  $10^{-3}$ ) for end-to-end optimization, thus ensuring balanced updates across both networks through gradient backpropagation from the aggregated loss  $L_{\text{total}} = L_{\text{actor}} + L_{\text{critic}}$ . The Actor and Critic networks are updated as follows.

We use  $A^{\pi_\theta}$  as the advantage function, which represents the advantage of taking action  $a_t$  in state  $s_t$  under policy  $\pi_\theta$  relative to the average action. The advantage function measures how much better an action is compared to others. The advantage estimate  $A_t$  is defined as follows:

$$A_t = r_t + \gamma V^{\pi_\theta}(s_{t+1}) - V^{\pi_\theta}(s_t), \quad (15)$$

where  $r_t$  is the immediate reward,  $\gamma$  is the discount factor, used to balance the contribution of current and future rewards, and  $V^{\pi_\theta}(s_t)$  and  $V^{\pi_\theta}(s_{t+1})$  are the state value functions, representing the value estimates of the current and next states, respectively to update the Actor network, PPO uses the clipped surrogate objective  $L_{\text{CLIP}}$ , which is defined as follows:

$$L_{\text{CLIP}}(\theta) = \mathbb{E}_t [\min (r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t)], \quad (16)$$

where  $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$  is the ratio of the new policy's probability to the old policy's probability.  $\epsilon$  is the clipping range for the policy update, which is typically set to  $\epsilon = 0.2$ .  $\text{clip}(\cdot)$  restricts  $r_t(\theta)$ 's update range to prevent large updates that could destabilize training.

By optimizing  $L_{\text{CLIP}}$ , the parameters of the Actor network are updated, thus resulting in a stable policy improvement. Using  $V_\phi^\pi$  to denote the output value of the Critic network under policy  $\pi_\phi$ , the Critic network is updated by minimizing the following mean squared error (MSE) loss function:

$$L_{\text{Critic}}(\phi) = \mathbb{E}_t \left[ (V_\phi^\pi(s_t) - \hat{R}_t)^2 \right], \quad (17)$$

where  $V^{\pi_\theta}(s_t)$  is the output value of the Critic network (i.e., the state value estimate).  $\hat{R}_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$  denotes the cumulative return estimate, thus representing the total accumulated reward starting from the time step  $t$ . By jointly updating the Actor and Critic networks, PPO achieves a stable and efficient policy and value optimization.

## 5. Experimental settings

### 5.1. Radar system settings

The system uses a 24GHz mmwave radar sensor installed 30 cm above the human chest and abdomen. The radar transmitter operates at a center frequency of 7.29 GHz with a bandwidth of 0.5 GHz. The receiver samples the reflected signal at a rate of 23.328 GS/s and digitizes the radar signal at 60 fps. MATLAB 2022a is used for data acquisition, processing, and storage from the radar. As shown in Figure 6, the system adopts a hardware design for BP monitoring based on bio-radar technology. The hardware setup consists of a Saiyang J1900 industrial controller, a gigabit network router, and a Wi-Fi communication module, thus achieving a sampling rate of up to 1000 Hz for BP estimation. The bio-radar operates in the 24 GHz frequency band. The optimal installation positions of the radar equipment (chest, abdomen, carotid artery) have been verified. The bio-radar operates at a power of 140 mW.



**Figure 6.** Mmwave radar signal processing system.

### 5.2. Experiment settings

As shown in Figure 7, to validate the performance of EMMRBP, we built a testbed comprised of three categories of physical devices: 1) End Device: Raspberry Pi 4B is used to simulate real-world data collection scenarios; 2) Network Access Point: TP-Link AX6000 router with 5GHz WiFi support is utilized for data transmission; 3) Edge Server: NVIDIA Jetson TX2 is employed for computation at the edge.



**Figure 7.** Experiment environment.

The end device is responsible for simulating the collection of data, which is then offloaded to the edge server for processing. The detection algorithm deployed on the edge is developed based on the PyTorch deep learning framework using an algorithm based on multi-head attention to estimate BP. Our reinforcement learning-based optimization algorithm is implemented in Python and deployed on the edge server.

