Information Fusion 95 (2023) 426–445

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Multitask deep label distribution learning for blood pressure prediction

Keke Qin [a], Wu Huang [b,*], Tao Zhang [a]

[a] Chengdu Techman Software Co., Ltd, Chengdu, SC, 610000, China
[b] School of Computer Science, Sichuan University, Chengdu, SC, 610000, China

ARTICLE INFO

Keywords:
Blood pressure
Label distribution learning
Multitask learning
Information fusion
Photoplethysmogram (PPG)

ABSTRACT

Cuffless continuous blood pressure (BP) monitoring is of vital importance for personal health management. Currently, there are extensive studies devoted to cuffless BP prediction based on advanced machine learning techniques and by fusing a variety of physiological signals such as Photoplethysmogram (PPG) and Electrocardiogram (ECG) signals. However, the prediction accuracy still cannot meet the requirements, and it is inconvenient to collect multiple signals at the cost of additional sensors, which limits its potential application scenarios. Different from the conventional routine of modeling BP prediction as a classification or regression question, we model BP prediction as a label distribution learning question (*sample level information fusion*) for the first time and an end-to-end model is trained based on the proposed adaptive multitask weighted loss to predict systolic BP (SBP), diastolic BP (DBP) and mean BP (MBP) in parallel (*task level information fusion*), with only PPG signal as input. Resultly, not only precise BP but also predictive confidence interval are reported, and the *normalize target* technique usually used in regression modeling is no longer needed. To fully delve useful information for BP prediction from the only PPG signal, an end-to-end network is proposed for learning and fusing information from different modalities (original signal and its derivatives, time domain and time–frequency domain) of the signal (*feature fusion*). Besides, taking into account the varying informativeness of each learned feature accounting for different prediction tasks, task-specific attention module is introduced to learn the varied importance of each feature learned to different prediction tasks, under the hard parameter sharing mode of multitask learning (MTL) network. Extensive experiments on a publicly available database indicate that: (1) The proposed MTL model achieves superior performance over the corresponding single-task learning (STL) model at the cost of only about 1/3 times the amount of parameters. (2) The distribution learning mode enables superior generalization ability of the model over the regression modeling mode in both MTL and STL settings. (3) Compared with regression modeling, the distribution learning mode can alleviates the predictive bias of the trained model due to skewed distribution in dataset, given TFNet as feature learner. (4) The fusion of information of different modalities of PPG signal can significantly improve the generalization ability of the prediction model. (5) The proposed model has achieved superior performance over several representative methods/systems, while using only PPG signal and no any calibration procedure is required.

1. Introduction

Blood pressure (BP) plays an essential role in people's health, and continuous BP tracking provides a powerful reference to help physicians make decisions in the diagnosis of several diseases, especially cardiovascular disease. In commercial area, BP prediction module has become a standard configuration of health-related wearable devices and health care products [1], such as smart bracelet.

In fact, the research of BP measurement has a long history. From the mercury sphygmomanometers, the so-called gold standard, to the kirschner stethoscope method [2], the oscillometric-based [3], and the volume clamp method [4], these approaches belong to physical methods based on pressure conversion. The main drawback of these methods is that they are cuff-based and will cause discomfort when worn for long periods of time, which prevents it for continuous BP monitoring. Pulse transit time (PTT)-based method [5] could be used to real-time BP monitoring. Whereas, it is an ideal model and frequent calibration per subject is needed to ensure accuracy [6,7]. Therefore, it is not suitable for complex clinical situations well, as patients suffer from various symptoms such as bleeding and are affected by drugs [8].

Benefiting from the development of machine learning (ML) technology, many researchers have started to achieve continuous BP prediction in a data-driven manner, where ML algorithms are employed to learn

* Corresponding author.
E-mail addresses: keke.qin@tme.com.cn (K. Qin), huangwu@scu.edu.cn (W. Huang), zhangtao@tme.com.cn (T. Zhang).
URLs: http://www.tme.com.cn (K. Qin), http://scu.edu.cn (W. Huang), http://www.tme.com.cn (T. Zhang).

https://doi.org/10.1016/j.inffus.2023.02.019
Received 5 January 2022; Received in revised form 13 February 2023; Accepted 15 February 2023
Available online 17 February 2023

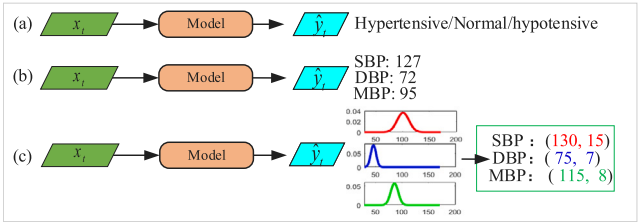


Fig. 1. Comparison of different learning modes for blood pressure prediction. (a) classification formulation; (b) regression modeling; (c) distribution learning.

the complex mapping between inputs and BP from vital signals. Existing ML-based approaches for BP prediction can be categorized into classification-based methods and regression-based methods, as illustrated in Fig. 1(a)~(b). In classification-based methods [9–11], the total BP range is divided into several disjoint intervals, each of which represents an independent category, and then the model is trained to correctly predict the belonging category given inputs. Apparently, this type of modeling method can only be used to roughly estimate the BP range and diagnose whether an individual's BP is normal or not. In addition, the order of BP interval corresponding to category is ignored, which is obviously unreasonable and may lead to very abnormal prediction results of the model. In regression-based methods [7,8,12–28], the model is trained by minimizing the distance between the predicted BP and the genuine BP to directly predict BP values. Note that almost all work in this area follows this paradigm. Treating different BP values as distinct classes may not yield the best performance because the similarity of samples between adjacent BP value is not considered. Furthermore, due to the large range of target value, *normalize target* [17,29] technique is often employed to reduce the range of target to boost training.

Converting hard target (i.e. the binary vector generated through one-hot encoding) to soft target in some form, such as label smoothing [30], have been proved to achieves gains in performance and generalization ability of predictive model in classification settings. Especially, in knowledge distillation [31], a smaller model is trained to match the output (soft target) of a complex model, instead of the ground-truth hard label, and the reason why knowledge distillation works is that a lot of helpful information that could not be encoded with a single hard label can be carried in soft label. Taking BP prediction into account, for a sample with fixed target BP, if the similarity information of the samples with BP near the target can be exploited for training, it is reasonable to achieve better performance. However, unlike the very limited number of categories in classification problems, individual's BP is a continuous real number within a certain range, and there are no explicit boundaries between different BP values.

Label distribution learning (LDL) [32], as a new learning paradigm, has been applied to several tasks such as multi-label classification [33] and age estimation [33–35]. In LDL, the label distribution itself can naturally reflect the ambiguous information of sample label with respect to label space. Note that LDL is suitable to both classification and regression tasks [33]. Instead of following the convention of modeling BP prediction as a classification or regression question, to leverage the information of samples near the target, based on a reasonable assumption—BP is assumed to be an integer, we firstly model BP prediction as a label distribution learning question. Specifically, the label distribution of each sample is firstly generated based on its ground-truth target value, and then a model is trained to predict the label distribution, instead of the isolated BP value.

Furtherly, based on the feature statistics (mean and variance) of the predicted label distribution, a new multitask loss function consisting of multiple loss terms for distribution learning is proposed, which models the distribution loss, the mean loss and variance loss of the distribution, the inconsistency loss between the predicted distributions of adjacent samples, and the model complexity term, respectively. More importantly, the loss scale of different tasks is modeled in the loss function, which has not been considered in existing related work [8,16–18,22]. Actually, the prediction difficulties of SBP, DBP and MBP present a big difference, which has been confirmed by almost all existing relevant studies [7,8,15–20,22–24,28,36–42]. Specifically, SBP is more difficult to predict than DBP, and MBP is between SBP and DBP. The difference will be more obvious with more diversified records included in the dataset. This difference is called loss scale and is modeled as uncertainty in the designed loss function.

In the present study, we also employ end-to-end deep learning for our purpose—the model is trained by using only raw PPG signal as input to predict label distribution. Therefore, both the automatic feature learning and the predictive model building processes exploit the label distribution information. As for the design of the neural network model, to fully delve and exploit the implied information useful for BP prediction from the easily acquired PPG signal, we consider from the following two aspects: (*i*) *original signal and its derivatives*: the motivation is that derivatives of the original signal contain more detailed information characterizing pulse information, which is useful in analyzing PPG signal [43]. Specifically, the 1st order derivative of PPG signal, i.e velocity PPG, contains slope information related to BP, and the 2nd order derivative of PPG signal, i.e accelerated PPG, contains dominating information about the dichroic notch and the diastolic point [44]; (*ii*) *time domain information and time frequency domain information*: signal in the time domain captures the waveform contour changes over time, while time–frequency information reveals the frequency changes over time. Time–frequency analysis is advantages in analyzing non-stationary signal accompanied with abnormal pattern. In fact, learning in the frequency domain by converting time domain signal through the powerful time–frequency analysis techniques has shown its superiority over time-spatial domain in computer vision area [45] and IoT applications [46] with various sensor inputs.

The feature learner is designed by learning and fusing information from different modalities of PPG signal as above. Furthermore, previous studies based on explicit feature extraction have shown that the contribution of each feature varies to different prediction tasks (i.e SBP, DBP and MBP) [14,20,47–49]. To settle this issue under the hard parameter sharing mode for MTL, we introduced an attention module in each task network to learn the varied importance of each feature learned for different prediction tasks.

Based on the above considerations, we proposed a novel time domain and frequency domain network based multitask deep distribution learning framework (TFNet-MTD^{2}L) for BP prediction. From the perspective of information fusion, the proposed method considers three levels of fusion: (*i*) *sample level*: based on the novel distribution learning paradigm, the information of adjacent samples with BP nearing a sample is utilized through soft target during training; (*ii*) *task level*: Unlike traditional single-task learning, here, different tasks are correlated by a shared feature learner, and the multitask network is jointly trained to make predictions; (*iii*) *feature level*: informative feature related to BP is delved by learning and fusing information from different modalities of PPG signal.

Summarily, the main contributions of this study include:

- Modeling BP prediction as a label distribution learning question for the first time, which is different from the traditional routine of formalizing BP prediction as a regression or classification question. Resultly, not only precise BP value but also confidence interval is reported, and the *normalize target* technique as used in regression-based methods is no longer needed.
- A new multitask loss function for distribution learning is proposed for join-training of the model by utilizing the statistical information of the distribution and modeling the loss scale of different prediction tasks.

- An end-to-end network is proposed for learning and fusing information from different modalities (original signal and its derivatives, time domain and time–frequency domain) of PPG signal. Besides, task-specific attention module is introduced to learn the importance of each learned feature for different prediction tasks.
- Extensive ablation experiments established the effectiveness of the proposed method—TFNet-MTD^{2}L. Through ablation experiments, we showed a path to reduce the predicted MAE of SBP, DBP, and MBP to 5.8815, 3.3549, and 3.4859 mmHg, respectively.

The rest of the paper is organized as follows: Related work is firstly reviewed in Section 2. Then, the concept of multitask label distribution learning for blood pressure prediction is firstly formulated in Section 3. Next, Section 4 presents the proposed network architecture. The adaptive multitask weighted loss for distribution learning is proposed in Section 5. Experimental results are reported and analyzed in Section 6. Finally, we conclude the paper in Section 7.

2. Related work review

2.1. Traditional ML for BP prediction

Traditional ML-based studies in this area mainly focus on feature exploration and signal fusion [6,7,12,15,26,38,47]. Moreno et al. [15] firstly analyzed the features comprehensively from PPG signal that required for building BP prediction model. Kachuee et al. [7] systematically investigated two types of features required for BP prediction—physiological parameters, and whole-based features. Fujita et al. [27] proposed level-crossing features (LCFs) based on the derivatives of PPG signal for BP prediction. Thambiraj et al. [20,41] proposed a new feature—Womersley number for BP prediction. Xing et al. [40] extracted frequency domain features from PPG signal based on FFT for beat-to-beat BP prediction. Bose S et al. [28] firstly model the explicitly feature extraction as dictionary learning and the sparse representation of the PPG signal is used as feature for further beat-to-beat BP prediction. Studies [50,51] investigated the use of ECG only to measure BP. Studies [24,36,52,53] investigated BP prediction by using oscillometric waveform. Fong et al. [23] proposed a SVR-based ensemble model for BP prediction based on multi-channel PPG signal. Miao et al. [14] proposed a multi-instance regression model for BP prediction by fusing one ECG signal and two pulse pressure wave signals. Firstly, although the fusion of multiple signals can effectively improve the prediction accuracy, it is inconvenient and added additional burden in practice, which limits its potential usage scope. In contrast, we attempt to achieve BP prediction using the easily available PPG signal only. Secondly, tedious feature engineering is needed in the above work, and feature selection has to be performed per task before training model for SBP, DBP and MBP prediction, respectively. Besides, the most informative features selected may vary from individual to individual [47]. In contrast, in our end-to-end deep learning based multitask joint-training framework, both feature engineering and feature selection are not required. In fact, we introduced task-specific attention module to learn the task-dependent weights for each learned feature automatically, and only one model with multiple outputs is trained to predict SBP, DBP and MBP in parallel.

2.2. Label distribution learning

As a new ML paradigm, label distribution learning (LDL) [32] has attracted wide attention. It naturally provides a way to express label ambiguity. LDL has been successfully applied to general classification task [54] and computer vision tasks such as age estimation [33–35] and expression recognition [55]. As far as we know, to date, BP prediction is considered mainly as a regression question in existing studies. Here, we model BP prediction as a multitask label distribution learning question for the first time. Specifically, the label space consists of discrete blood pressure values within the possible BP range. Unlike the regression paradigm where the model is trained directly to map the input to its corresponding BP value, in LDL, the model is trained to map the input to a distribution in the label space. Then, BP is computed as the mean of the distribution with respect to the label space. This paradigm actually exploits the information of the samples whose labels are close to a target implicitly through the soft target during training.

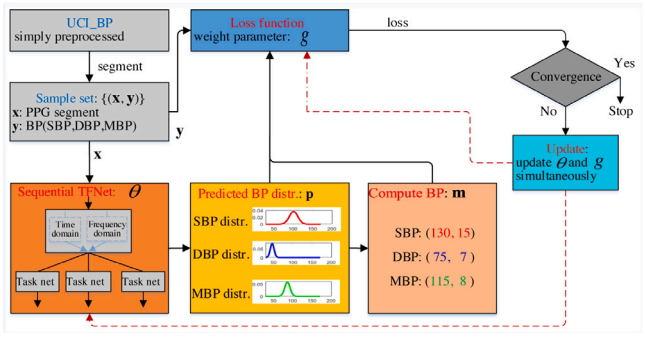


Fig. 2. Schematic diagram of the proposed multitask deep distribution learning framework—TFNet-MTD^{2}L for blood pressure prediction.

2.3. MTL based BP prediction

Currently, studies [8,17,18,22,25] related to multitask learning (MTL) in this area are all in neuro models owning to the modularity architecture of neural network. Su et al. [17] proposed a model named DeepRNN by stacking multiple LSTM layers for long-term BP prediction. Similarly, Tanveer et al. [8] proposed a model consisting of ANN and multiple LSTM layers to predicting BP using raw PPG signal. Baek et al. [18] proposed a fully convolutional network by stacking the proposed Extraction-Concentration block for BP prediction. Slapnicar et al. [25] proposed a spectro-temporal deep neural network for BP prediction. Note that the residual connection technique is utilized in studies [17,18,25]. Eom et al. [22] proposed a model comprised of a VGG-style module, Bi-GRU and attention module for BP prediction. In the context of [56], all of the above MTL models follow the hard parameter sharing mode [56], i.e. several layers for learning informative representations are shared among all tasks, which is followed by multiple independent task networks. Compared with single-task learning (STL), in MTL, only one model is trained and multiple tasks are predicted in parallel given input. More importantly, the generalization ability of the model may be improved if multiple related tasks share knowledge in some way appropriately.

One problem with the above work is that the total loss used for back propagation is simply the average of respective task losses, that is, the loss scale of each task is equal by default. Whereas, previous research has clearly shown that the prediction accuracy between different tasks (SBP, DBP and MBP) exhibits a significant difference, which is called task-dependent or homoscedastic uncertainty in Bayesian modeling [57]. Meanwhile, balancing the contribution of different tasks through adaptive loss weighting has shown superiority in several computer vision application scenarios [58–60]. Here, we model the loss scale of different tasks as task-dependent uncertainty and proposed a new distribution learning-based loss function. Moreover, in the above work, there is no explicit mechanism to handle the difference between the most informative feature sets related to different prediction tasks, which may affect the joint-training of multiple tasks. We settle this question by introducing an attention module in each task network to learn the task-dependent weights of each feature.

