

Received August 7, 2019, accepted August 16, 2019, date of publication August 20, 2019, date of current version September 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936516

A Parallel GRU Recurrent Network Model and Its Application to Multi-Channel Time-Varying Signal Classification

SHAOHUA XU¹, JINGJING LI¹, KUN LIU¹, AND LU WU^{1,2}

¹College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

²Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250101, China

Corresponding author: Kun Liu (liukun9026@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC1406203 and in part by the SDUST Research Fund.

ABSTRACT This study presents a modified recurrent neural network (RNN) model designed as a parallel computing structure for serial information processing. The result is a novel parallel recurrent neural network (P-RNN), proposed for application to time-varying signal classification. The network uses gated recurrent units (GRUs) for basic information processing and consists of a multi-channel time series signal input layer, parallel processing structure units, a signal feature fusion layer, and a softmax classifier. The P-RNN expands the existing RNN serial processing mode for multi-channel time-varying signals into parallel mode and realizes the embedding of multi-channel signal structure features. In these parallel processing units, the input signal for each channel corresponds to a GRU recurrent network. Feature extraction and attribute association of single-channel signals were performed to achieve parallel processing of all-channel signals. In the feature fusion layer, feature vectors from each channel signal were integrated to generate a comprehensive feature matrix. On this basis, the softmax function was used as a classifier for multi-channel signals. With this mechanism, the P-RNN model achieved independent feature extraction of single-channel signals, characteristic fusion of each channel signal, and signal classification based on an integrated feature matrix. This approach maintained characteristic combination relationships that improved serial modes for existing RNN multi-channel signal processing, reduced the loss of structural feature information, and improved the representation ability of combined feature in local time region and the efficiency of the algorithm. In this paper, the properties of the proposed P-RNN are analyzed and a comprehensive learning algorithm is developed. Seven disease classification types commonly diagnosed using 12-lead ECG signals were used to validate the technique experimentally. Results showed the computational efficiency improved by a factor of 11.519, compared with existing RNN serial processing times, producing a correct recognition rate of 95.976%. In particular, the resolution of signal samples with similar distribution characteristics improved significantly, which demonstrates the effectiveness of the proposed technique.

INDEX TERMS Time-varying signal classification, recurrent neural network, parallel structures, feature fusion.

I. INTRODUCTION

Pattern classification problems using multi-channel nonlinear time-varying signals are common in a variety of computer vision applications [1]. Such signals are multi-component waveforms with a frequency and amplitude that vary with time, exhibiting both non-linearity and non-stationarity [2].

The associate editor coordinating the review of this article and approving it for publication was Md. Kamrul Hasan.

This combination of attributes and structural features complicates the corresponding classification process, making this a critical issue in the fields of signal analysis and machine learning [3]. Multi-channel signals must be considered not only a characteristic distribution of single-channel data, but also as global structural information between signals in each channel and a combination relationship for local regional features [4]. Time-varying samples are also subject to random disturbances, noise, and coupling between signals. As a

result, such data tend to show multi-peak characteristics, as well as scaling, drift, and noise artifacts that are highly dependent on the associated time scale [5]. Process characteristics for multiple signal combinations in multi-variable systems exhibit a particularly high degree of complexity [6]. As such, most existing algorithms consider time-varying signals to be both short-term stationary and linear, focusing instead on information contained in the time and frequency domains [7]. They also extract digital features describing signal distribution patterns for further analysis. As a result, any information carried by these signals that changes over time is partially ignored or lost, which affects the objectivity and accuracy of time-varying information extraction [8]. This omission can have a detrimental effect in practical applications such as high-frequency sensor signal acquisition, massive stream data processing, and multi-source time-varying signal fusion. These fields are currently experiencing rapid growth that continues to increase efficiency requirements and introduces new challenges for the analysis and modeling of multi-channel time-varying signals.

Artificial neural networks are a common tool that have proven to be effective for signal processing. With the development of deep learning, modified neural network models have been proposed for time-varying signal analysis, including deep convolutional neural networks [9], deep auto-encoder neural networks [10], deep recursive networks [11], deep recurrent networks, and Markov chains [12]. Deep recurrent networks have exhibited particularly good adaptability for the classification and analysis of time-varying signals [13] and they have successfully been applied to speech recognition [14], natural language processing [15], and document analysis [16]. Recurrent networks are based on initial work done by Rumelhart in 1986 [17]. Elman (1990) proposed simple RNNs (SRNs), which are 3-layer networks with context units added to the hidden layer. SRNs are also capable of completing sequence prediction tasks that standard MLPs cannot solve [18]. Schuster (1997) extended regular RNNs and introduced bidirectional recurrent neural networks (BRNNs), which can be trained without the limitation of providing input information to a preset future frame [19]. Hochreiter (1997) introduced a novel, efficient, gradient-based method called long short-term memory (LSTM), a deep learning system that avoids the vanishing gradient problem. LSTM performs well even when presented with long delays between significant events, allowing it to process signals with mixed low- and high-frequency components [20]. Jaeger (2004) presented a special type of RNN method, for learning nonlinear systems, called echo state networks (ESNs). These large-scale randomly connected recurrent networks are used to replace the middle layer of classical neural networks in order to simplify the network training process [21]. Graves (2013) investigated deep recurrent neural networks, which combine multiple levels of representation and a flexible use of long-range context. The combination of deep, bidirectional, long short-term memory RNNs with end-to-end training and weighting noise