Our experiments are conducted on the PPG-BP\_Database dataset, where the range of the interval is set to  $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$  and the range of the sequence length is set to  $[10, 20, 30, \dots, 990, 1000]$ . The resulting solution space is comprised of 534 configurations.

For comparison, we implemented the following mechanisms as benchmarks:

- (1) **Ours:** The implementation of our EMMRBP.
- (2) **Greedy Method:** Iteratively selects the current locally optimal solution, quickly approximating the global optimum.
- (3) **SA (Simulated Annealing):** A heuristic algorithm that gradually converges to the global optimum by allowing suboptimal solutions to be accepted with a certain probability.
- (4) **Accuracy Focus:** Prioritizes the solution with the highest accuracy in the solution space, ignoring latency and computational constraints.
- (5) **Latency Focus:** Prioritizes the solution with the lowest inference latency, ignoring accuracy losses.

### 5.3. Metrics

To assess the accuracy of our BP estimation, we employed the RMSE, the formula for which is given as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (18)$$

where  $n$  is the number of observations,  $y_i$  represents the observed values, and  $\hat{y}_i$  are the predicted values.

Subsequent experiments demonstrated the reward values (reward), accuracy (RMSE), and the combined metric ( $\alpha \times$  latency) of our method, EMMRBP, alongside four comparative experiments. Here,  $\alpha$  is used to weigh the priority between the accuracy and latency; in this paper, it is set to 3000. Given that a lower RMSE value indicates a higher accuracy, our reward value is calculated via the following formula:

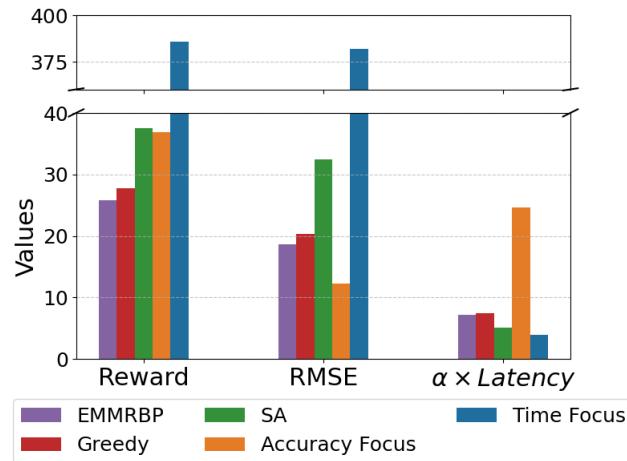
$$\text{reward} = -\text{Accuracy} - \alpha \cdot \text{latency}. \quad (19)$$

This represents the overall experimental outcome. It is worth noting that since the reward is negative, for ease of graphical representation, we plot its negative value. Thus, a smaller (more negative) value on the graph signifies a better overall performance.

## 6. Results

### 6.1. Overall system performance

Figure 8 presents the experimental results of the EMMRBP compared with four other methods: Greedy, SA, Accuracy Focus, and Latency Focus. The comparison focuses on three metrics: Reward, RMSE, and the performance trade-off indicator  $\alpha \times$  latency.



**Figure 8.** Overall experiment on EMMRBP and baseline.

From the Figure 8, we can observe that, the EMMRBP achieves the lowest value compared to all other methods in terms of the Reward, outperforming Greedy, SA, Accuracy Focus, and Latency Focus, which demonstrates its advantage in optimizing the overall performance. For RMSE, the EMMRBP also achieves lower error values, significantly reducing them compared to the Greedy and SA methods, while achieving a level close to Accuracy Focus and Latency Focus. Regarding the  $\alpha \times$  latency metric, the EMMRBP performs the best, with a significantly lower value than Accuracy Focus and Latency Focus and better results compared to Greedy and SA, thus indicating its remarkable advantage in reducing latency.