3. Problem formulation

In this section, we first provide a formal description of the proposed multitask deep distribution learning framework for BP prediction. Fig. 2 presents a schematic diagram of the training process of the proposed multitask distribution learning framework. Assume the possible BP range is $[pl, ph], pl, ph \in Z^+, pl \ll ph$. Then, discretize the possible BP range into a complete and ordered label space with a step size of 1 (note that a resolution of 1 is large enough for BP), i.e. $l = \{pl, pl+1, \dots, ph\}$. The size of the label space is denoted as C, $C = ph-pl+1$. Let K denotes the number of tasks, all of the K tasks share the same input. Here, K is set to 3, representing the three prediction tasks of SBP, DBP and MBP, respectively. Let fs denotes the sampling frequency, T denotes the segmentation time interval (unit: second). Suppose we are given a sample set—$\{(x_1, y_1), \dots, (x_i, y_i), \dots\}$, $x_i \in \mathbf{R}^L$, representing the input of the ith sample composed of signal sequence with length—L, $L = T \cdot fs$. $y_i = (y_i^s, y_i^d, y_i^m)$, representing the output of the ith sample composed of SBP, DBP and MBP, respectively. Here, we consider sequential deep learning where input is a sequence composed of s samples. In other words, the input shape is $(N, s \times L, 1)$, where N denotes the batch size, s denotes the length of the sequence.

Time–frequency network (TFNet), which extracts information from both time domain and time–frequency domain, is utilized to learn informative intermediate features from raw PPG signal, which is followed by three independent networks to learn the mapping between features and label space for each task. Suppose $f_{\theta^c}(x)$ represents the mapping from raw signal sequence to intermediate features, which is defined by parameters θ^c, $f_{\theta^c}(x) \in \mathbf{R}^{(N \cdot s) \times n_{total}}$, where n_{total} denotes the dimension of the output of the final Concatenation layer. Denotes z^s, z^d and z^m the output logits of the three task networks. Then,

$$z^s = f_{\theta^c}(x)\theta^{s^T}, z^d = f_{\theta^c}(x)\theta^{d^T}, z^m = f_{\theta^c}(x)\theta^{m^T}, \tag{1}$$

where θ^s, θ^d, $\theta^m \in \mathbf{R}^{C \times n_{total}}$ are the task-specific parameters of SBP, DBP and MBP prediction task, respectively. θ^c is shared parameters for the three tasks. For convenience, let $\theta := (\theta^c, \theta^s, \theta^d, \theta^m)$. Suppose p^s, p^d and p^m the predicted distribution for SBP, DBP and MBP prediction, respectively. $p^s, p^d, p^m \in \mathbf{R}^{(N \cdot s) \times C}$. $p_{i,j}^s = (p_{i,j,1}^s, p_{i,j,2}^s, \dots, p_{i,j,C}^s)$, $p_{i,j}^d = (p_{i,j,1}^d, p_{i,j,2}^d, \dots, p_{i,j,C}^d)$, $p_{i,j}^m = (p_{i,j,1}^m, p_{i,j,2}^m, \dots, p_{i,j,C}^m)$, $i \in \{1, 2..., N\}$, $j \in \{1, 2, \dots, s\}$. For convenience, let $p := (p^s, p^d, p^m)$. s denotes the softmax activation function, $s(x)_i = \exp(x_i) / \sum_k \exp(x_k)$. Then,

$$p_{i,j,c}^s = s(z_{i,j}^s)_c, p_{i,j,c}^d = s(z_{i,j}^d)_c, p_{i,j,c}^m = s(z_{i,j}^m)_c, \tag{2}$$

where $p_{i,j,c}^t$ denotes the degree that the cth label describes the jth sample of the ith sequence for the tth task. Obviously, $\sum_c p_{i,j,c}^t = 1$ holds for $t \in \{1, 2, \dots, K\}$, $i \in \{1, 2, \dots, N\}$, $j \in \{1, 2, \dots, s\}$. Then, the mean and variance of the distribution of SBP, DBP and MBP can be computed respectively as follows,

$$\begin{cases} m_{i,j}^s = \sum_{c=1}^{C} p_{i,j,c}^s \cdot (pl + c - 1), \\ v_{i,j}^s = \sum_{c=1}^{C} p_{i,j,c}^s \cdot (pl + c - 1 - m_{i,j}^s)^2, \\ m_{i,j}^d = \sum_{c=1}^{C} p_{i,j,c}^d \cdot (pl + c - 1), \\ v_{i,j}^d = \sum_{c=1}^{C} p_{i,j,c}^d \cdot (pl + c - 1 - m_{i,j}^d)^2, \\ m_{i,j}^m = \sum_{c=1}^{C} p_{i,j,c}^m \cdot (pl + c - 1), \\ v_{i,j}^m = \sum_{c=1}^{C} p_{i,j,c}^m \cdot (pl + c - 1 - m_{i,j}^m)^2. \end{cases} \tag{3}$$

In the training phase, as Fig. 2 illustrates, the network accepts x as input and performs forward propagation, and then the task-dependent multitask weighted loss $L(.,.,.;\theta, g)$ is computed based on the predicted m, p and the ground-truth y. Next, the computed loss $L(m, p, y; \theta, g)$ is back propagated to update the network parameters θ and the task weights g simultaneously. The above process iterates until convergence. In the prediction phase, the input signal (e.g. PPG signal) is feed into the predictive model. Through forward propagation, the predicted SBP, DBP and MBP are then computed according to Eq. (3) as m^s, m^d and m^m, respectively.

4. Model architecture

We try to design such a network by considering the raw signal and its derivatives, time-domain and time–frequency-domain information of the signal, so as to extract useful information for BP prediction by learning and fusing different signal modalities from the easily obtained signal (such as PPG signal) as much as possible. As Fig. 3 illustrates, the proposed network—TFNet-MTD^{2}L, suitable for processing one-dimensional signal, follows the hard parameter sharing mode—the shared network tries to learn useful information, which is followed by several task networks, with each corresponds to a prediction task. The shared network contains two branches to extract information from time domain and time–frequency domain, respectively. Each branch contains three streams corresponding to the original signal and its first and second order derivatives.

4.1. Three-stream U^{2}Net

U^{2}Net [61] is a two-level nested U-Net structure used for image segmentation and object detection. The time domain network is designed based on U^{2}Net architecture. For each stream, the network captures waveform of signal with different derivatives. As Fig. 3 (middle) illustrates, U^{2}Net follows the Encoder–Decoder architecture and the basic component of which is the residual U-block (RSU). RSU is composed of three components: convolution layer (F) for extracting local feature, U-like structure (U_l) for extracting multi-scale feature and residual connection [62,63] for fusing local feature and multi-scale feature. It is formally defined as follows given input $X \in \mathbf{R}^{N \times M \times C_{in}}$,

$$\mathrm{RSU}_l(X) = \mathrm{F}(X) + \mathrm{U}_l(\mathrm{F}(X)), \tag{4}$$

where the output $\mathrm{RSU}_l(X) \in \mathbf{R}^{N \times M \times C_{out}}$, l indicates the depth of the U-shaped structure. C_{in} and C_{out} denote the input channel and output channel, respectively. Additionally, considering the various transitions behavior of signal along time, instead of identity mapping, the multi-scale extraction (MSE) module is developed for extracting multi-scale patterns that is used in conjunction with the decoder output of the last stage for reconstructing the high resolution signal. The MSE module is composed of four parallel dilated-convolution (DConv) [64] operations with different dilation rates for extracting various multi-scale patterns between different neighboring pixels. The outputs of these convolutions are concatenated together, and then the channel of which is reduced to original dimension through several conventional convolution layers. It is formally defined as follows given input $X \in \mathbf{R}^{N \times M \times C_{in}}$,

$$\begin{aligned} X^i &= \mathrm{DConv}_i(X), i \in \{1, 2, 3, 4\}, \\ \mathrm{MSE}(X) &= \mathrm{Conv}^{(2)}(Concat(X^1, X^2, X^3, X^4)), \end{aligned} \tag{5}$$

where $X^i \in \mathbf{R}^{N \times M \times C_{out}}$, $\mathrm{MSE}(x) \in \mathbf{R}^{N \times M \times C_{out}}$, C_{out} is the output channel. $\mathrm{Conv}^{(2)}$ denotes two convolution layers in series.

Finally, instead of using just the output of the last decoder stage, the multi-level outputs (MLO) at different decoder stages of U^{2}Net are concatenated to form the final output—$X_t^{o'}$, $X_t^{o'} \in \mathbf{R}^{N \times (s \cdot L) \times C_{out}}$. Because our goal is to respond BP to each sample in a signal sequence consisting of s consecutive samples, and the duration of each sample is T seconds. Therefore, $X_t^{o'}$ has to be reshaped to $X_t^o \in \mathbf{R}^{N \times s \times C_{out}}$ through Sequence-to-Segment conversion (S2SC) operation. S2SC is comprised of a Reshape operation that reshape $X_t^{o^{i'}}$ to the shape of $\mathbf{R}^{N \times s \times L \times C_{out}}$, and is followed by global average pooling (GAP [65]) to further squeeze it into the shape of $\mathbf{R}^{N \times s \times C_{out}}$.

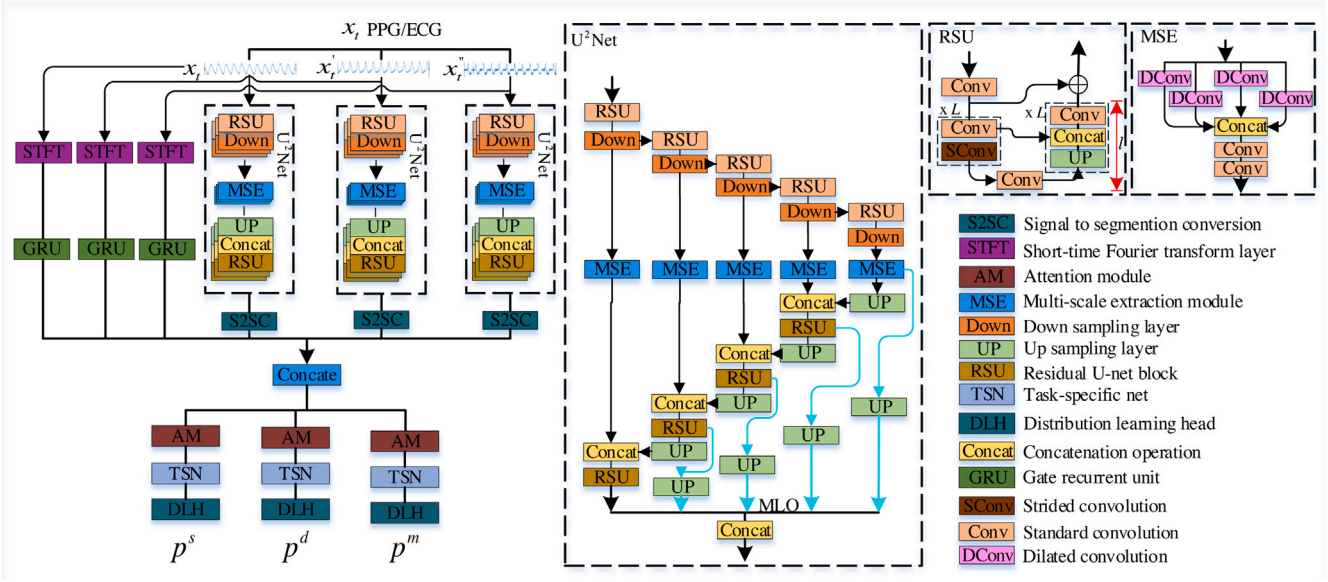


Fig. 3. The network architecture of the proposed TFNet-MTD^{2}L for BP prediction. **Left**: the TFNet-MTD^{2}L framework is composed of a shard network that extract informative features from time-domain and time–frequency domain of the input signal and its derivatives. x_t, x_t' and x_t'' denote the input signal and its corresponding 1st and 2nd order differential signals, respectively; **Middle**: the U^{2}Net-based module with multi-level outputs (MLO); **Right**: the RSU module and the MSE module.

4.2. Three-stream recurrent network

For each stream in the time–frequency domain, the signal is firstly transformed into time–frequency domain by short-time Fourier transform (STFT). Suppose sliding window $W_\tau(t)$ with length τ only have non-zero value when $t \in [1, \tau]$. Given real-valued input signal $X \in \mathbf{R}^{N\times(s\cdot L)\times 1}$, STFT is formally defined as follows,

$$\mathrm{STFT}^{\tau,\upsilon}(X)_{[n,m,k]} = \sum_{i=1}^{s\cdot L} X_{[n,i,1]} W_\tau(i-\upsilon m)\exp\{-j\frac{2\pi k}{\tau}(i-\upsilon m)\}, \tag{6}$$

where $\mathrm{STFT}^{\tau,\upsilon}(X) \in \mathbf{R}^{N\times M\times K}$ denotes STFT with window width τ and sliding step υ. M denotes the number of time chunks, $M = s\cdot L/\tau$. K denotes the number of frequency components, $K = \lfloor\tau\rfloor+1$. In experiment, $\tau = \upsilon = L$, M is therein equals s. Then GRU is used to learn useful, time-varying frequency domain information from the time–frequency spectrum. Formally,

$$X_f^o = \mathrm{GRU}(\mathrm{STFT}^{\tau,\upsilon}(X)), \tag{7}$$

where $X_f^o \in \mathbf{R}^{N\times s\times n_{unit}}$, n_{unit} denotes the dimensionality of the output space in GRU.

4.3. Task network

Each task network is composed of attention module (AM), task-specific network (TSN) and distribution learning head (DLH). Attention module is incorporated to distinguish different importance between different channels in time domain and different positions in time–frequency domain. Suppose the input to the attention module is $X_{tf} \in \mathbf{R}^{N\times s\times 3(C_{out}+n_{unit})}$, $X_{tf} = Concat(X_t^o, X_t^{o'}, X_t^{o''}, X_f^o, X_f^{o'}, X_f^{o''})$. For convenience, let $n_{total} := 3(C_{out} + n_{unit})$. Then, the attention weights are computed as follows [66]: (1) compress X_{tf} to $X'_{tf} \in \mathbf{R}^{N\times 1\times n_{total}}$ through GAP operation, i.e. $X'_{tf} = \mathrm{GAP}(X_{tf})$; (2) compute attention weights $W(X_{tf}) \in \mathbf{R}^{N\times 1\times n_{total}}$ by a single hidden bottleneck layer multi-layer perceptron (MLP), which is formally defined as,

$$W(X_{tf}) = \sigma(\mathrm{FC}_2(\delta(\mathrm{FC}_1(X'_{tf})))), \tag{8}$$

where δ denotes ReLU activation function, σ denotes sigmoid activation function. Last, the final output of the attention module is computed as $X_{tf} = X_{tf} \odot W(X_{tf})$. Note that AM is attached in each task network, in other words, it is task-specific.

TSN is comprised of a fully-connected layer, and DLH is a fully-connected layer with softmax activation and output dimension equals C, which is specialized for distribution learning.

5. Adaptive multitask loss weighting

In addition to the predicted distribution—p, the statistics of p (such as mean and variance) are also used to design auxiliary loss terms. Specifically, the proposed loss function is composed of five loss terms: (1) *softmax loss*; (2) *mean loss*; (3) *variance loss*; (4) *inconsistency loss*; (5) *regularization term*. The *softmax loss* penalizes the difference between the predicted and the genuine distributions. The *mean loss* penalizes the difference between the expectations of the predicted and the genuine label distributions. The *variance loss* penalizes the dispersion of the predicted label distribution. The *inconsistency loss* penalizes the inconsistency of the predicted label distributions of adjacent samples in a sequence. The *regularization term* controls the model complexity.

Before introducing the loss function, we review a work firstly, on which the modeling of loss scale of different tasks in the proposed loss is based. Kendall et al. [60] proposed a general loss function that unified classification task and regression task in multitask settings, where the losses of different tasks are weighted based on task-dependent uncertainty modeling with the form of $\sum_j \log\delta_j + L_j/{\delta_j}^2$, where L_j denotes the loss of the jth task, $1/{\delta_j}^2$ serve as a scaling factor to reduce the contribution of task-j with high uncertainty to the total loss, $\log\delta_j$ represents regularization term actually, which makes the weight of loss (i.e. δ_j) learnable. The thought of determining loss scale based on uncertainty modeling is utilized in the *mean loss* and *variance loss* terms to balance the loss weights of different tasks. Next, we first introduce each loss term separately, and then give a formal definition of the final loss function.

5.1. Softmax loss

For simplicity, we omit the sequence dimension index of p. The distribution likelihood of the ith sample on the jth task: $\prod_{c=1}^{C} {p_{i,c}^j}^{\tilde{y}_{i,c}^j}$,

$\tilde{y}_i^j$ is the distribution of y_i^j in the label space. The total likelihood can be formulated as,

$$\prod_{i=1}^{N}\prod_{j=1}^{K}\prod_{c=1}^{C} p_{i,c}^{j\ \tilde{y}_{i,c}^{j}}.$$

Then, the distribution loss is the negative log-likelihood as follows,

$$L_0 = \frac{1}{N\cdot K}\sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{c=1}^{C} -\tilde{y}_{i,c}^{j}\log p_{i,c}^{j}, \tag{9}$$

where $L_0^j = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} -\tilde{y}_{i,c}^{j}\log p_{i,c}^{j}$ is actually the softmax loss regard to the jth task.