has produced state-of-the-art results for phoneme recognition using the TIMIT database [22]. Auli (2013) proposed a joint language and translation model based on a recurrent neural network, which predicted target words using an unbounded history of both source and target words [23]. Cho (2014) proposed gated recurrent unit (GRU) networks for neural machine translation. GRUs provide improvements over conventional RNNs [24]. Koutnik (2014) introduced a simple yet powerful modification to the standard RNN architecture, the clockwork RNN (CW-RNN), for sequence prediction and classification [25]. Li (2015) introduced a novel parallel recursive deep model (PRDM) used for predicting sentiment label distributions. The primary advantage of this technique is that it not only uses composition units, but also exploits information encoded among the structure of sentiment labels [26]. Hidasi (2016) introduced a number of parallel RNN architectures to model sessions based on the features (images and text) of clicked items [27]. Wang (2016) presented a novel parallel-fusion RNN-LSTM architecture by combining the advantages of a simple RNN and LSTM, which produced better results than a dominated algorithm and improved the operational efficiency [28]. Turchenko (2010) developed a parallel algorithm for batch pattern training of a recurrent neural network using a back-propagation algorithm and a general-purpose parallel computer [29]. In recent years, the design of parallel computing framework has attracted extensive attention in machine learning. Larhlimi (2018) proposed a GPU parallel Neural Hierarchical Multi Objective solver, which has strong scalability and adaptability in framework [30]. Pofap (2018) proposed three parallel strategies of meta-heuristic algorithm, which greatly improves the accuracy and efficiency of the solution [31].

Comprehensive analysis suggests that existing parallel RNN models are primarily intended for applications such as image and text analysis, sentiment label prediction, and speech recognition. These tasks typically require multiple RNNs to construct parallel architectures. Most of the RNN models used for time-varying signal analysis implement serial feature extraction and the association of temporal information based on feature vectors. In the classification of multi-channel signals, the RNN is processed in a serial mode in which each channel signals are sequentially connected. This delays the correlation and memory allocation of time-varying signal features, causing suppression of network property parameters determined by previous signals. Partial combination features and structural information between multi-channel signals are subsequently lost, increasing the required training time. If the serial processing mode of the existing RNN for multi-channel time-varying signals is extended to the parallel mode, and the structural information of the signal can be embedded in the mechanism, the computational efficiency and classification accuracy of the RNN can be improved.

This study proposes a parallel-structure recurrent neural network (P-RNN) model for multi-channel time-varying signal classification. In the presented architecture, serial signal

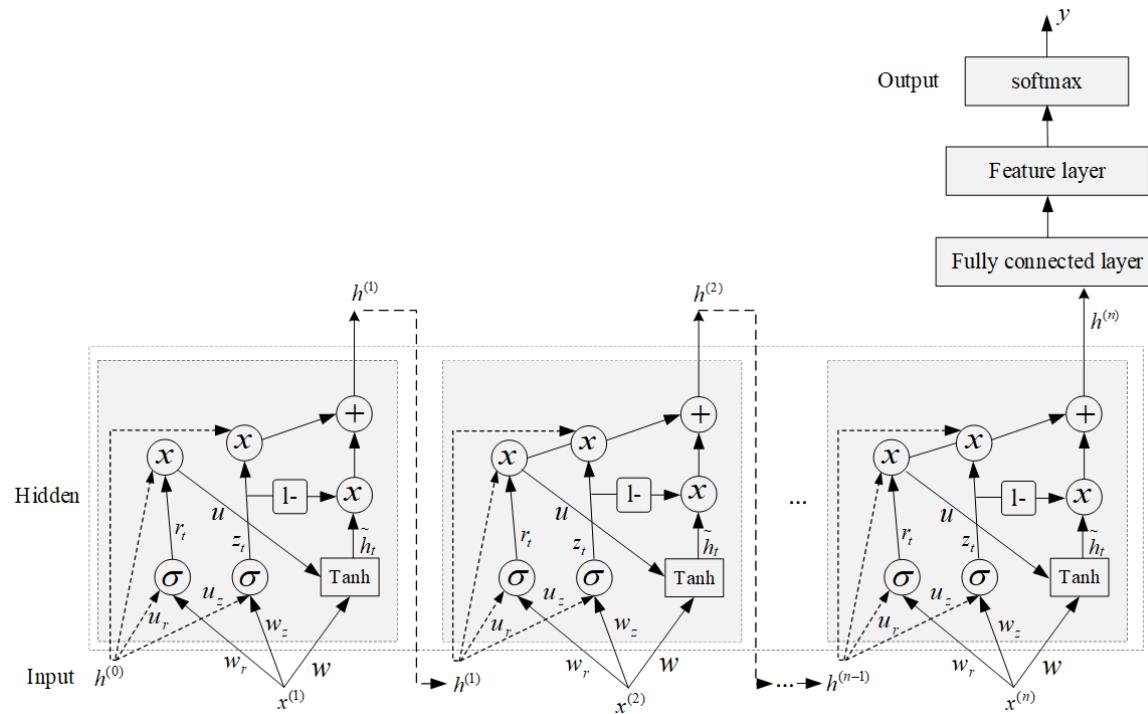


FIGURE 1. The recurrent network and information processing workflow based on GRU gated units.

processing models in existing RNNs are modified to form parallel calculation structures. Each channel input then corresponds to a deep GRU recurrent network, used for feature extraction and attribute association of single-channel signals. Feature vectors in each channel were sparsely fused to produce a comprehensive feature matrix of multi-channel signals, by establishing a fusion layer after the parallel structure units. A softmax classifier was then used to classify multi-channel signals based on this comprehensive matrix. The proposed P-RNN not only achieved independent feature extraction of single-channel signals, but also maintained structural information and combination relationships for multi-channel signals. This result improved the RNN's ability to identify global features in multi-channel signals and significantly improved algorithm efficiency.