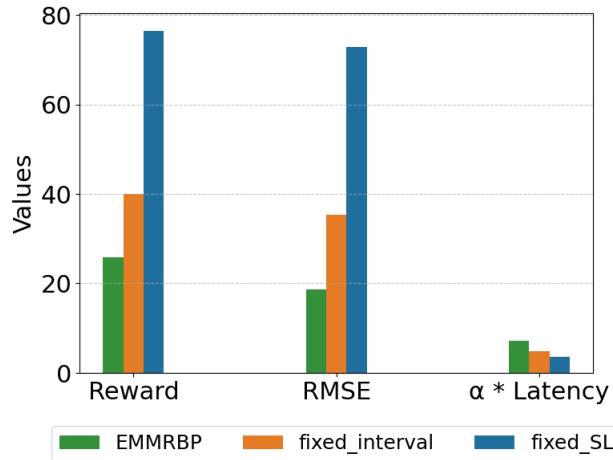
Specifically, the EMMRBP reduces the Reward by approximately 7.01% compared to Greedy and by 31.02% compared to SA. For RMSE, it achieves reductions of 7.40% and 42.39% compared to Greedy and SA, respectively. In the  $\alpha \times$  latency metric, it achieves optimizations of about 4%

and  $-47\%$  compared to Accuracy Focus and Latency Focus, respectively. These results indicate that the EMMRBP maintains a high accuracy while achieving a lower error and latency, thus achieving better balance across multiple performance dimensions.

In conclusion, the EMMRBP achieves significant performance advantages by comprehensively optimizing the reward, error, and latency, thus validating its robustness and applicability in various scenarios and resource-constrained environments.

## 6.2. Ablation study

To comprehensively evaluate the effectiveness of the EMMRBP, we conducted a series of ablation experiments, thereby comparing it with two fixed configuration methods: fixed interval and fixed SL. The baselines fixed interval and fixed SL refer to variants of our own proposed framework. These baselines were implemented to benchmark the performance of our edge-assisted mmWave radar-based BP monitoring system. Notably, the fixed interval aims to minimize the latency, while the fixed SL focuses on maximizing the accuracy. These baselines are intended to demonstrate how varying system parameters can impact the performance of the BP monitoring system. The experiments evaluated the performance across three metrics: Reward, RMSE, and latency-weighted performance ( $\alpha \times \text{latency}$ ), as shown in Figure 9.



**Figure 9.** Resolution ablation experiment on EMMRBP and fixed configuration methods.

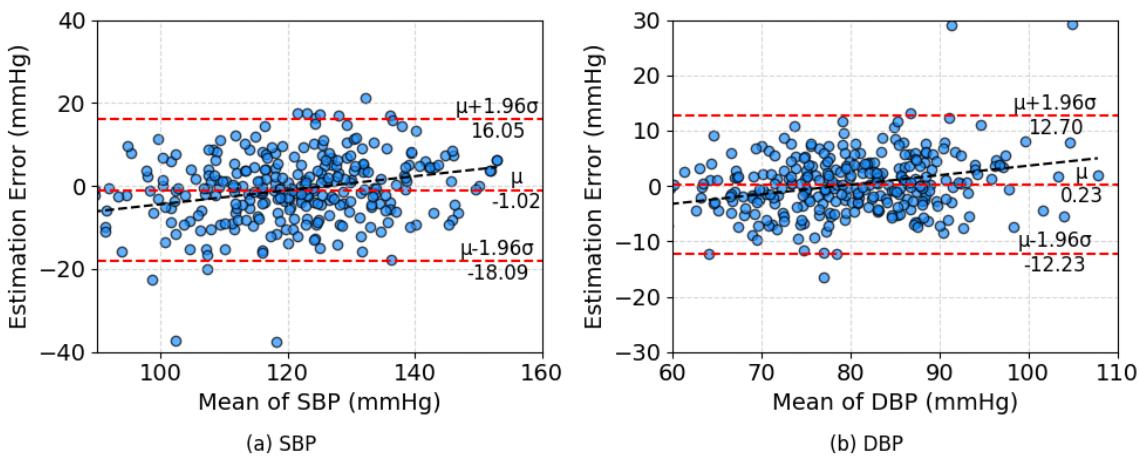
The results demonstrate that the EMMRBP outperforms both fixed configuration methods across all metrics, showcasing its dynamic adaptability and global optimization capabilities. For Reward, the EMMRBP achieved the lowest value, significantly outperforming both the fixed interval and the fixed SL, which focus on single-objective optimization. Regarding RMSE, the EMMRBP effectively reduced the error, further validating its advantage in accuracy and stability. Unlike the fixed configuration methods that focus on single-objective optimization, these findings highlight that the EMMRBP dynamically adjusts its strategy to achieve a balance between accuracy and latency across varying scenarios.