5.2. Mean loss

Suppose $m^j - y^j \sim \mathcal{N}(0,{\delta_j}^2)$, the variance—${\delta_j}^2$ is the task-related parameter, then the total likelihood can be formulated as, $\prod_{i=1}^{N}\prod_{j=1}^{K}\mathcal{N}(m_i^j - y_i^j;0,{\delta_j}^2)$. Then, the negative log-likelihood loss regard to the mean of the distribution as follows,

$$\begin{aligned} L_1 &= \frac{-1}{N\cdot K}\log\{\prod_{i=1}^{N}\prod_{j=1}^{K}\mathcal{N}(m_i^j - y_i^j;0,{\delta_j}^2)\} \\ &\propto \frac{1}{N\cdot K}\sum_{i=1}^{N}\sum_{j=1}^{K}\{1/2\cdot\log{\delta_j}^2 + (m_i^j - y_i^j)^2/(2{\delta_j}^2)\} \\ &= \frac{1}{K}\sum_{j=1}^{K} L_1^{j*}, \end{aligned} \tag{10}$$

where $L_1^{j*} = 1/2\cdot\log{\delta_j}^2 + \frac{1}{{\delta_j}^2}L_1^j$ is the L_1 loss regard to the jth prediction task, $L_1^j = \frac{1}{2N}\sum_{i=1}^{N}(m_i^j - y_i^j)^2$.

5.3. Variance loss

Suppose $\sqrt{v^j} \sim \mathcal{HN}({\delta_j}^2)$, $\mathcal{HN}$ denotes half-normal distribution parameterized by—${\delta_j}^2$, representing the task-related parameter. Then, the total likelihood can be formulated as, $\prod_{i=1}^{N}\prod_{j=1}^{K}\mathcal{HN}(\sqrt{v^j};{\delta_j}^2)$. Then, the negative log-likelihood loss regard to the variance of the distribution is formulated as:

$$\begin{aligned} L_2 &= \frac{-1}{N\cdot K}\log\{\prod_{i=1}^{N}\prod_{j=1}^{K}\mathcal{HN}(\sqrt{v^j};{\delta_j}^2)\} \\ &\propto \frac{1}{N\cdot K}\sum_{i=1}^{N}\sum_{j=1}^{K}\{\frac{1}{2}\log{\delta_j}^2 + v_i^j/(2{\delta_j}^2)\} \\ &= \frac{1}{K}\sum_{j=1}^{K} L_2^{j*}, \end{aligned} \tag{11}$$

where $L_2^{j*} = \frac{1}{2}\log{\delta_j}^2 + \frac{1}{{\delta_j}^2}L_2^j$ is the L_2 loss regard to jth prediction task. $L_2^j = \frac{1}{2N}\sum_{i=1}^{N} v_i^j$.

5.4. Inconsistency loss

For the ith sequence of samples, similar to [67], we impose a clustering constraint to its predicted distributions—$\{p_{i,j}^k\}_{j=1}^{s}$. The mean of the predicted distributions of the ith sequence for the kth task is $\bar{p}_i^k = \frac{1}{s}\sum_{j=1}^{s} p_{i,j}^k$. Then, based on the L_2 distance, the total *inconsistency loss* can be formulated as:

$$L_3 = \frac{1}{N\cdot K\cdot s}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{s}\|p_{i,j}^k - \bar{p}_i^k\|. \tag{12}$$

Obviously, L_3 ensures the intra-sequence prediction consistency for each task.

The final loss is formulated as, $L = L_0 + \mu\cdot L_1 + \tau\cdot L_2 + \rho\cdot L_3 + \lambda\cdot L_4$. The last regularization term—L_4 is L_2 norm controlling the model complexity, i.e. $L_4 = \|\theta\|^2$. Hyper parameters—μ, τ, ρ and λ are used to balance the weights of different loss terms. Replace L_0, L_1, L_2 and L_3 with Eqs. 5.1–(12), respectively, and L is then reformulated as,

$$\begin{aligned} L = \frac{1}{K}\sum_{j=1}^{K}\{&\frac{1}{2}\log({\delta_j^1}^{\mu}{\delta_j^2}^{\tau})^2 \\ &+ \frac{1}{2N}\sum_{i=1}^{N}(2\sum_{c=1}^{C}\hat{y}_{l,c}^j\log(1/s(z_{i,c}^j)) + \frac{\mu(m_i^j - y_i^j)^2}{{\delta_j^1}^2} + \frac{\tau\cdot v_i^j}{{\delta_j^2}^2}) \\ &+ \frac{\rho}{N\cdot s}\sum_{i=1}^{N}\sum_{j=1}^{s}\|p_{i,j}^k - \bar{p}_i^k\|\} + \lambda\|\theta\|^2, \end{aligned} \tag{13}$$

where ${\delta_j^l}^2$ denotes the variance of the jth prediction task regard to the lth loss term. θ denotes the trainable parameters of the model.

In practice, to avoid possibly meaningless negative value and the danger division by zero, without predicting ${\delta_j^l}^2$ directly, instead, we predict $g_j^l = \log{\delta_j^l}^2$ as in [57,68]. In addition, logarithmic operation is more numerical stable. Subsequent g_j^l into Eq. (13), and then L can be reformulated as:

$$\begin{aligned} L = \frac{1}{K}\sum_{j=1}^{K}\{&\frac{\mu\cdot g_j^1 + \tau\cdot g_j^2}{2} \\ &+ \frac{1}{2N}\sum_{i=1}^{N}(2\sum_{c=1}^{C}\hat{y}_{l,c}^j\log\frac{1}{s(z_{ic}^j)} + \mu\cdot e^{-g_j^1}(m_i^j - y_i^j)^2 + \tau\cdot e^{-g_j^2}\cdot v_i^j) \\ &+ \frac{\rho}{N\cdot s}\sum_{i=1}^{N}\sum_{j=1}^{s}\|p_{i,j}^k - \bar{p}_i^k\|\} + \lambda\|\theta\|^2. \end{aligned} \tag{14}$$

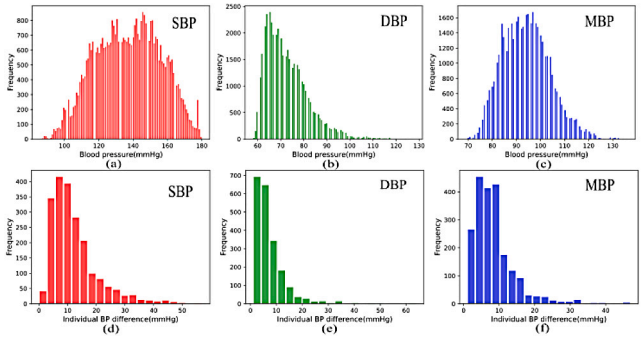


Fig. 4. Blood pressure distribution and individual dynamics of the final processed dataset. (a)~(c) denote the distributions of SBP, DBP and MBP, respectively; (d)~(f) denote the individual dynamics of SBP, DBP and MBP, respectively.

6. Data and experiments

6.1. Data preparation

The Cuff-Less Blood Pressure Estimation Data Set from UCL (https://archive.ics.uci.edu/ml/datasets/Cuff-Less+Blood+Pressure+Estimation) was used for experiments. This version of the dataset is derived from the MIMIC-II database [69] and has been widely used in the relevant studies [18–20,38] due to its ease of use. The data is stored hierarchically in four *.mat* format files containing a total of 12 000 records, each with a duration between 8 and 592 s. Each record is divided into disjoint segments, each of which has a duration of 10 s. For PPG signals, each PPG segment was processed with FFT to remove baseline drift and then filtered with a Butterworth band-pass filter with cutoff frequencies of 0.5 and 8 Hz. Very few extreme outliers in the segments were scaled based on statistical anomaly detection. Referring to studies [18,40], low quality fragments were discarded based on BP range criteria and peak analysis. The final processed dataset contains 2076 subject records, each containing 20 valid samples, for a total of

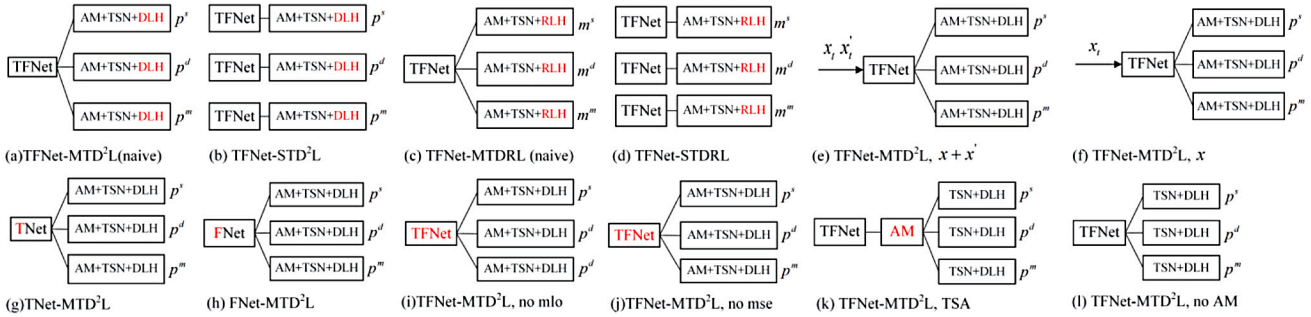


Fig. 5. Graphical illustration of the variants of the proposed TFNet-MTD²L. Abbr., mlo: multi-level output, mse: multi-scale extraction, TSA: task-shared attention, AM: attention module, DLH: distribution learning head, RLH: regression learning head.

41 520 samples. Each sample includes a PPG signal segment with 1250 sampling point and the corresponding BP values (include SBP, DBP and MBP), where MBP is computed as MBP=(2 ·DBP+SBP)/3. Training, validation and test sets are determined by dividing the data set in the ratio of 6:2:2 based on the unique record ID, and therefore all samples of each record appears only in training set, validation set or test set. Fig. 4 illustrates the BP distribution and individual BP dynamics of the dataset. The summary statistics of BP are SBP with 137.25±18.88 mmHg, DBP with 72.62.89±9.18 mmHg, MBP with 94.16±9.75 mmHg. The value of SBP covers an extensively large range. Individual BP dynamics are with mean of 12.17±8.17 mmHg, 7.46±6.08 mmHg, 8.37±5.73 mmHg for SBP, DBP and MBP, respectively. In addition, the MIMIC III database [70] is further used to validate the proposed method/model, which is described in Section 6.12.

6.2. Experiment settings

All network parameters are initialized with truncated normal distribution with *mean* equals 0 and *std* equals 0.05, and the model is optimized using Adam optimizer with mini-batch of 16. In each batch training, gradient—g is clipped to $gv/\|g\|_2$ if $\|g\|_2 > v$, v is set to 5 in experiment. Learning rate is initially set to 0.001 and is exponentially decreased by 0.05 after each epoch, and the maximum number of epochs is set to 80. Finally, the model with the lowest loss on validation set is selected as the final trained model for further test. To have a further exploration of the proposed method TFNet-MTD²L, we additionally implemented thirteen variants of the proposed method. A simple graphical illustration of these methods is shown in Fig. 5, and a detailed description of these methods is then given in Appendix. We declare that no any other post-processing procedure was used in the evaluation of the proposed method and its variants. All experiments were performed on a Windows platform equipped with an RTX 2080Ti GPU. The code is implemented based on Tensorflow framework with version of 2.1. Note that for each method, the experiment was repeated ten times by generating ten different splits of training, validation and test sets to get more reliable results, and the results given in each table are the average of the results of ten runs.

The experiments include: we firstly illustrates the training behavior and resulting prediction with confidence interval in Section 6.5. Next, in Section 6.6, we systematically compare different learning modes (STL vs. MTL, and distribution learning vs. regression modeling). In Section 6.7, we explore the network designing from the following three aspects: *time-frequency module*, *attention module* and *input combination*. In Section 6.8, we try to understand the role of adaptive weighted loss mechanism, which is followed by hyper parameters sensitivity analysis & verification of loss term in Section 6.9. Section 6.10 includes the evaluation of the proposed model according to several standards, and the statistical analysis of model fitting effect. The comparison with several representative methods/systems is presented in Section 6.11. Finally, Section 6.12 presents the external validation of the proposed method/model on the MIMIC III database.

6.3. Generation of ground-truth label distribution

For BP prediction, we assume the target distribution should concatenate around the ground-truth target y (SBP, DBP or MBP). Therefore, the ground-truth target distribution $\tilde{y}$ is generated based on a normal distribution. Concretely, $\tilde{y}_c = p(l_c|y,\sigma)/\sum_{k=1}^{C} p(l_k|y,\sigma)$, where $p(l_c|y,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\cdot\exp\{-\frac{(pl+c-1-y)^2}{2\sigma^2}\}$.

6.4. Evaluation metrics

To measure the performance of the proposed method in predicting BP, six popular metrics, namely Mean absolute error (MAE), Mean absolute percentage error (MAPE), Mean error (ME), Standard Deviation (STD), R-square (R^2) and Spearman rank correlation coefficient (SRC) were used. Among them, ME, STD and MAE are three metrics related to the AAMI and the BHS standards [71,72]. R^2 measures the fitting effect of the model, the closer its value is to 1, the better the fitting effect. MAPE measures the mean absolute percentage error. SRC is a non-parametric indicator that measures the dependence/correlation of two variables, the closer its value is to 1, the greater the positive correlation. These metrics are defined as follows,

$$\mathrm{MAE} = \frac{1}{N_{test}}\sum_{i=1}^{N_{test}}|m_i - y_i|, \tag{15}$$

$$\mathrm{MAPE} = \frac{1}{N_{test}}\sum_{i=1}^{N_{test}}\frac{|m_i - y_i|}{y_i}, \tag{16}$$

$$\mathrm{ME} = \frac{1}{N_{test}}\sum_{i=1}^{N_{test}} m_i - y_i, \tag{17}$$

$$\mathrm{STD} = \sqrt{\frac{1}{N_{test}}\sum_{i=1}^{N_{test}}(m_i - y_i)^2}, \tag{18}$$

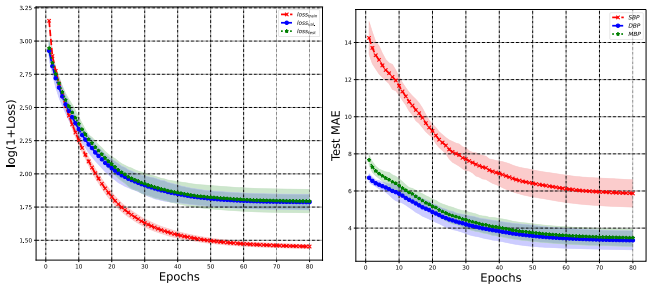
$$\mathrm{R}^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{N_{test}}(y_i - m_i)^2}{\sum_{i=1}^{N_{test}}(y_i - \hat{y})^2}, \tag{19}$$

$$\mathrm{SRC} = 1 - \frac{6\sum_{i=1}^{N_{test}}(rank_{y_i} - rank_{m_i})^2}{n^3 - n}, \tag{20}$$

where N_{test} denotes the total number of test samples, $\hat{y} = \sum_{i=1}^{N_{test}} y_i/N_{test}$, m_i and y_i denote the predicted and the ground-truth BP, respectively.

6.5. Learning the distribution for BP prediction

Fig. 6 presents the behavior of the proposed TFNet-MTD²L during the training process. As Fig. 6(a) illustrates, the training, validation and test losses continue to decrease steadily as the number of training epochs increase, and gradually converge after the 50th epoch, and there is no over-fitting phenomenon. As expected, the predictive performance (i.e. MAE) of the model on the test set has improved steadily. To further understand the behavior of predictive distribution changes during



(a) (b)

Fig. 6. Performance monitoring of the proposed TFNet-MTD^{2}L during the training process. (a) the loss variation; (b) the test performance (i.e. MAE) variation.

training, Fig. 7 shows the trend of the prediction distribution for a given test sample using the model trained after different number of epochs. It is clear that the predicted distribution for the three task gradually approaching (with respect to mean and variance) the corresponding ground-truth distribution.

Compared with classification-based and regression-based models, one of the advantage of the proposed model based on distributed learning is that it can not only directly predict blood pressure, but also output the confidence of the prediction. Fig. 8 illustrates the continuous BP prediction results of the proposed model on ten representative test records, in which different records have different BP levels and various BP change patterns. The top row (a)∼(e) of Fig. 8 shows relatively good results. It is revealed that the trained model can accurately predict SBP, DBP and MBP, and capture different BP change patterns precisely. At the same time, the confidence interval of prediction is relatively tight. The bottom row (f)∼(j) of Fig. 8 indicates the relatively poor results. Specifically, in subfigure (f) of Fig. 8, the model performs well on MBP prediction, but poorly on SBP and DBP predictions (the BP levels between the predicted and the ground truth values differed significantly, although the BP trends were similar between the two). In subfigures (g) and (h) of Fig. 8, the model had significant uncertainty in predicting SBP, DBP and MBP, although the trend in prediction values was similar to that of true BP. In sub-figure (i) of Fig. 8, there is obvious BP level difference between the predicted value and ground-truth value for the three tasks. In sub-figure (j) of Fig. 8, the prediction result has unexpected, completely different BP level and BP change pattern from the ground-truth value for SBP, DBP and MBP prediction. This is mainly due to individual differences. Note that the data is collected from ICU patients and there is such a case where the PPG signals of the two individuals are very similar, but the corresponding BP levels and change patterns are completely different [73]. A practical approach to tackle this question is to calibrate the general model by using partial data of the test individual.