Cardiovascular disease diagnosis using electrocardiograms (ECGs) is currently of significant interest in the field of artificial intelligence [32]. ECG signals exhibit unique characteristics including multi-peak, non-stationary, periodic, and high background-noise. These structural properties can complicate the combination of 12-lead signals. In clinical settings, some diseases may exhibit abnormal changes in only a few leads or local time interval, while others remain unaffected. Diagnosis is then dependent on the structural characteristics of multiple leads in the local region. In this study, classification of seven disease types was conducted using 12-lead ECG signals. The proposed model was validated by identifying local sensitive characteristics of single-lead signals and the combined characteristics of multiple leads.

The remainder of this paper is organized as follows. After presenting current challenges in time-varying signal classification and the status of artificial neural networks for signal processing, a novel P-RNN classification model and corresponding algorithm are proposed. Section 2 analyzes the P-RNN and its theoretical properties. In Section 3, a comprehensive learning algorithm is developed for the P-RNN. Classification experiments using ECG signals are presented in Section 4 and the results are discussed. The study is then summarized and the advantages and limitations of the proposed model are detailed in the conclusion.

II. THE PARALLEL-STRUCTURE RECURRENT NETWORK MODEL

This study introduces a parallel-structure deep recurrent neural network (P-RNN) with a GRU gating serving as the basic information recurrent unit. The structure and performance parameters of this GRU were used to extract and memorize distribution characteristics of time-varying signals. Each channel corresponded to a GRU recurrent network, which facilitated the feature extraction and characterization of single-channel signals. All processing was implemented in a parallel structure to form a parallel GRU recurrent network. Signal features in each channel were integrated by adding a fusion layer, thereby producing a comprehensive feature matrix. A softmax classifier, developed from this matrix, was then used to classify multi-channel time-varying signals. The complexity of such signals varies in practical applications, which introduces a variety of deep GRU recurrent network

structures to achieve full extraction and high-level characterization of complex signal features.

A. THE GRU RECURRENT NETWORK MODEL

Gated recurrent units (GRUs) include reset and update gates. In this structure, which embeds self-renewal states into hidden states after linear accumulation, lower GRU model parameters typically result in higher algorithm efficiency. The recurrent network structure and GRU-based information processing workflow are shown in Fig. 1. The forward propagation formula can be expressed as follows:

(1) The input layer includes input data given by $\{x^{(t)} \in \mathcal{R}^{n \times d}\}_{t=1}^T$, where $x^{(t)}$ represents the input at time t and T is the length of the time series.

(2) The GRU recurrent unit consists of reset and update gates. Reset gates ($r_t \in \mathcal{R}^{n \times d}$) can be expressed as:

$$r_t = \sigma \left(x^{(t)} W_{xr} + h^{(t-1)} U_{hr} + b_r \right). \quad (1)$$

Update gates ($z_t \in \mathcal{R}^{n \times d}$) are represented by:

$$z_t = \sigma \left(x^{(t)} W_{xz} + h^{(t-1)} W_{hz} + b_z \right). \quad (2)$$

In the above expressions, $W_{xy}, W_{xz} \in \mathcal{R}^{d \times h}$ and $U_{hr}, U_{hz} \in \mathcal{R}^{h \times h}$ are weighting parameters, $b_r, b_z \in \mathcal{R}^{1 \times h}$ are deviation parameters, and σ is a sigmoid function over the range [0, 1]. The range of r_t and z_t is also [0, 1]. The candidate hidden state at time step t is $\tilde{h}^{(t)} \in \mathcal{R}^{n \times h}$, expressed as:

$$\tilde{h}^{(t)} = \tanh \left(x^{(t)} W_{xh} + \left(r_t \odot h^{(t-1)} \right) U_{hh} + b_h \right). \quad (3)$$

Here, the terms $W_{xh} \in \mathcal{R}^{d \times h}$, $U_{hh} \in \mathcal{R}^{h \times h}$ are weighting parameters, $b_h \in \mathcal{R}^{1 \times h}$ is a deviation parameter, and \odot indicates multiplication of individual elements. The hidden state at time step t is $h^{(t)} \in \mathcal{R}^{n \times h}$, represented by:

$$h^{(t)} = z_t \odot h_i^{(t-1)} + (1 - z_t) \odot \tilde{h}^{(t)}. \quad (4)$$

Here, z_t represents the update gate at the current time step, $h^{(t-1)}$ indicates the hidden state at the previous time step, and $\tilde{h}^{(t)}$ represents a candidate hidden state at the current time.

(3) The GRU recurrent network includes hidden layer states given by:

$$h_i^{(t)} = z_t \odot h_i^{(t-1)} + (1 - z_t) \odot \tilde{h}_i^{(t)}, \quad (5)$$

where z_t represents the update gate for the i^{th} layer at the current time step, $h_i^{(t-1)}$ indicates the hidden state of the i^{th} layer at the previous time step, and $\tilde{h}_i^{(t)}$ represents the candidate hidden state of the i^{th} layer at the current time. The output of the recurrent unit is:

$$y_i^{(t)} = \sigma(W_o \cdot h_i^{(t)}) + b_o, \quad (6)$$

where W_o is a weighting parameter and b_o is a deviation parameter. The output of the fully-connected layer is given by:

$$y_i^{(o)} = \text{relu}(W_{fc} y_i^{(t)}) + b_{fc}, \quad (7)$$

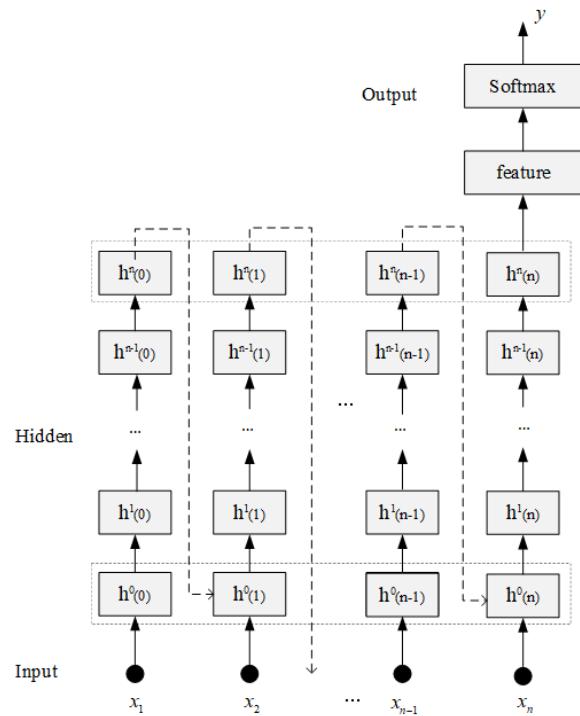


FIGURE 2. The structure of the deep recurrent network model.

where W_{fc} is a weight parameter in the fully-connected layer, b_{fc} is a deviation parameter, and relu is a linear activation function.

B. THE DEEP GRU RECURRENT NETWORK

Due to the complexity of characteristic distributions for time-varying signals, several GRU recurrent networks were stacked to form a deep recurrent neural network model. The structure and information processing workflow for this proposed technique are shown in Fig. 2. The hidden layer in the deep recurrent network was stacked with multiple GRU recurrent networks, with temporal expansion in the horizontal direction and stack depth in the vertical direction. Time-varying signals were input from the input layer to the recurrent network. Inputs to the current layer were composed of outputs from the previous hidden layer and the current input signal. The information processing workflow for these deep recurrent neural networks can be summarized as follows:

(1) The signal input at time t consists of the input $h_j^{(t-1)}$ from the hidden layer at time $t - 1$ and the signal input $x^{(t)}$ at the current time. This process can be represented as: $h_j^{(t)} = Wx^{(t)} + Uh_j^{(t-1)}$, where W, U are weight matrices.

(2) Connections exist between hidden layer neurons, which reflects the expansion of time and ensures the continuity of time-varying signals ($t \in \{t_1, t_2, \dots, t_n\}$).

(3) Multiple GRU recurrent networks were stacked to form deep recurrent networks $h = \{h^{(1)}, h^{(2)}, \dots, h^{(n)}\}$. The output of the previous hidden layer was then used as the input for

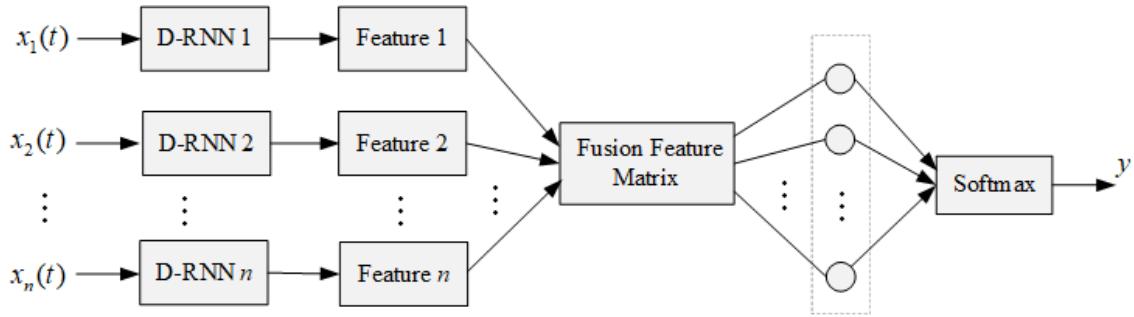


FIGURE 3. The parallel GRU recurrent network model.

the next hidden layer ($h_{in}^{(n)} = h_{out}^{(n-1)}$). This formed an n-layer recurrent network.

(4) Layer features, prior to the extraction of classifiers to form a feature matrix, were used as input to the softmax classifier. The multi-classification softmax function was defined as follows:

$$h_\theta = \begin{bmatrix} p(y^{(i)} = 1|x^{(i)}; \theta) \\ p(y^{(i)} = 2|x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k|x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}, \quad (8)$$

where $\theta_1, \theta_2, \dots, \theta_k \in \Re^{n+1}$ are model parameters and $\frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}}$ was used to normalize the probability distribution with the sum of all probabilities being 1. The term $p(y^{(i)} = j|x^{(i)}; \theta)$ represents the estimated probability value belonging to category j .

C. THE PARALLEL-STRUCTURE RNN MODEL

This P-RNN model was used to design serial processing modes for GRU deep recurrent networks. The time-varying signals discussed in Section 2.2 were organized into a parallel structure and each channel signal was processed separately by a GRU recurrent network. Parallel structures were formed by multiple GRU networks with single-channel signals. Multi-channel signal features were integrated in the subsequent feature fusion layer to generate a comprehensive feature matrix. A softmax function was then used as a classifier for multi-channel signals. The structure and information processing workflow for this P-RNN are shown in Fig. 3.

The first half of the network model adopts a parallel structure, with each channel input signal corresponding to a single deep GRU recurrent network. Each channel time-varying signal was then processed and the features were extracted by a recurrent network to generate a signal feature vector. In the second half of the model, a feature fusion layer was stacked after the parallel structure to achieve feature convergence of all channel signals, thereby producing a comprehensive feature matrix of multi-channel signals. A softmax classifier was linked after the feature fusion layer to achieve

multi-channel signal classification based on a comprehensive feature matrix.