In summary, this experiment validates the flexibility and robustness of the EMMRBP. By dynamically adjusting the resolution and overlapping the padding ratios, it effectively adapts to diverse application environments, significantly enhancing the system performance and paving the way for

broader practical applications.

### 6.3. Bland-Altman analysis

Additionally, this study used the Bland-Altman plot to analyze the consistency between the EMMRBP and the reference device used in the dataset for BP measurement, as shown in Figure 10. The horizontal axis represents the mean BP values from the two methods, and the vertical axis represents the difference in BP values between them. The blue scatter points represent the BP measurement samples, the red dashed line in the middle represents the mean difference ( $\mu$ ) between the two methods, and the upper and lower red dashed lines represent the 95% limits of agreement, calculated as  $\mu \pm 1.96\sigma$ , where  $\sigma$  is the standard deviation of the differences.



**Figure 10.** Bland-Altman plot of estimated blood pressure by EMMRBP.

The results show that the  $\mu$  for systolic blood pressure (SBP) is -1.02, with 95.87% of sample points falling within the agreement range of [-18.09, 16.05]. For the diastolic BP (DBP),  $\mu$  is 0.23, with 95.47% of sample points within the agreement range of [-12.23, 12.70]. The mean differences for both SBP and DBP are close to zero, indicating that there is no significant systematic bias between the EMMRBP and the reference device, thus ensuring a great accuracy in practical use.

## 7. Related work

BP measurement has undergone significant developments over decades, with the “gold standard” remaining the direct measurement using invasive medical devices inserted into the arterial line of the subject [13]. While this method ensures a high accuracy, its application in routine monitoring is limited by the risks of pain and potential infection. To address these challenges, non-invasive BP measurement techniques have been introduced as safer and more convenient alternatives. These methods include auscultation [14], oscillometry [15], volume clamp techniques [16], sphygmomanometers [17], and ultrasounds [18]. Traditional contact-based methods, such as cuff-based sphygmomanometers and wearable devices employing photoplethysmography (PPG) [19] or electrocardiography (ECG), have been widely used in clinical and daily settings. However, these approaches have notable limitations, including user discomfort during inflation for cuff-based devices, a low user compliance, skin condition

---

constraints, and a susceptibility to motion artifacts for wearable devices [20, 21].

To overcome these limitations, non-contact BP measurement technologies have gained attention. These include camera-based systems utilizing pulse-induced motion from video data [22], ultrasound-based techniques that detect arterial wall dynamics [23], and wireless RF-based systems monitoring skin or blood vessel displacement [24]. Among these, mmWave radar stands out due to its ability to capture micro-level physiological movements, such as arterial pulsations and thoracic displacements [25]. Compared to other non-contact methods, mmWave radar provides unique advantages, including a high spatial resolution, a robustness to environmental conditions, and strong privacy protection, making it suitable for unobtrusive and continuous health monitoring [26]. Moreover, its high-frequency signal can penetrate clothing without obstruction, thus avoiding the illumination sensitivity issues associated with optical sensors. This technology excels in both static and dynamic scenarios, thereby enabling high-precision tracking of vital signs during motion and supporting multi-user scalability for the simultaneous monitoring of multiple vital signs [27].

The growing interest in mmWave radar technology highlights its potential as a cornerstone in the development of advanced, non-contact BP monitoring systems, thus offering unique solutions to the challenges faced by traditional and other non-invasive methods.