Generally, the essence of learning distribution enables the proposed model with the characteristics of stable training, easy convergence and predictive value accompanied with confidence.

6.6. Learning mode comparison

As mentioned earlier, the training process of the proposed distributed learning-based model with the characteristics of stable training. While, compared with the regression-based method, the performance of the distributed learning-based method is unknown. In addition, from the perspective of machine learning, BP prediction is naturally a multitask learning (MTL) problem. MTL can significantly reduce the amount of parameters, and can be easily implemented in DL-based methods due to the modularity nature of neural network in supporting multiple outputs. However, a considerable part of studies [74,75] in this area adopts the STL mode and there are few articles systematically analyze and compare the two modes. In this section, the following three questions will be studied: (1) *how does distribution learning mode perform relative to regression modeling?* (2) *can MTL mode achieve comparable performance to STL mode?* (3) *Is the answer to question-2 related to the learning modes (distribution learning, regression modeling)?* To answer these questions, we implemented four comparison algorithms named TFNet-MTD^{2}L, TFNet-STD^{2}L, TFNet-MTDRL and TFNet-STDRL, respectively.

Table 1 presents the results. For question (1), model-*a* outperforms model-*c* on almost all metrics. Specifically, in terms of MAE, 17.42%,26.82% and 17.17% improvement are achieved for SBP, DBP and MBP prediction, respectively. This indicates that distribution learning mode helps to improve the generalization ability of the model compared to regression modeling. For question (2), in the distribution learning settings, MTL model-*a* achieves slightly performance gain than STL model-*b* while with only 1/3 times of the number of parameters. However, in the regression modeling settings, MTL model-*c*'s performance is significantly decreased compared to the counterpart STL model-*d*. This indicates that MTL does not necessarily guarantee that the model can achieve the comparative performance as STL for BP prediction given TFNet as feature learner; For question (3), the answer is yes. The reason behind this phenomenon is that, in regression-based methods, the more significant loss scales between different tasks hinders the training of the model in MTL settings.

In fact, distribution learning-based model is significantly superior to regression-based model in both MTL and STL settings (model-*a* vs. model-*c*, and model-*b* vs. model-*d*). In order to further understand the results, in Fig. 9, given the MTL settings, we visualize the difference in predictive accuracy of the two learning modes at different BP levels. It is interesting that both the regression-based model and the distribution-based model perform well in the central area of the possible BP range and perform poorly in areas far away from the central BP region, regardless of the prediction of SBP, DBP and MBP. This is due to the skewed distribution of BP in the training set, making the trained model predicts prefer to central BP region. This is actually imbalance phenomenon that is rarely noticed and mentioned in this area, although it is very important. It can also be seen that distribution-based method can mitigate the bias in model predictions due to imbalanced dataset, which is best viewed in Fig. 9(b). To further quantitatively evaluate the difference of the two learning modes, we propose a new metric—bin-balanced MAE (b^{2}MAE) as follows,

$$\mathrm{b}^2MAE = \frac{1}{N_{bin}} \sum_{i=1}^{N_{bin}} \frac{1}{|\{y_j \in bin_i\}|} \sum_{y_j \in bin_i} |y_j - \hat{y}_j|, \tag{21}$$

where N_{bin} denotes the number of bins with equal length s included in the BP range $[\underline{bp}, \overline{bp}]$, i.e. $N_{bin} = (\overline{bp} - \underline{bp})/s$, the ith bin $bin_i = (\underline{bp} + (i-1)s, \underline{bp} + i \cdot s]$. This metric can give a more objective assessment when the test set is imbalanced, and it degenerated to MAE when the number of test samples per bin is equal.

Table 2 presents the results. Distribution learning-based model—TFNet-MTD^{2}L achieves 11.47%, 25.78% and 0.37% improvements on metric—b^{2}MAE compared to regression-based model—TFNet-MTDRL. This fully explain that distribution learning can alleviates the model bias due to imbalanced dataset to a certain extent (see Fig. 10).

6.7. Model designing validation

In this section, we investigate the network architecture designing from the following three aspects: (1) *Time–frequency module*; (2) *Attention module*; (3) *Input composition*. In the *time–frequency module*, there are three questions to be explored: (1) *Do both the time domain and frequency domain modules in TFNet help improve the performance?* (2) *Does the multi-level output (MLO) module in the time domain feature learner help improve the performance?* (3) *Does the multi-scale extraction (MSE) module in the time domain feature learner help improve the performance?* In the *attention module*, (4) *Does the task-specific attention*

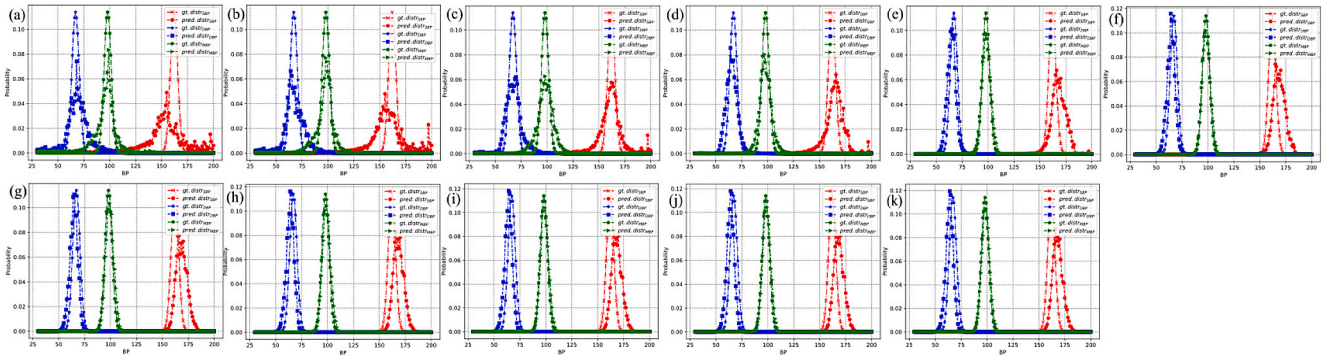


Fig. 7. Comparison results (Ground-truth label distribution vs. the corresponding predicted label distribution) on a test sample using the model trained after different #epochs, the maximum number of #epochs is set to 80. (a)~(k): after the 1th/2nd/5th/10th/20th/30th/40th/50th/60th/70th/80th epochs.

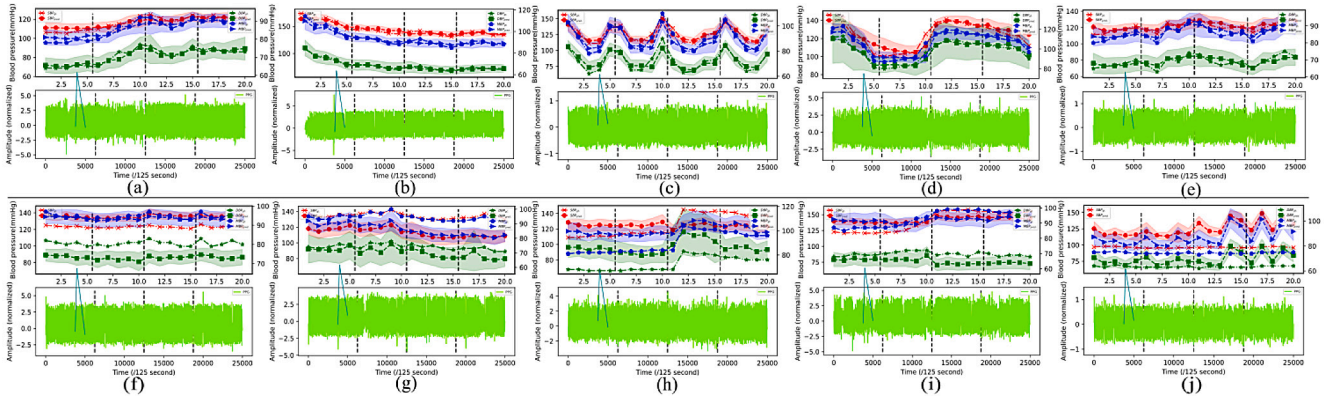


Fig. 8. Continuous blood pressure prediction results on ten representative test records. The top line ((a)~(e)) indicates the good results and the bottom line ((f)~(j)) the relatively poor results. In each subfigure, the comparison for SBP, DBP and MBP predictions is lied above the input PPG signal. The PPG signal is windowed by the black vertical dotted line and each window contains five consecutive sample sequences.

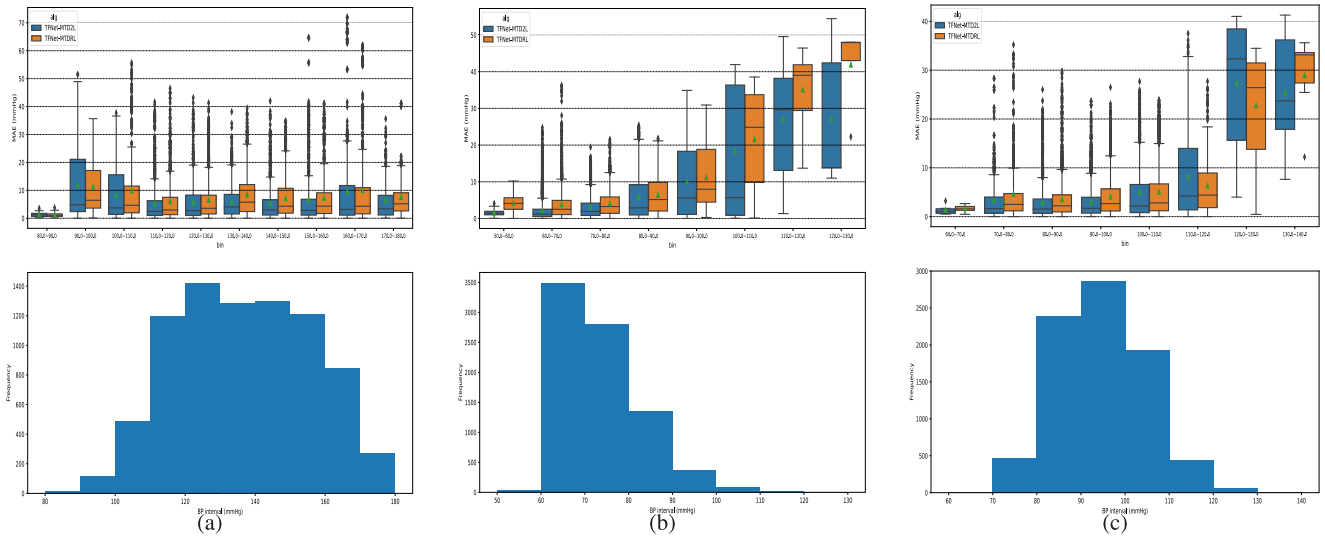


Fig. 9. Comparison of bin error distribution between TFNet-MTDRL and TFNet-MTD^{2}L for BP predictions in an experiment. Each subplot corresponds to a prediction task. For each subplot, the figure below shows the ground-truth BP distribution in the test set, the possible BP range was divided into disjoint bins of width 10 mmHg, and the figure above presents a comparison of the mean absolute error distribution of TFNet-MTDRL and TFNet-MTD^{2}L predictions for each bin within the possible BP range. (a) comparison results for SBP prediction; (b) comparison results for DBP prediction; (c) comparison results for MBP prediction.

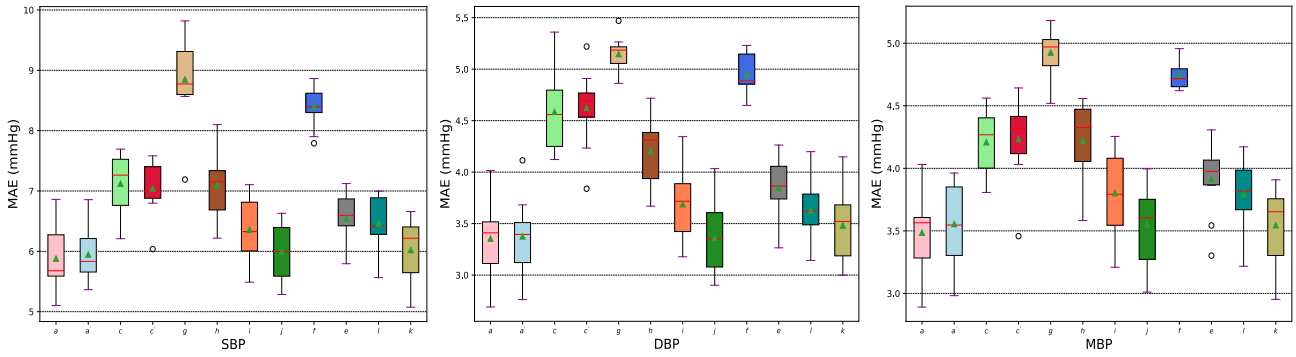


Fig. 10. Boxplots shows the difference in MAE between the twelve related approaches run ten times in the three prediction tasks of SBP, DBP and MBP. Each approach is marked with a letter, and the corresponding relationship between the two is shown in Fig. 5, x' denotes the naive version of approach x. (a) SBP prediction; (b) DBP prediction; (c) MBP prediction.

Table 1
Comparison results of different learning modes.

Flag	Method	Task	Metrics (unit: mmHg)					
			MAE↓	MAPE↓	ME↓	STD↓	R^2 ↑	Spearman↑
a	TFNet-MTD^{2}L	SBP	5.8815	0.0438	0.4796	8.9264	0.6065	0.8031
		DBP	3.3549	0.0445	0.3857	5.0826	0.5079	0.7588
		MBP	3.4859	0.0365	0.3974	5.1916	0.5502	0.7694
c	TFNet-MTDRL	SBP	7.1221	0.0529	0.9980	9.9856	0.5488	0.7767
		DBP	4.5847	0.0613	0.3372	6.2920	0.3053	0.6430
		MBP	4.2087	0.0443	0.4765	5.8201	0.4717	0.7315
b	TFNet-STD^{2}L	SBP	5.9171	0.0440	0.5795	9.1109	0.5881	0.7998
		DBP	3.4684	0.0460	0.4070	5.3062	0.4518	0.7371
		MBP	3.6984	0.0388	0.4723	5.5402	0.4947	0.7463
d	TFNet-STDRL	SBP	6.6793	0.0497	0.6589	9.4655	0.5849	0.7881
		DBP	3.8844	0.0515	0.5251	5.6445	0.4173	0.7207
		MBP	4.0962	0.0432	0.4829	5.7314	0.4734	0.7328

Table 2
Comparison of b^{2}MAE between regression modeling and distribution learning for BP prediction.

Flag	Method	b^{2}MAE↓ (unit: mmHg)		
		SBP	DBP	MBP
a	TFNet-MTD^{2}L	6.6293	11.8710	9.5654
c	TFNet-MTDRL	7.4884	15.9954	9.6007

(TPA) module help improve the performance? In the *input composition*, (5) *Does the derivative of the signal help improve the performance?* To answer these questions, eight additional methods named TNet-MTD^{2}L, FNet-MTD^{2}L, TFNet-MTD^{2}L no mlo, TFNet-MTD^{2}L no mse, TFNet-MTD^{2}L TSA, TFNet-MTD^{2}L no AM, TFNet-MTD^{2}L x+x' and TFNet-MTD^{2}L x are implemented. A graphical comparison of these methods for SBP, DBP and MBP prediction is illustrated in Boxplot in Fig. 10(a)~(c), respectively.

Time–frequency module. Table 3 presents the results. For question (1), using only time domain module, the predictive MAE has drops by 50.45%, 53.41%, 41.30% for SBP, DBP and MBP, respectively (model-*g* vs. model-*a*). Using only frequency domain module, the predictive MAE has drops by 20.65%, 25.32%, 21.04% for SBP, DBP and MBP prediction, respectively (model-*h* vs. model-*a*). This indicates that: (*i*) frequency domain information seems more effective than time domain information for BP prediction, which implies the advantages of processing signal in frequency domain; (*ii*) frequency domain and time domain information complement each other, and their combination significantly improves the predictive performance. For question (2), the exclusion of the MLO module has drops the predictive MAE by 8.11%, 9.92%, 9.05% for SBP, DBP and MBP prediction, respectively (model-*i* vs. model-*a*). This indicates that the concentration of different levels of decoder layer in the U^{2}Net module helps to extract more informative features for BP prediction. For question (3), the exclusion of the MSE module has drops the predictive MAE by 1.90%, 0.19%, 1.90% for SBP, DBP and MBP prediction, respectively (model-*j* vs. model-*a*). This indicates that using MSE module instead of identity mapping helps to learn more informative sequence that contributes to BP prediction.