This multi-channel softmax classifier is defined as follows:

$$h_\theta = \begin{bmatrix} p(y^{(i)} = 1|fea^{(i)}; \theta) \\ p(y^{(i)} = 2|fea^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k|fea^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T fea^{(i)}}} \begin{bmatrix} e^{\theta_1^T fea^{(i)}} \\ e^{\theta_2^T fea^{(i)}} \\ \vdots \\ e^{\theta_k^T fea^{(i)}} \end{bmatrix}, \quad (9)$$

where $\theta_1, \theta_2, \dots, \theta_k \in \Re^{n+1}$ are model parameters. The classification of multi-channel signals was achieved using maximum probability membership.

According to the structure of P-RNN model and the way of information processing, the time series signal corresponds to the GRU processing unit. The longer the time series, the more neurons in each layer. At the same time, the more complex the process characteristics of time-varying signals, the higher the requirement for feature extraction and memory ability of the network, and the more hidden layers of the network.

This P-RNN has the following advantages. (1) The model not only considers differences between the characteristics of single-channel signals for each sample, but also the combined characteristics of all channel samples. (2) The efficiency of the signal processing algorithm was significantly improved by the inclusion of a parallel computing structure. (3) Signal samples can be processed in various sampling time intervals.

III. THE P-RNN LEARNING ALGORITHM

The P-RNN learning process can be divided into the following steps. (1) Each channel signal corresponded to a GRU recurrent network, with the class label of the multi-channel signal serving as the desired output for iterative training. The exchange of parameters was not performed between parallel modules. (2) Feature vectors were then generated and feature integration of multi-channel signals was achieved in the feature fusion layer. (3) The softmax classifier was then trained and fine-tuning of P-RNN parameters was conducted using the BP algorithm.

Learning algorithm steps in the parallel GRU deep recurrent network were as follows:

(1) Construction of the n -channel signal training dataset was represented as $\{x^{(t)} \in \mathbb{R}^n, y^{(t)} \in \mathbb{R}^m\}_{t=1}^T$, where $x^{(t)}$ represents the input at time t . The length of the time-series was T and the output $y^{(t)}$ was related to the input before (and including) time t . The i^{th} channel signal sample set was given by: $\{x_i^{(t)} \in \mathbb{R}, y^{(t)} \in \mathbb{R}^m\}_{i=1}^T$, where $i = 1, 2, \dots, n$. Here, $x_i^{(t)}$ represents the input of the i^{th} channel signal at time t .

(2) The training of single-channel GRU recurrent networks was conducted by calculating reset gates, update gates, the output of candidate hidden states, and hidden states in each GRU unit of the recurrent network corresponding to each channel signal. This calculation is described in (1) – (4) above. A loss function was constructed using the negative logarithmic likelihood.

The loss function can be expressed as:

$$\begin{aligned} L\left(\{x^{(1)}, x^{(2)}, \dots, x^{(t)}\}, \{y^{(1)}, y^{(2)}, \dots, y^{(t)}\}\right) &= \sum_t L^{(t)} \\ &= -\sum_t \log p_{model}\left(y^{(t)} | \{x^{(1)}, x^{(2)}, \dots, x^{(t)}\}\right), \end{aligned} \quad (10)$$

where $L^{(t)}$ is the negative logarithmic likelihood of y given $x^{(1)}, x^{(2)}, \dots, x^{(t)}$. The term $p_{model}(y^{(t)} | \{x^{(1)}, x^{(2)}, \dots, x^{(t)}\})$ reads the item corresponding to $\{x^{(1)}, x^{(2)}, \dots, x^{(t)}\}$, and the output of the model is the probability of $y^{(t)}$. Here it is assumed that the data obey Gauss distribution.

The stochastic gradient descent algorithm was then used to develop an updating iterative formula for the recurrent neural network state:

$$\begin{aligned} \nabla_{h^{(t)}} L &= \left(\frac{\partial h^{(t+1)}}{\partial h^{(t)}}\right)^T (\nabla_{h^{(t+1)}} L) + \left(\frac{\partial o^{(t)}}{\partial h^{(t)}}\right)^T (\nabla_{o^{(t)}} L) \\ &= W^T (\nabla_{h^{(t+1)}} L) \operatorname{diag}\left(1 - (h^{(t+1)})^2\right) + V^T (\nabla_{o^{(t)}} L), \end{aligned} \quad (11)$$

where $\operatorname{diag}\left(1 - (h^{(t+1)})^2\right)$ represents a diagonal matrix containing elements $1 - (h_i^{(t+1)})^2$. W represents the connection weight parameter matrix from the input layer to the hidden layer, and V represents the connection weight parameter matrix from the hidden layer to the output layer.

Iterative calculation formulas for other network parameters are as follows:

$$\nabla_c L = \sum_t \left(\frac{\partial o^{(t)}}{\partial c}\right)^T \nabla_{o^{(t)}} L = \sum_t \nabla_{o^{(t)}} L, \quad (12)$$

$$\nabla_b L = \sum_t \left(\frac{\partial h^{(t)}}{\partial b^{(t)}}\right)^T \nabla_{h^{(t)}} L = \sum_t \operatorname{diag}\left(1 - (h^{(t)})^2\right) \nabla_{h^{(t)}} L, \quad (13)$$

$$\nabla_V L = \sum_t \sum_i \left(\frac{\partial L}{\partial o_i^{(t)}}\right)^T \nabla_{V o_i^{(t)}} = \sum_t (\nabla_{o^{(t)}} L) h^{(t)T}, \quad (14)$$

$$\begin{aligned} \nabla_W L &= \sum_t \sum_i \left(\frac{\partial L}{\partial h_i^{(t)}}\right) \nabla_{W^{(t)}} h_i^{(t)} \\ &= \sum_t \operatorname{diag}\left(1 - (h^{(t)})^2\right) (\nabla_{h^{(t)}} L) h^{(t-1)T}, \end{aligned} \quad (15)$$

$$\begin{aligned} \nabla_U L &= \sum_t \left(\frac{\partial L}{\partial h_i^{(t)}}\right)^T \nabla_{U^{(t)}} h_i^{(t)} \\ &= \sum_t \operatorname{diag}\left(1 - (h^{(t)})^2\right) (\nabla_{h^{(t)}} L) x^{(t)T}, \end{aligned} \quad (16)$$