## 8. Conclusions

In this paper, we presented a novel edge-assisted framework for non-contact BP monitoring using mmWave radar, thus addressing the challenges of multi-user environments with limited computational resources and a large number of users submitting inference requests at the same time. By leveraging deep reinforcement learning, the proposed EMMRBP method dynamically adjusts configurations such as the sampling interval and input signal sequence length, thus achieving an optimal balance between accuracy and latency. A DRL-based approach was employed to address the problem by enabling the agent to learn optimal decision-making for the interval and sequence lengths through interactions with the environment. Finally, we implemented a testbed to evaluate the performance of our method. Extensive experimental results showed that our method outperforms the baselines, thus achieving a latency reduction of up to 70.3% and improving the reward by up to 29.7%, while maintaining accuracy loss within 5%. These findings highlight the potential of integrating edge computing and advanced AI techniques for efficient and scalable health monitoring applications. Future work will explore integrating additional physiological parameters and improving the system robustness in diverse real-world settings.

### Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

### Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants, No. 62232004, Jiangsu Provincial Frontier Technology Research and Development Program under Grant BF2024070, Shenzhen Science and Technology Program under Grant

KJZD20240903100814018, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grants No. 93K-9, and partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization and Collaborative Innovation Center of Wireless Communications Technology. We also thank the Big Data Computing Center of Southeast University for providing the experiment environment and computing facility.

### Conflict of interest

Fang Dong and Jun Shen are the editorial board members for *Applied Computing and Intelligence* and were not involved in the editorial review and the decision to publish this article.

### References

1. D. S. Picone, Accurate measurement of blood pressure, *Artery Res.*, **26** (2020), 130–136. <https://doi.org/10.2991/arteryres.k.200624.001>
2. G. Mancia, G. Parati, Ambulatory blood pressure monitoring and organ damage, *Hypertension*, **36** (2000), 894–900. <https://doi.org/10.1161/01.HYP.36.5.894>
3. N. Tomitani, S. Hoshide, K. Kario, The importance of regular home blood pressure monitoring over the life course, *Hypertens. Res.*, **47** (2024), 540–542. <https://doi.org/10.1038/s41440-023-01492-8>
4. K. Kario, Morning surge in blood pressure and cardiovascular risk: evidence and perspectives, *Hypertension*, **56** (2010), 765–773. <https://doi.org/10.1161/HYPERTENSIONAHA.110.157149>
5. S. Iyer, L. Zhao, M. P. Mohan, J. Jimeno, M. Y. Siyal, A. Alphones, et al., mm-wave radar-based vital signs monitoring and arrhythmia detection using machine learning, *Sensors*, **22** (2022), 3106. <https://doi.org/10.3390/s22093106>
6. M. Ebrahim, F. Heydari, T. Wu, K. Walker, K. Joe, J. Redoute, et al., Blood pressure estimation using on-body continuous wave radar and photoplethysmogram in various posture and exercise conditions, *Sci. Rep.*, **9** (2019), 16346. <https://doi.org/10.1038/s41598-019-52710-8>
7. Y. Liang, A. Zhou, X. Wen, W. Huang, P. Shi, L. Pu, et al., Airbp: monitor your blood pressure with millimeter-wave in the air, *ACM T. Internet Thing.*, **4** (2023), 28. <https://doi.org/10.1145/3614439>
8. Y. Ran, D. Zhang, J. Chen, Y. Hu, Y. Chen, Contactless blood pressure monitoring with mmwave radar, *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2022, 541–546. <https://doi.org/10.1109/GLOBECOM48099.2022.10001592>
9. U. Senturk, K. Polat, I. Yucedag, A non-invasive continuous cuffless blood pressure estimation using dynamic recurrent neural networks, *Appl. Acoust.*, **170** (2020), 107534. <https://doi.org/10.1016/j.apacoust.2020.107534>
10. Q. Hu, Q. Zhang, H. Lu, S. Wu, Y. Zhou, Q. Huang, et al., Contactless arterial blood pressure waveform monitoring with mmwave radar, *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, **8** (2024), 178. <https://doi.org/10.1145/3699781>

11. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, et al., Attention is all you need, *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, 6000–6010.
12. A. Zhao, E. Zhu, R. Lu, M. Lin, Y. Liu, G. Huang, Augmenting unsupervised reinforcement learning with self-reference, arXiv: 2311.09692. <https://doi.org/10.48550/arXiv.2311.09692>
13. Z. Jiang, S. Li, L. Wang, F. Yu, Y. Zeng, H. Li, et al., A comparison of invasive arterial blood pressure measurement with oscillometric non-invasive blood pressure measurement in patients with sepsis, *J. Anesth.*, **38** (2024), 222–231. <https://doi.org/10.1007/s00540-023-03304-2>
14. G. Van Montfrans, G. Van Der Hoeven, J. Karemaker, W. Wieling, A. Dunning, Accuracy of auscultatory blood pressure measurement with a long cuff, *Br. Med. J. (Clin. Res. Ed.)*, **295** (1987), 354–355. <https://doi.org/10.1136/bmj.295.6594.354>
15. M. Ramsey, Noninvasive automatic determination of mean arterial pressure, *Med. Biol. Eng. Comput.*, **17** (1979), 11–18. <https://doi.org/10.1007/BF02440948>
16. J. Penaz, Photoelectric measurement of blood pressure, volume and flow in the finger, *Proceedings of the 10th international conference on medical and biological engineering*, 1973, 104.
17. G. Pressman, P. Newgard, A transducer for the continuous external measurement of arterial blood pressure, *IEEE Transactions on Biomedical Electronics*, **10** (1963), 73–81. <https://doi.org/10.1109/TBME.1963.4322794>
18. I. Black, N. Kotrapu, H. Massie, Application of doppler ultrasound to blood pressure measurement in small infants, *J. Pediatr.*, **81** (1972), 932–935. [https://doi.org/10.1016/S0022-3476\(72\)80546-8](https://doi.org/10.1016/S0022-3476(72)80546-8)
19. Y. Cao, H. Chen, F. Li, Y. Wang, Crisp-bp: continuous wrist ppg-based blood pressure measurement, *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, 378–391. <https://doi.org/10.1145/3447993.3483241>
20. N. Pilz, D. S. Picone, A. Patzak, O. S. Opatz, T. Lindner, L. Fesseler, et al., Cuff-based blood pressure measurement: challenges and solutions, *Blood Pressure*, **33** (2024), 2402368. <https://doi.org/10.1080/08037051.2024.2402368>
21. Z. Shi, T. Gu, Y. Zhang, X. Zhang, mmbp: contact-free millimetre-wave radar based approach to blood pressure measurement, *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2023, 667–681. <https://doi.org/10.1145/3560905.3568506>
22. J. Zou, S. Zhou, B. Ge, X. Yang, Non-contact blood pressure measurement based on ippg, *Journal of New Media*, **3** (2021), 41–51. <https://doi.org/10.32604/jnm.2021.017764>
23. L. Xu, P. Wu, P. Xia, F. Geng, P. Wang, X. Chen, et al., Continuous and noninvasive measurement of arterial pulse pressure and pressure waveform using an image-free ultrasound system, arXiv: 2305.17896. <https://doi.org/10.48550/arXiv.2305.17896>
24. M. Alizadeh, G. Shaker, J. De Almeida, P. Morita, S. Safavi-Naeini, Remote monitoring of human vital signs using mm-wave fmcw radar, *IEEE Access*, **7** (2019), 54958–54968. <https://doi.org/10.1109/ACCESS.2019.2912956>

---

- 25. S. Churkin, L. Anishchenko, Millimeter-wave radar for vital signs monitoring, *Proceedings of IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS)*, 2015, 1–4. <https://doi.org/10.1109/COMCAS.2015.7360366>
- 26. Z. Ling, W. Zhou, Y. Ren, J. Wang, L. Guo, Non-contact heart rate monitoring based on millimeter wave radar, *IEEE Access*, **10** (2022), 74033–74044. <https://doi.org/10.1109/ACCESS.2022.3190355>
- 27. F. Shamsfakhr, D. Macii, L. Palopoli, M. Corrà, A. Ferrari, D. Fontanelli, A multi-target detection and position tracking algorithm based on mmwave-fmcw radar data, *Measurement*, **234** (2024), 114797. <https://doi.org/10.1016/j.measurement.2024.114797>



AIMS Press

© 2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)