Attention module. For question (4), Table 4 presents the results. Firstly, the exclusion of TPA module has drops the predictive MAE by 9.86%, 8.15%, 8.78% for SBP, DBP and MBP prediction, respectively (model-*l* vs. model-*a*). This indicates that paying attention to the features in the learned feature vector by different degrees can effectively improve the predictive ability. In addition, the replacement of the TPA module with the TSA module decreased the predictive MAE by 2.39%, 3.77%, 1.70% for SBP, DBP and MBP prediction, respectively (model-*k* vs. model-*a*). This further indicates that the features in the learned feature vector have different importance/contribution to different prediction tasks, and paying attention to different tasks by different attention weights helps to improve the predictive ability. In fact, both the performance of model-*k* and model-*l* is inferior to the performance of the corresponding single-task learning model-*b* (refer Table 1), which fully shows the existence of differences between the most informative feature sets for different learning tasks and the effectiveness of the TPA module. Fig. 11 shows an illustrative example explaining the role of the TPA module.

Input composition. For question (5), Table 5 presents the results. Using the 1st derivative of PPG signal improves the predictive MAE by 22.14%, 22.32%, 17.61% for SBP, DBP and MBP prediction, respectively (model-*e* vs. model-*f*). Furthermore, using the 2nd derivative of PPG signal improves the predictive MAE by 10.10%, 12.73%, 10.92% for SBP, DBP and MBP prediction, respectively (model-*a* vs. model-*e*). This indicates that PPG's derivatives contain more informative features that contribute to BP prediction.

Table 3
Comparison results of the TFNet module, the mlo module, and the mse module.

Flag	Method	Task	Metrics (unit: mmHg)					
			MAE↓	MAPE↓	ME↓	STD↓	R^2 ↑	Spearman↑
a	TFNet-MTD^{2}L	SBP	5.8815	0.0438	0.4796	8.9264	0.6065	0.8031
		DBP	3.3549	0.0445	0.3857	5.0826	0.5079	0.7588
		MBP	3.4859	0.0365	0.3974	5.1916	0.5502	0.7694
g	TNet-MTD^{2}L	SBP	8.8487	0.0659	0.7319	11.6466	0.4339	0.7067
		DBP	5.1466	0.0691	0.3613	6.8713	0.1784	0.5811
		MBP	4.9255	0.0521	0.3075	6.5114	0.3639	0.6619
h	FNet-MTD^{2}L	SBP	7.0963	0.0527	0.6809	10.1248	0.5275	0.7610
		DBP	4.2042	0.0561	0.4203	5.9519	0.3518	0.6707
		MBP	4.2193	0.0443	0.4345	5.9627	0.4384	0.7085
i	TFNet-MTD^{2}L, no mlo	SBP	6.3586	0.0472	0.5083	9.3574	0.5824	0.7900
		DBP	3.6878	0.0492	0.3245	5.4213	0.4478	0.7223
		MBP	3.8014	0.0399	0.4206	5.5242	0.5002	0.7441
j	TFNet-MTD^{2}L, no mse	SBP	5.9931	0.0445	0.6464	9.1297	0.5838	0.7987
		DBP	3.3613	0.0446	0.3956	5.0818	0.4961	0.7513
		MBP	3.5522	0.0372	0.4728	5.2890	0.5293	0.7631

Table 4
Comparison results of the attention module.

Flag	Method	Task	Metrics (unit: mmHg)					
			MAE↓	MAPE↓	ME↓	STD↓	R^2 ↑	Spearman↑
a	TFNet-MTD^{2}L	SBP	5.8815	0.0438	0.4796	8.9264	0.6065	0.8031
		DBP	3.3549	0.0445	0.3857	5.0826	0.5079	0.7588
		MBP	3.4859	0.0365	0.3974	5.1916	0.5502	0.7694
k	TFNet-MTD^{2}L, TSA	SBP	6.0223	0.0447	0.5078	9.1032	0.5993	0.7979
		DBP	3.4814	0.0462	0.3497	5.1829	0.4936	0.7466
		MBP	3.5450	0.0372	0.3978	5.2589	0.5367	0.7645
l	TFNet-STD^{2}L, no AM	SBP	6.4616	0.0479	0.6274	9.6212	0.5737	0.7821
		DBP	3.6283	0.0483	0.3341	5.3704	0.4732	0.7346
		MBP	3.7921	0.0398	0.4134	5.5584	0.5073	0.7435

Table 5
Comparison results of the proposed model with different inputs.

Flag	Method	Task	Metrics (unit: mmHg)					
			MAE↓	MAPE↓	ME↓	STD↓	R^2 ↑	Spearman↑
a	TFNet-MTD^{2}L	SBP	5.8815	0.0438	0.4796	8.9264	0.6065	0.8031
		DBP	3.3549	0.0445	0.3857	5.0826	0.5079	0.7588
		MBP	3.4859	0.0365	0.3974	5.1916	0.5502	0.7694
e	TFNet-MTD^{2}L, x+x'	SBP	6.5421	0.0487	0.3589	9.5532	0.5654	0.7786
		DBP	3.8444	0.0512	0.4181	5.6057	0.4119	0.7093
		MBP	3.9133	0.0411	0.4439	5.6480	0.4877	0.7371
f	TFNet-MTD^{2}L, x	SBP	8.4019	0.0627	0.8104	11.2204	0.4595	0.7184
		DBP	4.9488	0.0663	0.4382	6.7349	0.2029	0.5956
		MBP	4.7499	0.0501	0.4231	6.4472	0.3770	0.6755

6.8. Validation of adaptive multitask weighted loss

In related studies in this area, the work on training predictive models based on MTL usually utilizes the naive weighting strategy—the total loss is computed as the average of multiple task losses. However, as Fig. 12(a) illustrates, the loss scale of different tasks is significantly different and constantly changes during the training process. Simply averaging the losses of different tasks in MTL scenario may lead to a situation where the difficult task dominants the gradient when updating the model parameters, hindering the training of the model. In this section, to validate the effect of task-dependent uncertainty-based loss weighting strategy in different learning modes, two additional methods named TFNet-MTD^{2}L$_{naive}$ and TFNet-MTDRL$_{naive}$ are implemented.

Table 6 presents the comparison results. In distribution learning settings, the utilization of adaptive loss weighting improves the predictive MAE by 1.11%, 0.61%, 1.93% for SBP, DBP and MBP prediction, respectively (model-a vs. model-a'). Besides, as Fig. 12(b)~(c) illustrates, the weight of hard task—SBP is adaptively suppressed in the training process. This indicates that by balancing the loss scales of different tasks during training, the generalization ability of the model can be improved. Note that this improvement is desirable with almost no additional cost (only six additional parameters are required). In regression-based scenario, the utilization of adaptive loss weighting improves the predictive MAE by −1.15%, 0.85%, 0.60% for SBP, DBP and MBP prediction, respectively (model-c vs. model-c'). It can be seen that the predictive accuracy on the simple tasks—DBP and MBP has been slightly improved, but the accuracy on the difficult task—SBP is excessively suppressed. Note that the uncertainty-based loss weighting strategy is equivalent to applying a logarithmic operation to the losses of different tasks, and the larger the loss value, the stronger the suppression.

6.9. Hyperparameter sensitivity analysis & verification of loss item

Since hyper parameters—μ, τ, ρ, λ in Eq. (8) balance the five loss terms. In addition, σ determines the degree of steepness or flatness of the prior normal distribution for generating label distribution. We experimentally evaluated the effect of the five parameters.

Table 6
Comparison results of the multitask loss weighting strategy.

Flag	Method	Task	Metrics (unit: mmHg)					
			MAE↓	MAPE↓	ME↓	STD↓	R^2 ↑	Spearman↑
a	TFNet-MTD^{2}L	SBP	5.8815	0.0438	0.4796	8.9264	0.6065	0.8031
		DBP	3.3549	0.0445	0.3857	5.0826	0.5079	0.7588
		MBP	3.4859	0.0365	0.3974	5.1916	0.5502	0.7694
a′	TFNet-MTD^{2}L$_{naive}$	SBP	5.9478	0.0442	0.5342	9.1163	0.5946	0.7964
		DBP	3.3756	0.0447	0.4941	5.0584	0.5090	0.7553
		MBP	3.5546	0.0372	0.4980	5.2781	0.5389	0.7625
c	TFNet-MTDRL (L_2)	SBP	7.1221	0.0529	0.9980	9.9856	0.5488	0.7767
		DBP	4.5847	0.0613	0.3372	6.2920	0.3053	0.6430
		MBP	4.2087	0.0443	0.4765	5.8201	0.4717	0.7315
c′	TFNet-MTDRL$_{naive}$(L_2)	SBP	7.0412	0.0523	1.0525	9.8671	0.5544	0.7788
		DBP	4.6239	0.0618	0.5174	6.3374	0.2750	0.6406
		MBP	4.2342	0.0446	0.5200	5.8743	0.4608	0.7280

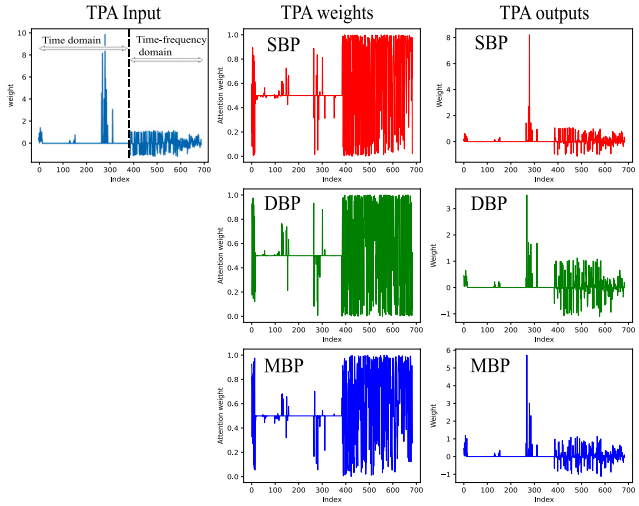


Fig. 11. An example interpreting the role of the TPA module. The TPA module helps to improve the predictive accuracy by assigning the learned time domain and time–frequency domain information with different, task-dependent weights.

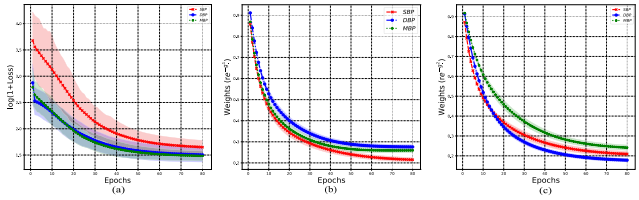


Fig. 12. The variation trend of the task losses and task weights during training. (a) the variation trend of the task losses; (b) the variation trend of the task weights of the *mean loss* term; (c) the variation trend of the task weights of the *variance loss* term.

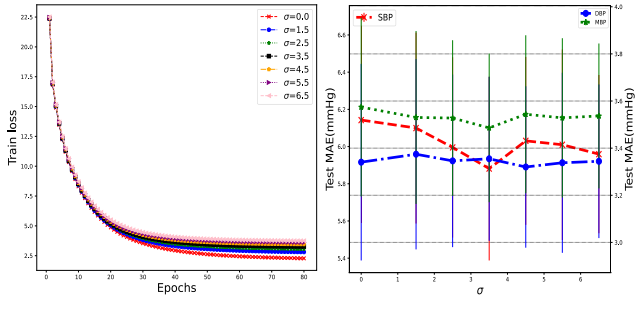


Fig. 13. The performance (MAE) of TFNet-MTD^{2}L with different prior variance—σ^2 values. (a) the training loss curve; (b) the test MAE.

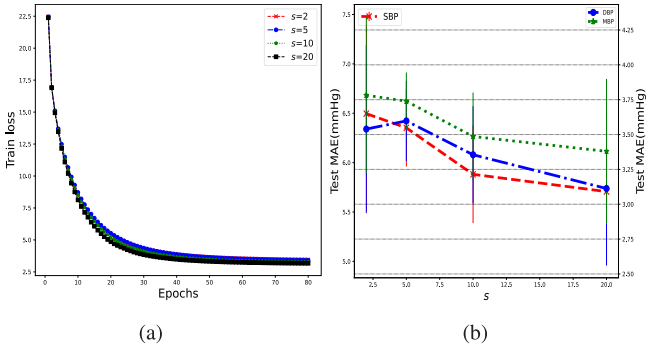


Fig. 14. The performance (MAE) of TFNet-MTD^{2}L with different sequence lengths—s values. (a) the training loss curve; (b) the test MAE.

The effect of σ. For σ, the smaller the σ value, the sharper the ground-truth label distribution. Especially, $\sigma = 0.0$ means the normal distribution assumption has degenerated to one-point distribution, i.e. $\hat{y}_c = 1$ if $pl + c - 1 = y$; else, 0. The parameter takes the values $\{0.0, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5\}$. As Fig. 13(a) illustrates, the training loss curve is always smooth, and the convergence speed is always very fast for different σ values. Fig. 13(b) depicts the variation in performance (MAE) of TFNet-MTD^{2}L when σ is set to different values. Overall, the performance of TFNet-MTD^{2}L keeps relative stable with different σ value, and the best generalization performance is acquired when σ takes 3.5. σ is set to 3.5 in the following experiments.

The effect of s. s indicates the length of the input sequence. For s, the parameter takes the values $\{2, 5, 10, 20\}$. As Fig. 14(a) illustrates, the training loss curve is smooth, and the larger the value of s, the faster the convergence speed. Fig. 14(b) shows the test performance of TFNet-MTD^{2}L given different s values. As expected, the larger the value of s, the better the prediction. However, large s value means more time delay is required before making predictions, which is a problem to be considered in practice.

The effect of μ, τ, ρ *and* λ. The results were shown in Fig. 15. For μ, the parameter takes the values $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. As Fig. 15(a) illustrates, the join of the *mean loss* term (i.e. $\mu > 0$, in contrast to exclusion when $\mu = 0$) can significantly improve the predictive performance. Finally, μ is set to 0.2. For τ, the parameter takes the values {0.0, 5e-6, 1e-5, 6e-5, 1e-4, 1e-3, 2e-3, 3e-3}. As Fig. 15(b) shows, the join of the *variance loss* term (i.e. $\tau > 0$, in contrast to exclusion when $\tau = 0$) improves the performance, and within a certain range, the larger the τ, the more obvious the performance improvement. Unlike the *mean loss* term, a single *variance loss* cannot be used as an optimization target and is more likely to cause instability in the training process. In addition, according to the Eq. (3), the final BP is computed as mean (weighted sum) of the BP

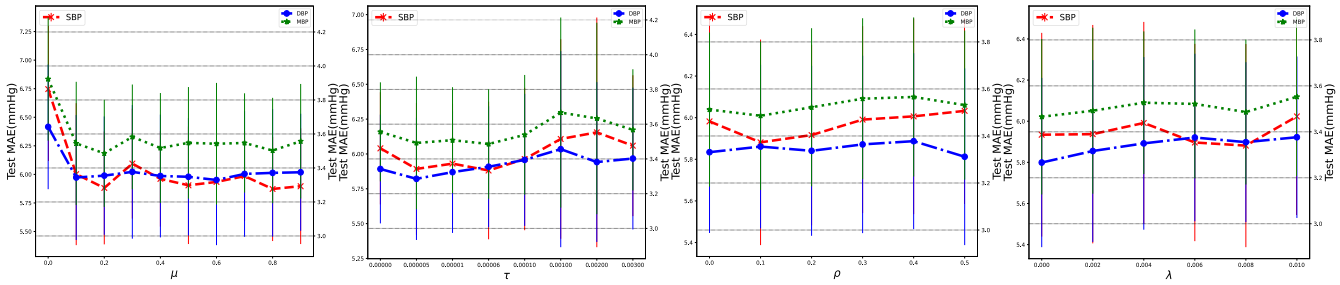


Fig. 15. Hyper parameters sensitivity analysis. (a) MAE vs. parameter μ; (b) MAE vs. τ; (c) MAE vs. ρ; (d) MAE vs. λ.

Table 7
Evaluation with the BHS standard. The standard deviation of the results of 10 runs in parentheses.

Method/ Standard	Task	Proportion of the subjects with MAE satisfying:			Grade
		≤5 mmHg	≤10 mmHg	≤15 mmHg	
TFNet-MTD^{2}L	SBP	67.179(2.63)%	82.055(2.03)%	88.984(1.64)%	≈ B
	DBP	80.721(2.73)%	92.456(1.55)%	96.393(0.93)%	A
	MBP	78.944(2.29)%	91.495(1.45)%	96.430(0.88)%	A
BHS	–	60%	85%	95%	Grade A
		50%	75%	90%	Grade B
		40%	65%	85%	Grade C

Table 8
Evaluation with the AAMI standard. The standard deviation of the results of 10 runs in parentheses.