In the above expressions, c represents the bias vector of linear transform from $h^{(t)}$ to $o^{(t)}$, and b represents the bias vector in linear combination transform $h^{(t-1)}$ and $x^{(t)}$. U represents the connection weight matrix from hidden states to hidden layer units.

(3) The fusion of multi-channel signal features was achieved using a combination of spatial and temporal feature fusion. Sparse feature matrices acquired from the multi-layer extraction of each recurrent network were stacked according to the channel number of the input signal. This fusion function was defined as: $f : x_t^1 + x_t^2 + \dots + x_t^n \rightarrow y_t$, where $x_t^1, x_t^2, \dots, x_t^n$ represent the characteristic matrices for each deep recurrent network at time t . The integrated feature matrix after fusion can be represented as follows:

$$\begin{bmatrix} x_{11}^{(t)} & x_{21}^{(t)} & \dots & x_{n1}^{(t)} \\ x_{12}^{(t)} & x_{22}^{(t)} & \dots & x_{n2}^{(t)} \\ \dots & \dots & \dots & \dots \\ x_{1n}^{(t)} & x_{2n}^{(t)} & \dots & x_{nn}^{(t)} \end{bmatrix}. \quad (17)$$

(4) The training softmax classifier accepts integrated feature matrices as input and produces multi-channel signal sample labels. The classification function is defined as:

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \dots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \dots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}. \quad (18)$$

The weighting parameter matrix is given by:

$$\theta = \begin{bmatrix} \theta_1^T \\ \theta_2^T \\ \dots \\ \theta_k^T \end{bmatrix}_{k \times (n+1)}. \quad (19)$$

In the training process, over-fitting is prevented using a dropout item [33] for control. The cost function for the softmax classifier during the training phase is given by:

$$\begin{aligned} J(\theta) &= -\frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \cdot \log(p(y^{(i)} = j | x^{(i)}; \theta)) \right] \\ &\quad + \frac{\lambda}{2} \sum_{i=1}^k \sum_{j=0}^n \theta_{ij}^2, \end{aligned} \quad (20)$$

where θ_j is a row vector representing the weight of each input x connected to the j^{th} classification output. Division in the logarithm above can be expressed in terms of subtraction:

$$\begin{aligned} J(\theta) &= -\frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \cdot \log(\theta_j^T x^{(i)}) - \log\left(\sum_{l=1}^k e^{\theta_l^T x^{(i)}}\right) \right] \\ &\quad + \frac{\lambda}{2} \sum_{i=1}^m \sum_{j=0}^n \theta_{ij}^2, \end{aligned} \quad (21)$$

where, $\lambda \in (0, 1)$ is a balance parameter.

By implementing the stochastic gradient descent algorithm, the updated iteration formula for softmax classifier parameters can be expressed as:

$$\begin{aligned} \frac{\nabla J(\theta)}{\nabla \theta_j} &= -\frac{1}{m} \sum_{i=1}^m \left[\frac{\nabla \sum_{j=1}^k 1\{y^{(i)} = j\} \theta_j^T x^{(i)}}{\nabla \theta_j} \right. \\ &\quad \left. - \frac{\nabla \sum_{j=1}^k 1\{y^{(i)} = j\} \log\left(\sum_{l=1}^k e^{\theta_l^T x^{(i)}}\right)}{\nabla \theta_j} \right] + \lambda \theta_j \\ &= -\frac{1}{m} \sum_{i=1}^m x^{(i)} \left[1\{y^{(i)} = j\} - p(y^{(i)} = j | x^{(i)}; \theta) \right] + \lambda \theta_j \end{aligned} \quad (22)$$

(5) P-RNN model parameters were optimized using the BP algorithm and the training set $\{x^{(t)} \in \mathbb{R}^n, y^n \in \mathbb{R}^m\}_{t=1}^T$.

IV. APPLICATION TO ECG SIGNAL CLASSIFICATION

ECG signals reflect changes in the electrical potential of the human heart. These data often exhibit multiple peaks, periodicity, non-stationary waveforms, and high levels of background noise. Various cardiovascular diseases correspond to different signal distribution characteristics [34].

A. THE EXPERIMENTAL DATASET

The data used in this validation experiment consisted of 12-lead ECG signal samples from the Chinese Cardiovascular Disease Database (CCDD) [35]. The sampling frequency was 500 Hz and each recording interval was longer than 10 seconds. Samples were marked by beat segmentation and the diagnosis of medical experts. Delete samples with vague markers, insufficient duration, or incomplete waveforms in candidate data. Seven such classes including 50408 samples were identified to construct the sample dataset. Relevant information, such as disease name, sample distribution, and category code are shown in Table 1.

As the dimension and magnitude of each lead differed, data were normalized using:

$$x'(t) = x(t) - \min x(t) / \max x(t) - \min x(t), \quad (23)$$

where $\min x(t)$ and $\max x(t)$ are the minimum and maximum values of ECG lead signals in a measurement interval.