Method/ Standard	Task	Metrics (unit: mmHg)		#Subject (test)	Pass
		ME	STD		
TFNet-MTD^{2}L	SBP	0.4796(0.3899)	8.9264(0.7658)	415	No
	DBP	0.3857(0.3554)	5.0826(0.5657)		Pass
	MBP	0.3974(0.2745)	5.1916(0.4750)		Pass
AAMI	–	≤ 5	≤ 8	≥ 85	–

distribution w.r.t the label space. Therefore, the value of τ is much smaller than μ. τ is set to 6e-5 in experiment. For ρ, the parameter takes the values $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. As Fig. 15(c) presents, the join of the *inconsistency loss* term (i.e. $\rho > 0$, in contrast to exclusion when $\rho = 0$) can improve the performance to a certain extent, if a suitable ρ value is set. ρ is set to 0.1. For λ, the parameter takes the values $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. As Fig. 15(d) shows, the join of the *regularization term* (i.e. $\lambda > 0$, in contrast to exclusion when $\lambda = 0$) can improve the performance of SBP prediction if λ is set properly. λ is set to 0.8. In conclusion, (1) the addition of each loss term helps to improve the predictive performance to varying degrees; (2) the performance of TFNet-MTD^{2}L is relative stable w.r.t the weight of each loss term; (3) the weight of the *variance loss* term is far less than that of the *mean loss* term.

6.10. Evaluation

Table 7 presents an evaluation of the proposed method by the British Hypertension Society (BHS) standard [71]. The test device achieves Grade A, Grade B or Grade C if the corresponding condition is satisfied. According to the BHS criterion, the proposed method is consistent with the grade A in the estimation of DBP and MBP. While, the proposed method predicts SBP very close to level B and far exceeds level C. Specifically, the proportion of the subjects with MAE no more than 5 mmHg and 10 mmHg has achieved 67.179% and 82.055%, respectively, which fully meets the grade B. However, there is about 1% gap compared with the entry criterion of the grade B with respect to the condition of ≤ 15 mmHg.

Table 8 presents an evaluation of the proposed method by the AAMI standard [72]. The test device meets the AAMI standard if its precision must not differ from the mercury standard by a mean error of ≤ 5 mmHg or a standard deviation of ≤ 8 mmHg. According to the AAMI standard, the proposed method has fully meets the AAMI standard in the estimation of DBP and MBP. However, for SBP prediction, there is a little gap compared with the standard. Specifically, the proposed method has achieved ME of 0.4796 and STD of 8.9264, which is over the limits of ≤ 8 mmHg on the metric of STD.

Fig. 16 illustrates the Bland–Altman plots, error plots, and regression plots for the estimates our TFNet-MTD^{2}L model versus the corresponding reference arterial BP values in an experiment. Specifically, according to the Bland–Altman recommendations, two methods are comparable if 95% of samples were fall within the agreement of limits (area within two black dashed lines). As Fig. 16(a)~(c) presents, the difference between the invasive arterial catheter and the estimates of TFNet-MTD^{2}L model is plotted against the average of the two methods. Of the 8300 samples from 415 test records, 92.76%, 93.48% and 92.72% achieve this agreement for SBP, DBP and MBP prediction, respectively. Among the ten runs, an average of 92.63%, 93.89% and 93.35% achieve this agreement. Besides, as Fig. 16(d)~(f) illustrates, we additionally show the results of the error plot, which differs from the Bland–Altman plot in that the variable Means (i.e. $(BP_{ground\_truth} + BP_{pred})/2$) on the horizontal axis is replaced by the ground-truth BP value. It can be found that the overall distribution of the error plot is very similar to the corresponding Bland–Altman plot, with the difference that the former has a certain skew. Specifically, the BP of samples with ground truth BP much smaller than the center of the possible BP range tends to be overestimated because most of the corresponding points fall below the mean difference line, while the BP of samples with ground truth BP much larger than the center of the possible BP range tends to be underestimated. The Pearson correlation coefficient—r in regression plot reflects how linearly correlated two sets of data are. As Fig. 16(g)~(i) presents, r is 0.829, 0.753 and 0.765 for SBP, DBP and

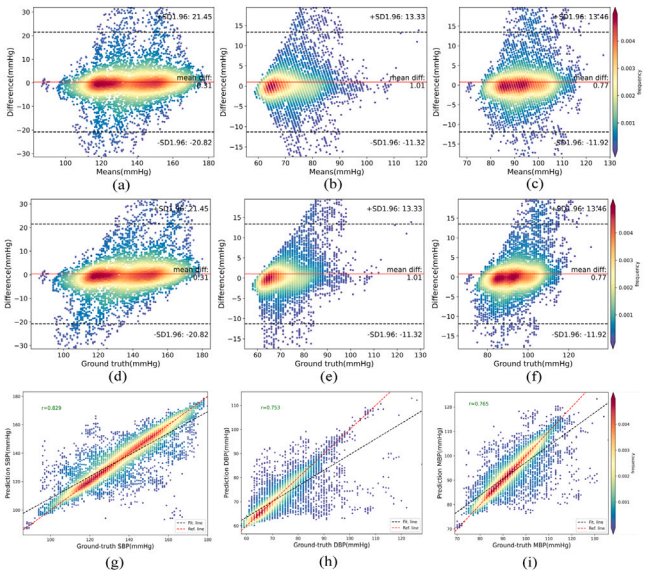


Fig. 16. Density Bland–Altman plots, density error plots, and density correlation plots of the proposed method—TFNet-MTD^{2}L for SBP, DBP and MBP predictions in an experiment. (a)~(c) the density Bland–Altman plots of SBP, DBP and MBP predictions, respectively; (d)~(f) the density error plots of SBP, DBP and MBP predictions, respectively; (g)~(i) the density correlation plots of SBP, DBP and MBP predictions, respectively. The blue solid line and the red dotted line indicate the fitted line and the reference line, respectively.

MBP prediction, respectively. Among the ten runs, an average of 0.854, 0.781 and 0.804 is achieved. These results indicate that there is a high degree of agreement between the predicted blood pressure of our model and the blood pressure measured by the arterial catheter.

6.11. Comparison with state-of-the-art methods/systems

How to make an objective comparison between the work in this area (even for those using the same data source) is difficult and challenging due to several reasons such as: (1) different preprocessing procedures, data cleaning rules, splitting strategies and splitting percentage for generating training set and test set, and evaluation procedure used, etc.; (2) the unavailability of code and the final experimental data [25,76]. Therefore, the mainstream practice for comparison is comparing a system directly [7,12–14,18,20–22,25,48,77]. In this section, in order to fairly and comprehensively compare the proposed TFNet-MTD^{2}L with the state-of-the-art methods/systems, we compare them from the following two aspects: (*i*) comparison using the identical evaluation procedure and the same final dataset; (*ii*) direct comparison with other systems.

6.11.1. Comparison using the identical evaluation procedure and dataset

We selected RF [78], FCN [18], STDNN [17], CNN-RNN-AM [22] and MTL pso [13] for comparison. Random forest is recognized as one of the best algorithms of BP prediction and is considered as the representative of traditional methods. Specifically, depending on the features used for training, we implemented two RF versions called RF (wbf) and RF (PCA). The rest four models belong to deep learning (DL) methods, where BP prediction is modeled as a regression task. As in our model, all these methods use raw signal as inputs. For fairness, the maximum number of training epochs is set to 80 for all deep learning methods. All methods were evaluated based on the identical training, validation and test sets, using the same evaluation procedure, with PPG signal as the only input source. In addition, *normalize target* is a technique used for scaling the range of the target (i.e BP) during training of regression model, which is an important factor affecting the performance of the model. In particular, note that if the *normalize target* technique is used during training, the prediction must be reversely normalized during the test/inference phase. Therefore, we trained two different versions of the model independently for each algorithm, one of which uses *normalize target* technique during training. Additionally, we implemented a baseline model named Mean predictor [76], which always responds the mean BP value of the training samples to any test sample. A method with good generalization ability should exceed the performance of the Mean predictor.

- Mean predictor: A fixed predictor that always responds the mean SBP, DBP and MBP of all the training samples for any test samples.
- RF: The model is trained based on Random forest [78]. Depending on the input feature used, it includes: (1) RF (wbf): the whole-based feature [7,21] extracted from raw signal is used as input; (2) RF (PCA): the reduced feature derived from raw signal based on PCA transformation is used as input.
- FCN [18]: The fully convolution network (FCN) contains two parts: encoder and decoder. L_1 distance is used to compute loss.
- STDNN [17]: The spectro-temporal deep neural network (STDNN), where the temporal representation is extracted by stacking the so-called ResNet block and the spectral information is extracted by the spectral layer. MAE is used to compute loss.
- CNN-RNN-AM [22]: A VGGNet-style convolution network is followed by a bidirectional GRU, and its output is weighted through an attention module to capture the time dependency.
- MTL pso [13]: A Bi-LSTM based multitask neural network is developed for BP prediction, and the task weights are updated based on particle swarm optimization (PSO) during the fine-tuning stage.

Table 9 presents the results. The performance of the proposed method far exceeds that of the Mean predictor, which indicates that the method is effective and capable of good generalization ability far beyond the Mean predictor. As for traditional methods, the performance of RF (wbf) exceeds that of RF (PCA) by a large margin. *normalize target* does not play a significant role in the performance improvement of both RF (wbf) and RF (PCA). The proposed method has achieved superior performance over RF (wbf). Specifically, we have achieved 23.99%, 20.25% and 22.51% improvement from the *normalize target* version of RF (wbf) on MAE for SBP, DBP and MBP prediction, respectively. As for deep learning methods, it seems that neither FCN nor CNN-RNN-AM can outperform the performance of the Mean predictor, where the only difference is that *normalize target* has a great impact on the performance of the latter. In fact, we observed severe gradient domination phenomenon during the training of CNN-RNN-AM model, which we believe is related to the design of the network. Note that in the original paper [22], CNN-RNN-AM is evaluated on a privately collected data set consisting of only 15 subject records, and the strategy of splitting at final sample level for generating training, validation and test sets, which is at the risk of data leakage, was used for experiment. Both STDNN and MTL pso achieved superior generalization ability over Mean predictor. However, both of the no *normalize target* version of them are inferior to the proposed method in performance. Specifically, we achieved a 40.12%, 42.05% and 36.28% improvement in predictive MAE from the no *normalize target* version of STDNN, 30.28%, 25.33% and 27.21% improvement in predictive MAE from the no *normalize target* version of MTL pso. In addition, *normalize target* plays an important role in improving the performance of STDNN and MTL pso. Specifically, the inclusion of the *normalize target* procedure improves the predictive MAE of STDNN by a 8.20%, 20.51% and 12.16%, and improves the predictive MAE of MTL pso by a 33.98%, 31.43% and 31.57%.

6.11.2. Comparison with other systems

As shown in Table 10, in addition to performance, we try to list the differences between different methods to reflect these methods as comprehensively and objectively as possible, rather than competing with each other. The selected systems for comparison are mainly that evaluated on relatively large datasets (especially patient data).

Table 9
Comparison results with representative methods using the same evaluation procedure, metrics and dataset (only PPG signal used). The results of each algorithm were presented with and without *normalize target* procedure. All methods were evaluated without calibration procedures.

Method	Norm. target	Task	Metrics (unit: mmHg)					
			MAE↓	MAPE↓	ME↓	STD↓	R^2 ↑	Spearman↑
TFNet-MTD^{2}L	–	SBP	5.8815	0.0438	0.4796	8.9264	0.6065	0.8031
		DBP	3.3549	0.0445	0.3857	5.0826	0.5079	0.7588
		MBP	3.4859	0.0365	0.3974	5.1916	0.5502	0.7694
Mean predictor	–	SBP	15.7515	0.1183	0.8623	18.8426	0.2225	*NaN*
		DBP	7.1893	0.0973	0.4046	9.1786	1.3625	*NaN*
		MBP	7.7908	0.0831	0.3580	9.6716	1.1789	*NaN*
RF (wbf)	Yes	SBP	7.7376	0.0577	0.5497	10.4485	0.6912	0.8417
		DBP	4.2067	0.0566	0.2460	6.0182	0.5682	0.7693
		MBP	4.4985	0.0475	0.3166	6.1669	0.5921	0.7989
	No	SBP	7.7377	0.0577	0.5534	10.4511	0.6910	0.8415
		DBP	4.2023	0.0565	0.2448	6.0119	0.5691	0.7698
		MBP	4.4996	0.0475	0.3155	6.1681	0.5919	0.7988
RF (PCA)	Yes	SBP	11.2126	0.0844	0.6659	14.0155	0.4444	0.7012
		DBP	5.9394	0.0802	0.3373	7.8600	0.2642	0.5352
		MBP	6.2600	0.0666	0.4039	7.9014	0.3307	0.6618
	No	SBP	11.2119	0.0844	0.6666	14.0144	0.4445	0.7011
		DBP	5.9410	0.0802	0.3364	7.8620	0.2638	0.5345
		MBP	6.2613	0.0666	0.4080	7.9020	0.3306	0.6619
FCN	Yes	SBP	15.6816	0.1188	0.4875	18.3536	0.2385	0.0051
		DBP	7.1847	0.0950	1.6662	8.9391	0.2180	0.0151
		MBP	8.0847	0.0857	0.8859	9.7232	0.2161	0.0072
	No	SBP	15.6182	0.1178	0.7713	18.2823	0.2308	0.2402
		DBP	7.1989	0.0943	2.2955	9.0432	0.2360	0.0552
		MBP	8.0624	0.0852	1.2098	9.7132	0.2128	0.1789
STDNN	Yes	SBP	9.0169	0.0666	1.5365	11.7812	0.4044	0.6933
		DBP	4.6023	0.0614	0.6974	6.1799	0.3025	0.6418
		MBP	4.8049	0.0504	0.8416	6.3495	0.3737	0.6761
	No	SBP	9.8225	0.0726	2.4012	12.4995	0.3469	0.6777
		DBP	5.7896	0.0774	1.3049	7.4445	0.0962	0.4960
		MBP	5.4703	0.0574	1.4131	7.0404	0.2557	0.6271
CNN-RNN-AM	Yes	SBP	15.6871	0.1185	0.3231	18.3526	0.2376	*NaN*
		DBP	7.3397	0.0996	0.2990	8.9002	0.2658	*NaN*
		MBP	8.0996	0.0867	0.2472	9.7031	0.2185	*NaN*
	No	SBP	35.8727	0.2486	35.7299	39.7977	5.6018	0.1289
		DBP	28.4301	0.4112	28.2423	29.6213	18.3995	0.0366
		MBP	10.0959	0.1139	6.9177	11.8915	1.0710	0.1126
MTL PSO[a]	Yes	SBP	5.5688	0.0414	0.4602	8.5543	0.6436	0.8187
		DBP	3.0812	0.0409	0.2230	4.8641	0.5041	0.7733
		MBP	3.2771	0.0345	0.2170	5.0571	0.5479	0.7782
	No	SBP	8.4354	0.0628	1.3439	11.2541	0.4471	0.7199
		DBP	4.4932	0.0599	0.7383	6.2362	0.2955	0.6581
		MBP	4.7893	0.0506	0.9112	6.4234	0.3552	0.6767

[a]Note that in our implemented version, the so-called conditioned loss is deleted because each task loss exceeds 5 in our experiment, so the total loss is always constant, and the model training cannot start actually. Besides, the SGD optimizer used in the original paper [13] is replaced with Adam optimizer, which brings better prediction performance.

Generally, the proposed method has achieved superior performance over these systems, while using only PPG signal and no additional *normalize target* procedure. We draw several opinions as below: (1) the results of the studies based on outpatients or healthy individual's data (e.g. [14,17,29]) are generally better than that based on ICU patients (e.g [7,25,28,77]). This is reasonable since individual's signal (e.g. PPG and ECG, etc.) is affected by several factors such as drugs, diseases and other factors [8]. An intuitive evidence is that the possible BP range of data collected from ICU patients is much larger (especially SBP) than that collected from outpatients or healthy individuals, which undoubtedly increases the difficulty of prediction; (2) the splitting strategy plays a crucial role in influencing the predictive performance, however, this issue has received little attention and discussion in previous studies. For example, in study [12], the replacement of the splitting strategy of '*rl*' with '*sl*' improves the predictive MAE by 52.23%, 30.63% and 41.63% for SBP, DBP and MBP prediction, respectively. However, good results do not necessarily mean good generalization ability. Specifically, the individual's waveform is highly regular and relatively stable [76], the splitting strategy of '*sl*' will cause data of a record appears simultaneously in training, validation and test sets, which is at the risk of data leakage [18,76] and may violates the basic independent-identical-distribution (I.I.D) principle in the evaluation of ML algorithm. This is the main reason for the apparently unrealistic results appeared in some studies; (3) individual differences is a key factor relating to the test performance. Due to individual differences, the general model trained from other individual's data perform poorly on unknown test individuals, and partial data of test individual is used to calibrate/fine-tune the general model. For example, in study [7], after using the one-point calibration procedure, the calibrated method outperforms the calibration-free method with a considerable margin. Similar results can also be found in studies [18,25,28,29,77]. Another solution to overcome individual differences is to first divide the training set into several disjoint subsets according to physiological conditions (e.g. BP category [12]), where each subset is used to train a prediction model, and then integrate these models for final prediction. However, this undoubtedly increase the storage burden and the complexity of

Table 10
Comparison results of the proposed method with current systems. '–' denotes not applicable.

Literature	Year	Data source	#Data used	Health state	Signal used	Method	Task corr.[a]	Input type	Split strategy[b]	Norm. target	Calibration	MAE (unit: mmHg))		
												SBP	DBP	MBP
[25]	2019	MIMIC-III	510 subjects, about 500000 samples	ICU patients	PPG, ABP	neural network	MTL	raw signal	*LOSO*	No	No Yes	15.41 9.43	12.38 6.88	– –
[7]	2016	MIMIC-II	3663 records, 3663 samples	ICU patients	PPG, ECG, ABP	ML(Adaboost)	STL	hand-crafted features	*rl*	No No	No Yes	11.17 8.21	5.35 4.31	5.92 –
[28]	2017	MIMIC-II	105 records, 8380 samples	ICU patients	PPG, ABP	Dictionary learning +ML (Random forest)	STL	raw signal with single cycle	*rl*	No	No Yes	17.08 5.04	10.77 2.99	– –
[14]	2019	privately collected	85 subject, 2720 samples	Outpatients	ECG, PPW, cuff-based BP	MLR (Multiple linear regression)	STL	hand-crafted features	Individual test	No	–	6.13	5.52	6.03
[12]	2020	4 data sources	51 subjects, 3129 samples	healthy, patients	ECG, ABP	ML (Random forest)	STL	hand-crafted features	*rl* *sl*	No	No –	16.60 7.93	9.24 6.41	9.80 5.72
[13]	2019	MIMIC II	Unknown	ICU patients	ECG, ABP	neural network	MTL	raw signal	*rl*	No	No	7.16	3.89	4.24
[21]	2019	MIMIC II	1323 records, 3969 samples	ICU patients	PPG, ABP	ML(AdaBoost)	STL	whole-based features	*sl*	No	–	3.97	2.43	2.61
[17][c]	2018	privately collected	84 subjects,	healthy	PPG, ECG, ABP	neural network	MTL	hand-crafted features	*sl*	Yes	–	3.73*	2.43*	–
[79]	2020	MIMIC I	28 records,	ICU patients	PPG, ABP	Multi-linear regression	STL	hand-crafted features	*rl*	No	Yes	6.10	4.65	4.32
[77]	2021	MIMIC-III	100 subject records, over 10 h data	ICU patients	PPG, ABP	hybrid neural network	MTL	raw signal	*rl*	No	No Yes	16.3 3.52	8.46 2.20	– –
[18]	2019	MIMIC-II	1912 subject records	ICU patients	PPG, ECG, ABP	neural network	MTL	raw signal	*rl*	No	No Yes	9.30 5.32	5.12 3.38	– –
[29][c]	2020	privately collected	11 subject records	Unknown	PPG, ABP	domain adversarial neural network	MTL	raw signal	*rl*	Yes	No Yes	7.46 6.79	4.68 4.48	– –
The proposed	–	MIMIC II MIMIC III	2076 records, 41520 samples 500 subjects, 500000 samples	ICU patients	PPG, ABP	neural network	MTL	raw signal	*rl*	–	No	5.88 14.54	3.35 6.45	3.49 7.58

[a]According to whether correlation between tasks is considered or not, the related work can be categorized into single-task learning (STL) based and multitask learning (MTL) based.

[b]*sl*: Denotes training, validation and test sets were split at the final aggregated sample level; *rl*: denotes training, validation and test sets were split at the subject record level; *LOSO*: denotes leave one subject record out, which can be seen as a special case of *rl*.

[c]In these works, only the result of RMSE is provided.

the model. Besides, the inclusion of demographic features (e.g. age, gender, etc.) that reflecting individual differences when training model can improve the prediction accuracy [15,75]; (4) a careful check is necessary to ensure the distribution between training, validation and test sets is consistent when splitting dataset, which is also the characteristic of the I.I.D principle. The problem is that under the premise of using '*rl*' splitting strategy, when the total number of subject records is too small, this is difficult to guarantee. Under this situation, based on domain adaption theory [80], the target error (i.e. test error) is upper bounded by the source error (i.e. training error) plus the difference (measured by H-divergence, etc.) between the distributions derived from the training and the test sets. In other words, the huge difference in the distribution between the training set and the test set is a non negligible factor affecting the test performance. That is why the results in studies [12,28,77] is exceptionally poor under the premise of with no calibration procedure, although the total number of training samples is large enough. Study [79] is an interesting work, although only 2 out of 28 records are used to build the multi-linear regression model, the reported result is well because the two carefully chosen training records contain sufficient variations in both SBP and DBP values, which enables the trained model with relatively good generalization ability to other records; (5) there is a lack of standardized preprocessing procedures for processing large-scale BP database such as MIMIC, which is objectively affect the fair comparison of related studies and then hinder the development of this area. For example, when comparing two methods such as studies [7,28], it is difficult to tell from which side the improvement in prediction results comes, different signal filtering, data cleaning, splitting strategies used or method/model improvements, or some of them, etc. It is encouraging that we have seen some efforts [81]; (6) building a general (no calibration) BP predictive model with good generalization ability on large/complex database is still challenging. In data-driven methods, data itself actually plays an critically important role in model building and model evaluation. For example, the dataset used in study [12] consists of four heterogeneous data sources collected by different sensors, and the effectiveness of the method is justified by extensive experiment under different evaluation strategies, although the result is relatively poor. From this point of view, purely pursuing the seemingly good results while ignoring data itself (BP range, BP distribution and measurement conditions, etc.) and the fairness of the evaluation process does not make much sense.

6.12. External validation

In order to further test the performance of the method/model on a larger dataset, we selected the MIMIC III dataset [70] for external validation. Specifically, the version published at the Kaggle website (https://www.kaggle.com/datasets/sirrabbit/ppg-dataset) was used, and a total of 500 000 sample pairs from 500 subject records were obtained for experiments. Each PPG signal segment is filtered using a 4th order Butterworth band-pass filter with cutoff frequencies of 0.5 Hz and 7 Hz, the upper limit of the cutoff frequency is empirically set smaller herein to better remove high-frequency noise. Then, the filtered segment is normalized using a ZScore normalizer. We designed two validation protocols: (a) direct validation without fine-tuning the model, this is used to evaluate the generalization performance of the model on external databases; (b) a model is trained from scratch and is evaluated on the MIMIC III database using the proposed TFNet-MTD^{2}L, this is used to further validate the proposed method. The parameter settings are the same as described in Section 6.2 except that the Batchsize is set to 64 for protocol (b). The experiments were performed on a Windows platform equipped with an RTX 3090 GPU. Note that for both of the two experiments, there are no any calibration/personalization techniques used like in similar studies [7,25,77,79,82,83].

Table 11 presents the results. It can be seen that a significant performance degradation occurs when the model trained with a small dataset of UCL is directly validated on the MIMIC III database, indicating that it is still difficult to train a generic BP predictor that performs well on various databases. We believe that this is mainly due to the following two reasons: First, there are differences between the data sets. The overall age distribution, sex ratio, and BP distribution of all individuals included in different datasets are different, and these characteristics of the data have a significant impact on the training as well as the evaluation of the model. Second, the complexity of the MIMIC III dataset to be tested is far greater than that of the dataset used for training models. Generally, the more individuals included in the dataset, the more diverse and complex the BP dynamics, which is more obvious in the dataset collected from patients. Note that the number of samples to be tested is more than 20 times the sample size of the previously used dataset for training BP predictor.

In addition, we have additionally trained a new BP predictor on the MIMIC III database (60% samples were used for training) using the proposed TFNet-MTD^{2}L method. The prediction results show a significant improvement compared with the results of protocol (a), since the first reason in the above analysis has been ruled out. Among the fewer relevant papers [25,77] that used the MIMIC III database for experiments and did not use any calibration procedures, we have achieved relatively good results. For example, Slapnicar et al. [25] used a total of 510 subjects from the MIMIC III database for experiments (to the best of our knowledge, the one that uses the most number of subjects). Compared with their results, our prediction accuracy for SBP,DBP improved by 5.65%, 47.90%, respectively.

However, it must be said that the more complex the data, the more difficult it is to achieve good results. Specifically, in addition to the bias in the overall statistics (we refer to BP range covered, BP distribution, etc.) of the data set, due to differences in demographics (gender, age, etc.), health status (diseases such as obesity, cardiovascular disease, etc.) and lifestyle habits (alcohol abuse, daily exercise), BP levels and patterns of variation vary greatly among individuals. In particular, for patients, medical intervention, such as drugs and surgery, is also one of the important external factors affecting BP changes [40]. Therefore, calibration/personalization or fine-tuning techniques are usually used to further fine tune the model to improve prediction accuracy. We noticed that, in addition to the usually used calibration/personalization techniques [7,25,77], some authors [79,82,83] try to predict the amount of change in BP rather than the ground-truth BP, and the predicted BP change plus the base BP is the final predicted BP, which is essentially equivalent to an individual-by-individual calibration.

7. Discussion and conclusion

In this study, we model cuff-less BP prediction as a multitask label distribution learning question for the first time. Specifically, BP value is firstly converted to label distribution in the label space. Then, a MTL network that capable of learning from different modalities of a signal is designed and the network is jointly trained based on the proposed adaptive multitask loss function to directly learn the mapping between the raw PPG signal and the label distribution of each task in an end-to-end manner. In addition, task-specific attention module is introduced in each task network to learn the different importance of each learned features for different prediction tasks, under the constraint of hard parameter sharing mode for MTL. Compared with classification-based methods that can only estimate the BP level and regression-based methods that directly predict BP value, the proposed distribution learning-based method can predicts BP value accompanied with predictive confidence, and the *normalize target* technique usually used in regression modeling is no longer needed.

Extensive ablation experiments justified the effectiveness of the proposed method. Further comparison with several representative methods/systems shows that the proposed model achieves competitive results, while using only PPG signals and does not require any calibration or *normalize target* procedures. We attribute this to three points: (1) based on the novel distribution learning paradigm, the information

Table 11
External validation results of the proposed method or the model trained using the Cuff-Less Blood Pressure Estimation Data Set from UCL, on the MIMIC III database.

Experimental protocol	Method	Task	Metrics (unit: mmHg)					
			MAE↓	MAPE↓	ME↓	STD↓	R^2 ↑	Spearman↑
(a)	TFNet-MTD^{2}L, no fine-tuning	SBP	17.9222	0.1380	0.4311	21.1192	−1.8135	0.0150
		DBP	9.2152	0.1287	−2.5693	11.1846	−2.4576	0.0351
		MBP	9.9272	0.1098	−1.7512	11.8188	−1.6467	0.0185
(b)	TFNet-MTD^{2}L, 6:2:2	SBP	14.5412	0.1141	1.6507	17.7342	0.6464	0.2976
		DBP	6.4543	0.0887	0.4432	8.1921	0.3721	0.2562
		MBP	7.5812	0.0837	0.8334	9.3665	0.3871	0.2936

of neighboring samples is modeled through soft target, which can be seen as sample level information fusion; (2) a MTL network with hard parameter sharing mode is developed for predicting SBP, DBP and MBP in parallel, and the difference in the most informative features accounting for different tasks is overcome by task-specific attention module, which can be seen as task level information fusion; (3) informative features related to BP are delved by learning and fusing information from different modalities of PPG signal, which can be seen as feature learning level information fusion.

Practical Implications. As a preliminary attempt, we propose a new paradigm for modeling BP prediction tasks, which can be seen as a compromise between the widely used classification and regression modeling techniques in the relevant studies. This paradigm enables to naturally quantify the uncertainty of prediction, and seems to alleviate the imbalance problem in BP estimation to a certain extent. In addition, we hope that our work will stimulate a broader reflection on several factors highly relevant to the fairness and objectivity of the model evaluation process itself, such as the splitting strategy,etc.

Limitations and Future Work. However, there are still some issues that need to be further explored. First, the model performs poorly on samples with very low/high BP. Note that this is a common phenomenon in this area [7,76]. As mentioned earlier, although the distribution learning paradigm can alleviate the model predictive bias caused by imbalanced data sets to some extent compared to regression models, this is not the focus of our study. In fact, imbalanced phenomenon in regression scenario is an extremely important but ignored topic in this area [84], and we plan to explore this question further in the future by drawing on technologies such as cost sensitive learning in classification scenarios.

Second, we model different modalities of input signal through a multi-branched structure, which increases the storage overhead and decreases the inference speed [85]. It is necessary to explore more parameter-efficient architecture.

Third, although the data used in the experiments are comparable to other similar studies in terms of the amount of data and the complexity of the data, note that there are newer versions of the MIMIC series database such as MIMIC III [70] and MIMIC IV [86] are available. For data-driven approaches, in addition to the methodology, the data itself plays an extremely important role in the final performance of the model [87]. As for BP estimation, these characteristics of the dataset — the quality of the waveform data, the number of participants, the final sample size, the diversity of the data in terms of age, gender, BP range covered, and BP dynamics, are important for training a model with good generalization ability and for objective evaluation of the model. Therefore, it is meaningful to further validate the proposed method and test the trained model on these larger datasets.

Last, the results reported in almost all relevant studies so far are based on a single specific database. It is very challenging to train a general BP estimator with strong generalization ability that across databases, since there are some other factors such as deviation/bias of different datasets, more diversified BP dynamics, differences in data collection tools, measurement methods, and measurement conditions in different databases that must be considered in addition to the complexity within a single dataset.

CRediT authorship contribution statement

Keke Qin: Conceptualization, Methodology, Software, Writing – original draft. **Wu Huang:** Conceptualization, Writing – review & editing, Project supervision. **Tao Zhang:** Writing – review & editing, Project supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We used public dataset, the link to which is attached to the manuscript, which is available for free download.

Appendix. Description of the variants of the proposed method TFNet-MTD^{2}L

- *a*. TFNet-MTD^{2}L: The abbreviation of the proposed method (i.e. a TFNet module followed by three task networks based on the distribution learning head, and the input includes the original PPG signal and its derivatives by default).
- *b*. TFNet-STD^{2}L: For each task, a TFNet followed by only a single distribution learning-based task network.
- *c*. TFNet-MTDRL: The TFNet module followed by three task networks based on the regression learning head, and L_2 loss is used by default.
- *d*. TFNet-STDRL: For each task, the TFNet module followed by only a single task network based on the regression learning head, and L_2 loss is used by default.
- *g*. TNet-MTD^{2}L: The TFNet-MTD^{2}L method except that one branch—frequency domain module is excluded.
- *h*. FNet-MTD^{2}L: The TFNet-MTD^{2}L method except that one branch—time domain module is excluded.
- *i*. TFNet-MTD^{2}L, no mlo: The TFNet-MTD^{2}L method except that the multi-level output (MLO) module of the U^{2}Net in the time domain module is excluded.
- *j*. TFNet-MTD^{2}L, no mse: The TFNet-MTD^{2}L method except that the Multi-scale extraction (MSE) module of the U^{2}Net in the time domain module is excluded.
- *k*. TFNet-MTD^{2}L, TSA: The TFNet-MTD^{2}L method except that the task-specific attention (TPA) module is replaced with the task-shared attention (TSA) module.
- *l*. TFNet-MTD^{2}L, no AM: The TFNet-MTD^{2}L method except that the task-specific attention (TPA) module is excluded.
- *e*. TFNet-MTD^{2}L, x+x': The TFNet-MTD^{2}L method except that only the raw PPG signal and its 1st order differential signal are used as input.
- *f*. TFNet-MTD^{2}L, x: The TFNet-MTD^{2}L method except that only the raw PPG signal is used as input.

• a'. TFNet-MTD^{2}L$_{naive}$: The TFNet-MTD^{2}L method except that the final loss is calculated by simply averaging the losses of the three tasks, i.e. ${\delta_j^l}^2 = 1$ for $j \in \{1, 2, \ldots, K\}$, $l \in \{1, 2\}$.

• c'. TFNet-MTDRL$_{naive}$: The TFNet-MTDRL method except that the final loss is calculated by simply averaging the losses of the three tasks, i.e. ${\delta_j}^2 = 1$ for $j \in \{1, 2, \ldots, K\}$.

References

[1] T. Arakawa, Recent research and developing trends of wearable sensors for detecting blood pressure, Sensors 18 (9) (2018) 2772.

[2] R.C. Cozby, R.R. Adhami, Low-frequency Korotkoff signal analysis and application, IEEE Trans. Biomed. Eng. 40 (10) (1993) 1067–1070.

[3] K.-i. Yamakoshi, S. Tanaka, Standard algorithm of blood-pressure measurement by the oscillometric method, Med. Biol. Eng. Comput. 31 (1993) 204, http://dx.doi.org/10.1007/BF02446682.

[4] K.-I. Yamakoshi, H. Shimazu, T. Togawa, Indirect measurement of instantaneous arterial blood pressure in the human finger by the vascular unloading technique, IEEE Trans. Biomed. Eng. 27 (3) (1980) 150–155.