TABLE 1. The distribution of experimental sample data.

Dictionary code	Disease type	Sample number	Category code
0X010101	Normal	16694	0
0X020201	Sinus arrhythmia	8058	1
0X020301	Sinus tachycardia	6197	2
0X030201	Atrial premature beat	5650	3
0X050202	Ventricular premature beat	5814	4
0X060305	Bundle branch block	3892	5
0X080205	Ventricular high voltage	4103	6
Total			50408

B. THE P-DRNN MODEL FOR ECG SIGNAL CLASSIFICATION

In the experiment, 12 input channels were established in a parallel GRU recurrent network, corresponding to the 12-lead ECG signals. The parallel structure unit was composed of 12 deep GRU recurrent networks, each of which was composed of 6 GRU information unit layers. The first layer was a time-series input layer and the other layers were GRU hidden layer units. There were 128 neurons in the first through fourth hidden layers, 64 neurons in the fifth layer, and 128 neurons in the sixth layer. A ReLU linear function was used as the activation function. Each channel input signal corresponded to a GRU deep recurrent network during feature extraction, producing a 128-dimensional feature vector. The feature fusion layer integrated signal feature vectors in each channel to form a multi-channel signal integrated feature matrix with dimensions of 12×128 . A softmax classifier developed from this matrix was used to classify multi-channel signals. This classifier included 12×128 input nodes and 7 output nodes.

C. EXPERIMENTAL RESULTS

1) ANALYSIS

The sample set was randomly divided into 2 groups according to the proportion of the disease. The training set consisted of 37,806 samples while the test set included 12,602. Property parameters and connection weights in the P-RNN were determined by the learning algorithm discussed in Section 3. The experiment was performed using an NVIDIA TITAN X GPU with a core frequency of 1418 MHz. Since the ECG sampling duration varied in the data set, a consistent processing step was included for standardization. In this process, 15 seconds were taken as the sampling interval. The parts that exceed 15 seconds were truncated, and a portion of less than that is complemented by a numerical value of 1000 [36]. The training error was set to 0.05, the maximum number of iterations was 10,000. Test samples were then classified and identified. The classification results for 7 disease types and various evaluation indicators are shown in Table 2, achieving a recognition accuracy of 95.86%.

In 12-lead ECG signals, each lead is indicative of different symptoms corresponding to different types of

TABLE 2. Experimental results for the proposed P-RNN.

Disease type	Accuracy(%)	Recall rate(%)	Precision (%)	F1-Score(%)
Normal	1	0.98	0.99	0.99
Sinus arrhythmia	0.9907	0.95	0.96	0.96
Sinus tachycardia	0.9843	0.98	0.97	0.97
Atrial premature beat	0.9452	0.94	0.93	0.94
Ventricular premature beat	0.7457	0.72	0.74	0.75
Bundle branch block	0.8115	0.80	0.81	0.81
Ventricular high voltage	0.9238	0.91	0.92	0.91

cardiovascular diseases. As such, the relative importance of each signal differs depending on the diagnosis. The P-RNN can independently extract signal characteristics for each lead during information processing and adaptively select the lead signal with the most significant effect on classification using feature weighting. The combined characteristics in local regions of 12-lead ECG signals also have an important impact on disease diagnosis. The P-RNN can simultaneously describe the role of these two factors through the learning of various models in large-scale training sets and the association of category attributes.

2) CONTRAST EXPERIMENTS AND ANALYSIS

This study conducted a series of contrast experiments utilizing three other deep neural network models that can directly classify multi-channel process signals. This included the multi-channel deep convolutional neural network proposed by Yi in 2014 [37], the LSTM+RF deep neural network based on an LSTM recurrent network, a random forest classification algorithm proposed by Sharma in 2018 [38], and the deep GRU recurrent network proposed by Rajan in 2018 [39]. These three models were compared to the proposed network using a series of tests involving the same training sample set and test set for disease classification and discrimination.

In this experiment, the architecture for the MC-DCNN model was I-C1(Size)-S1-C2(Size)-S2-H-O, where Size denotes the kernel size, C1 and C2 denote the number of filters, and S1 and S2 denote the subsampling factors. I, H, and O respectively denote the number of input layers, units in the hidden layer, and MLP output layer. Using comparative analysis, this architecture was determined to be 12-8(5)-2-4(5)-2-440-10. A series of two LSTM networks were constructed in the RNN + RF model and the number of hidden layers in each LSTM was set to 6. A random forest classifier was then established in the feature vector space to achieve classification, and the number of trees was 160. The GRU recurrent network was composed of 12 layers of GRU information units.

A 4-fold crossover experiment was conducted in which the sample set was randomly divided into 4 groups, according to

TABLE 3. Experimental results for comparable models.

Model	Accuracy(%)	Recall rate(%)	F1(%)	Training time(s)
MC-DCNN	85.84	85.3	84.8	19134.7
GRU-DRNN	93.97	91.22	91.19	50169.6
LSTM+RF	92.93	92.68	92.31	71342.8
The proposed method	95.86	94.73	95.66	4355.3

TABLE 4. The confusion matrix for 7 classification ECG signals using the proposed technique.

Type	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	0.092	0.9907	0.0001	0	0	0	0
2	0	0.0136	0.983	0	0.001	0	0
3	0	0.0049	0.0493	0.9452	0.0006	0	0
4	0	0.0012	0.0028	0.2459	0.7457	0	0.0008
5	0	0	0.0039	0.0020	0.1585	0.8115	0.0241
6	0	0	0.0017	0.0084	0.0661	0	0.9238

TABLE 5. The confusion matrix for a serial GRU recurrent network.