[5] L. Peter, N. Noury, M. Cerny, A review of methods for non-invasive and continuous blood pressure monitoring: Pulse transit time method is promising? IRBM 35 (5) (2014) 271–282.

[6] C. El-Hajj, P.A. Kyriacou, A review of machine learning techniques in photoplethysmography for the non-invasive cuff-less measurement of blood pressure, Biomed. Signal Process. Control 58 (2020) 101870.

[7] M. Kachuee, M.M. Kiani, H. Mohammadzade, M. Shabany, Cuffless blood pressure estimation algorithms for continuous health-care monitoring, IEEE Trans. Biomed. Eng. 64 (4) (2016) 859–869.

[8] M.S. Tanveer, M.K. Hasan, Cuffless blood pressure estimation from electrocardiogram and photoplethysmogram using waveform based ANN-LSTM network, Biomed. Signal Proces. Control 51 (2019) 382–392.

[9] F. Riaz, M.A. Azad, J. Arshad, M. Imran, A. Hassan, S. Rehman, Pervasive blood pressure monitoring using photoplethysmogram (PPG) sensor, Future Gener. Comput. Syst. 98 (2019) 120–130.

[10] H. Tjahjadi, K. Ramli, H. Murfi, Noninvasive classification of blood pressure based on photoplethysmography signals using bidirectional long short-term memory and time-frequency analysis, IEEE Access PP (2020) 1, http://dx.doi.org/10.1109/ACCESS.2020.2968967.

[11] A. El Attaoui, S. Largo, A. Jilbab, A. Bourouhou, Wireless medical sensor network for blood pressure monitoring based on machine learning for real-time data classification, J. Amb. Intel. Hum. Comp. 12 (9) (2021) 8777–8792.

[12] M. Simjanoska, S. Kochev, J. Tanevski, A. Madevska Bogdanova, G. Papa, T. Eftimov, Multi-level information fusion for learning a blood pressure predictive model using sensor data, Inform. Fusion 58 (2020) 24–39.

[13] X. Fan, H. Wang, F. Xu, Y. Zhao, K.-L. Tsui, Homecare-oriented intelligent long-term monitoring of blood pressure using electrocardiogram signals, IEEE Trans. Ind. Inform. 16 (11) (2019) 7150–7158.

[14] F. Miao, Z.-D. Liu, J.-K. Liu, B. Wen, Q.-Y. He, Y. Li, Multi-sensor fusion approach for cuff-less blood pressure measurement, IEEE J. Biomed. Health 24 (1) (2019) 79–91.

[15] E. Monte-Moreno, Non-invasive estimate of blood glucose and blood pressure from a photoplethysmograph by means of machine learning techniques, Artif. Intell. Med. 53 (2) (2011) 127–138.

[16] X. Li, S. Wu, L. Wang, Blood pressure prediction via recurrent models with contextual layer, in: WWW, 2017, pp. 685–693.

[17] P. Su, X.-R. Ding, Y.-T. Zhang, J. Liu, F. Miao, N. Zhao, Long-term blood pressure prediction with deep recurrent neural networks, in: Int. Conf. BHI, IEEE, 2018, pp. 323–328.

[18] S. Baek, J. Jang, S. Yoon, End-to-end blood pressure prediction via fully convolutional networks, IEEE Access 7 (2019) 185458–185468.

[19] O. Schlesinger, N. Vigderhouse, Y. Moshe, D. Eytan, Estimation and tracking of blood pressure using routinely acquired photoplethysmographic signals and deep neural networks, Crit. Care Expl. 2 (4) (2020).

[20] G. Thambiraj, U. Gandhi, U. Mangalanathan, V.J.M. Jose, M. Anand, Investigation on the effect of Womersley number, ECG and PPG features for cuff less blood pressure estimation using machine learning, Biomed. Signal Proces. Control 60 (2020) 101942.

[21] S.S. Mousavi, M. Firouzmand, M. Charmi, M. Hemmati, M. Moghadam, Y. Ghorbani, Blood pressure estimation from appropriate and inappropriate PPG signals using a whole-based method, Biomed. Signal Proces. Control 47 (2019) 196–206.

[22] H. Eom, D. Lee, S. Han, Y.S. Hariyani, Y. Lim, I. Sohn, K. Park, C. Park, End-to-end deep learning architecture for continuous blood pressure estimation using attention mechanism, Sensors 20 (8) (2020) 2338.

[23] M.W.K. Fong, E. Ng, K.E.Z. Jian, T.J. Hong, SVR ensemble-based continuous blood pressure prediction using multi-channel photoplethysmogram, Comput. Biol. Med. 113 (2019) 103392.

[24] S. Lee, A. Ahmad, G. Jeon, Combining bootstrap aggregation with support vector regression for small blood pressure measurement, J. Med. Syst. 42 (4) (2018) 1–7.

[25] G. Slapničar, N. Mlakar, M. Luštrek, Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network, Sensors 19 (15) (2019) 3420.

[26] M. Simjanoska, M. Gjoreski, M. Gams, A. Madevska Bogdanova, Non-invasive blood pressure estimation from ECG using machine learning techniques, Sensors 18 (4) (2018) 1160.

[27] D. Fujita, A. Suzuki, K. Ryu, PPG-based systolic blood pressure estimation method using PLS and level-crossing feature, Appl. Sci. 9 (2) (2019) 304.

[28] S.S.N. Bose, A. Kandaswamy, Sparse representation of photoplethysmogram using K-SVD for cuffless estimation of arterial blood pressure, in: ICACCS, IEEE, 2017, pp. 1–5.

[29] L. Zhang, N.C. Hurley, B. Ibrahim, E. Spatz, H.M. Krumholz, R. Jafari, M.J. Bobak, Developing personalized models of blood pressure estimation from wearable sensors data using minimally-trained domain adversarial neural networks, in: MLHC, PMLR, 2020, pp. 97–120.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: CVPR, 2016, pp. 2818–2826.

[31] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, Comput. Sci. 14 (7) (2015) 38–39.

[32] X. Geng, Label distribution learning, IEEE Trans. Knowl. Data Eng. 28 (7) (2016) 1734–1748.

[33] B.-B. Gao, C. Xing, C.-W. Xie, J. Wu, X. Geng, Deep label distribution learning with label ambiguity, IEEE Trans. Image Process. 26 (6) (2017) 2825–2838.

[34] H. Pan, H. Han, S. Shan, X. Chen, Mean-variance loss for deep age estimation from a face, in: CVPR, 2018, pp. 5285–5294.

[35] B.-B. Gao, H.-Y. Zhou, J. Wu, X. Geng, Age estimation using expectation of label distribution learning, in: IJCAI, 2018, pp. 712–718.

[36] M. Forouzanfar, H.R. Dajani, V.Z. Groza, M. Bolic, S. Rajan, Feature-based neural network approach for oscillometric blood pressure estimation, IEEE Trans. Instrum. Meas. 60 (8) (2011) 2786–2796.

[37] P.-H. Chiang, S. Dey, Personalized effect of health behavior on blood pressure: Machine learning based prediction and recommendation, in: IEEE Int. Conf. Healthcom., IEEE, 2018, pp. 1–6.

[38] M. Kachuee, M.M. Kiani, H. Mohammadzade, M. Shabany, Cuff-less high-accuracy calibration-free blood pressure estimation using pulse transit time, in: ISCAS, IEEE, 2015, pp. 1006–1009.

[39] S. Ahmad, S. Chen, K. Soueidan, I. Batkin, M. Bolic, H. Dajani, V. Groza, Electrocardiogram-assisted blood pressure estimation, IEEE Trans. Biomed. Eng. 59 (3) (2012) 608–618.

[40] X. Xing, M. Sun, Optical blood pressure estimation with photoplethysmography and FFT-based neural networks, Biomed. Opt. Express 7 (8) (2016) 3007–3020.

[41] G. Thambiraj, U. Gandhi, V. Devanand, U. Mangalanathan, Noninvasive cuffless blood pressure estimation using pulse transit time, Womersley number, and photoplethysmogram intensity ratio, Physiol. Meas. 40 (7) (2019) 075001.

[42] P.-H. Chiang, S. Dey, Offline and online learning techniques for personalized blood pressure prediction and health behavior recommendations, IEEE Access 7 (2019) 130854–130864.

[43] Q. Yousef, M. Reaz, M.A.M. Ali, The analysis of PPG morphology: Investigating the effects of aging on arterial compliance, Meas. Sci. Rev. 12 (6) (2012) 266–271.

[44] J. Cheng, Y. Xu, R. Song, Y. Liu, C. Li, X. Chen, Prediction of arterial blood pressure waveforms from photoplethysmogram signals via fully convolutional neural networks, Comput. Biol. Med. 138 (2021) 104877.

[45] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, F. Ren, Learning in the frequency domain, in: CVPR, 2020, pp. 1740–1749.

[46] S. Yao, A. Piao, W. Jiang, Y. Zhao, H. Shao, S. Liu, D. Liu, J. Li, T. Wang, S. Hu, et al., STFNets: Learning sensing signals from the time-frequency perspective with short-time Fourier neural networks, in: WWW, 2019, pp. 2192–2202.

[47] F. Miao, N. Fu, Y.T. Zhang, X.R. Ding, X. Hong, Q. He, Y. Li, A novel continuous blood pressure estimation approach based on data mining techniques, IEEE J. Biomed. Health (2017) 1.

[48] B. Ibrahim, R. Jafari, Cuffless blood pressure monitoring from an array of wrist bio-impedance sensors using subject-specific regression models: Proof of concept, IEEE Trans. Biomed. Circ. Syst. 13 (6) (2019) 1723–1735.

[49] X. Ding, B.P. Yan, Y.-T. Zhang, J. Liu, P. Su, N. Zhao, Feature exploration for knowledge-guided and data-driven approach based cuffless blood pressure measurement, 2019, arXiv:arXiv:1908.10245.

[50] C.-M. Wu, C.Y. Chuang, Y.-J. Chen, S.-C. Chen, A new estimate technology of non-invasive continuous blood pressure measurement based on electrocardiograph, Adv. Mech. Eng. 8 (6) (2016) 1687814016653689.

[51] M. Brown, The seventh report of the joint national committee on prevention, detection, evaluation, and treatment of high blood pressure. The JNC 7 report, Evid.-Based Eye Care 4 (2003) 179–181, http://dx.doi.org/10.1097/00132578-200307000-00027.

[52] A. Argha, J. Wu, S.W. Su, B.G. Celler, Blood pressure estimation from beat-by-beat time-domain features of oscillometric waveforms using deep-neural-network classification models, IEEE Access 7 (2019) 113427–113439.

[53] B.G. Celler, P.N. Le, A. Argha, E. Ambikairajah, GMM-HMM-based blood pressure estimation using time-domain features, IEEE Trans. Instrum. Meas. 69 (6) (2019) 3631–3641.
[54] J. Wang, X. Geng, Classification with label distribution learning., in: IJCAI, 2019, pp. 3712–3718.
[55] Y. Zhou, H. Xue, X. Geng, Emotion distribution recognition from facial expressions, in: ACM Multimedia, 2015, pp. 1247–1250.
[56] S. Ruder, An overview of multi-task learning in deep neural networks, 2017, arXiv:arXiv:1706.05098.
[57] A. Kendall, Y. Gal, What uncertainties do we need in Bayesian deep learning for computer vision? 2017, arXiv:arXiv:1703.04977.
[58] S. Liu, Y. Liang, A. Gitter, Loss-balanced task weighting to reduce negative transfer in multi-task learning, in: AAAI, Vol. 33, 2019, pp. 9977–9978.
[59] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, in: ICML, PMLR, 2018, pp. 794–803.
[60] A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: CVPR, 2018, pp. 7482–7491.
[61] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. Zaï ane, M. Jagersand, U2-Net: Going deeper with nested U-structure for salient object detection, Pattern Recognit. 106 (2020) 107404, http://dx.doi.org/10.1016/j.patcog.2020.107404.
[62] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.
[63] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: ECCV, Springer, 2016, pp. 630–645.
[64] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, 2015, arXiv:arXiv:1511.07122.
[65] M. Lin, Q. Chen, S. Yan, Network in network, 2013, arXiv:arXiv:1312.4400.
[66] J. Hu, L. Shen, G. Sun, S. Albanie, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. PP (2017) http://dx.doi.org/10.1109/TPAMI.2019.2913372.
[67] T. Evgeniou, C.A. Micchelli, M. Pontil, J. Shawe-Taylor, Learning multiple tasks with kernel methods, J. Mach. Learn. Res. 6 (4) (2005).
[68] Q. Cai, Y. Pan, Y. Wang, J. Liu, T. Yao, T. Mei, Learning a unified sample weighting network for object detection, in: CVPR, 2020, pp. 14173–14182.
[69] M. Saeed, M. Villarroel, A.T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T.H. Kyaw, B. Moody, R.G. Mark, Multiparameter intelligent monitoring in intensive care II (MIMIC-II): A public-access intensive care unit database, Crit. Care Med. 39 (5) (2011) 952.
[70] A. Johnson, T. Pollard, L. Shen, L.-w. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Celi, R. Mark, MIMIC-III, a freely accessible critical care database, Sci. Data 3 (2016) 160035, http://dx.doi.org/10.1038/sdata.2016.35.
[71] E. O'Brien, J. Petrie, W. Littler, M. de Swiet, P.L. Padfield, K. O'Malley, M. Jamieson, D. Altman, M. Bland, N. Atkins, The British hypertension society protocol for the evaluation of automated and semi-automated blood pressure measuring devices with special reference to ambulatory systems, J. Hypertens. 8 (7) (1990) 607–619.
[72] A. for the Advancement of Medical Instrumentation, et al., American national standards for electronic or automated sphygmomanometers, 1987, ANSI/AAMI SP 10-1987.
[73] G. Slapničar, M. Luštrek, M. Marinko, Continuous blood pressure estimation from PPG signal, Informatica 42 (1) (2018).
[74] S. Baker, W. Xiang, I. Atkinson, A hybrid neural network for continuous and non-invasive estimation of blood pressure from raw electrocardiogram and photoplethysmogram waveforms, Comput. Meth. Prog. Bio. 207 (2021) 106191.
[75] A. Attarpour, A. Mahnam, A. Aminitabar, H. Samani, Cuff-less continuous measurement of blood pressure using wrist and fingertip photo-plethysmograms: Evaluation and feature analysis, Biomed. Signal Process. Control 49 (2019) 212–220.
[76] F. Schrumpf, P. Frenzel, C. Aust, G. Osterhoff, M. Fuchs, Assessment of deep learning based blood pressure prediction from PPG and rPPG signals, in: CVPR Workshop, 2021, pp. 3820–3830.
[77] J.J. Leitner, P.-H. Chiang, S. Dey, Personalized blood pressure estimation using photoplethysmography: A transfer learning approach, IEEE J. Biomed. Health (2021).
[78] V. Svetnik, A. Liaw, C. Tong, J.C. Culberson, R.P. Sheridan, B.P. Feuston, Random forest: A classification and regression tool for compound classification and QSAR modeling, J. Chem. Inf. Comp. Sci. 43 (6) (2003) 1947–1958.
[79] S. Haddad, A. Boukhayma, A. Caizzone, Continuous PPG-based blood pressure monitoring using multi-linear regression, 2020, arXiv:arXiv:2011.02231.
[80] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J.W. Vaughan, A theory of learning from different domains, Mach. Learn. 79 (1) (2010) 151–175.
[81] A.E. Johnson, D.J. Stone, L.A. Celi, T.J. Pollard, The MIMIC code repository: Enabling reproducibility in critical care research, J. Am. Med. Inform. Assoc. 25 (1) (2018) 32–39.
[82] S. Yang, J. Sohn, S. Lee, J. Lee, H.C. Kim, Estimation and validation of arterial blood pressure using photoplethysmogram morphology features in conjunction with pulse arrival time in large open databases, IEEE J. Biomed. Health 25 (4) (2020) 1018–1030.
[83] F. Miao, B. Wen, Z. Hu, G. Fortino, X.-P. Wang, Z. Liu, M. Tang, Y. Li, Continuous blood pressure measurement from one-channel electrocardiogram signal using deep-learning techniques, Artif. Intell. Med. 108 (2020) 101919, http://dx.doi.org/10.1016/j.artmed.2020.101919.
[84] B. Krawczyk, Learning from imbalanced data: Open challenges and future directions, Prog. Artif. Intell. 5 (4) (2016) 221–232.
[85] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, RepVGG: Making VGG-style convnets great again, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733–13742.
[86] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L.A. Celi, R. Mark, MIMIC-IV, 2020, PhysioNet. Available Online at: https://physionet.org/content/mimiciv/1.0/. (Accessed 23 August 2021).
[87] W. Liang, G.A. Tadesse, D. Ho, L. Fei-Fei, M. Zaharia, C. Zhang, J. Zou, Advances, challenges and opportunities in creating data for trustworthy AI, Nat. Mach. Intell. 4 (8) (2022) 669–677.