Type	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	0.092	0.9757	0.0001	0	0	0	0
2	0	0.0286	0.9743	0	0.0121	0	0
3	0	0.0049	0.0493	0.9452	0.0006	0	0
4	0	0.0012	0.0028	0.4495	0.5457	0	0.0008
5	0	0	0.0039	0.0020	0.2586	0.7124	0.0241
6	0	0	0.0017	0.0084	0.0661	0	0.9238

the proportion of disease type, with 12602 samples per group. Three of these were combined to form a training set while the remaining group served as the test set. Average values for each evaluation index across four experimental trials were used as the contrast index. These results are shown in Table 3, where it is evident that the proposed method achieved the best results for each evaluation index. This is primarily because our method is based on information fusion of characteristics from each lead signal in the mechanism, which can better maintain distribution characteristics for single-lead signals and structural information between 12-lead signals. This improved the identification of 12-lead sample distributions. This multi-channel parallel computing framework considers model parallelism and data parallel processing, which lower parallel runtime by a factor of 11.519 (compared with serial computing). The high recognition rate, stability, and generalizability of the network have been significantly improved.

Feature recognition ability were used to compare the serial recurrent network and parallel models to samples with similar morphological distributions. Two methods were used to calculate a confusion matrix for recognition results from 7 classification tests. These results are shown in Tables 4 and 5 and

suggest the method presented in this paper can significantly improve the ability to distinguish samples with similar distribution characteristics and other easily confused samples. This is because 12-lead ECG signals are periodic and parallel models are better than serial models in describing characteristics and combination relationships for multi-channel signals. In addition, some diseases are only sensitive to multiple lead signals when typical features of the disease are present in only a local region of the signal. Serial processing considers long signals connected by all leads, which affects and suppresses the extraction and storage of local signal features. Parallel processing effectively overcomes this limitation.

V. CONCLUSION

In this paper, a parallel-structure deep recurrent neural network was established for structural feature extraction and multi-channel time-varying signal classification. In the model, an existing serial processing RNN for multi-channel signals was designed as a parallel processing structure for each single-channel signal. Each channel corresponded to a deep recurrent network used to extract feature information. The feature integration of each channel signal was then completed in the feature fusion layer. This approach considered both model and data parallelism, facilitating multi-core and multi-thread parallel computing. The proposed P-RNN was applied to the classification of 12-lead ECG signals to validate its effectiveness. Experimental results showed that the performance evaluation index and the confusion matrix properties improved significantly, indicating the P-RNN can distinguish distribution characteristics for single-channel signals and structural features for multi-channel signals. The results of 4-fold crossover experiment show that it improved the robustness and generalizability of the model, which increased the computational efficiency. As a result, the proposed network has significant potential for application to multi-channel signal classification.

REFERENCES

- [1] H.-M. Shim and S. Lee, "Multi-channel electromyography pattern classification using deep belief networks for enhanced user experience," *J. Central South Univ.*, vol. 22, no. 5, pp. 1801–1808, May 2015.
- [2] P. Clemson, G. Lancaster, and A. Stefanovska, "Reconstructing time-dependent dynamics," *Proc. IEEE*, vol. 104, no. 2, pp. 223–241, Feb. 2016.
- [3] P. C. Nayak, K. P. Sudheer, D. M. Rangan, and K. S. Ramasastri, "A neuro-fuzzy computing technique for modeling hydrological time series," *J. Hydrol.*, vol. 291, nos. 1–2, pp. 52–66, 2004.
- [4] H. Chao, L. Dong, Y. Liu, and B. Lu, "Emotion recognition from multiband EEG signals using CapsNet," *Sensors*, vol. 19, no. 9, p. 2212, May 2019.
- [5] C. Li, C. Zheng, and C. Tai, "Detection of ECG characteristic points using wavelet transforms," *IEEE Trans. Biomed. Eng.*, vol. 42, no. 1, pp. 21–28, Jan. 1995.
- [6] S. Bououden, M. Chadli, and H. R. Karimi, "Control of uncertain highly nonlinear biological process based on Takagi–Sugeno fuzzy models," *Signal Process.*, vol. 108, pp. 195–205, Mar. 2015.
- [7] J. Meyer and K. U. Simmer, "Multi-channel speech enhancement in a car environment using wiener filtering and spectral subtraction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, Apr. 1997, pp. 1167–1170.
- [8] S. Sun and C. Zhang, "Adaptive feature extraction for EEG signal classification," *Med. Biol. Eng. Comput.*, vol. 44, no. 10, pp. 931–935, Oct. 2006.
- [9] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, and M. Adam, "Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals," *Inf. Sci.*, vol. 415, pp. 190–198, Nov. 2017.
- [10] S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.
- [11] J. Touch, Y.-S. Wang, and V. Pingali, "A recursive network architecture," USC/ISI, Tech. Rep. ISI-TR-2006-626, 2006.
- [12] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. 3, no. 1, pp. 4–16, Jan. 1986.
- [13] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2017.
- [14] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2015, pp. 167–174.
- [15] T. Shen, J. Jiang, T. Zhou, S. Pan, G. Long, and C. Zhang, "DiSAN: Directional self-attention network for RNN/CNN-free language understanding," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, pp. 5446–5455.
- [16] R. Messina and J. Louradour, "Segmentation-free handwritten Chinese text recognition with LSTM-RNN," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 171–175.
- [17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.
- [18] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.
- [19] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [21] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, Apr. 2004.
- [22] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
- [23] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct. 2013, pp. 1044–1054.
- [24] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: <https://arxiv.org/abs/1409.1259>
- [25] J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork RNN," 2014, *arXiv:1402.3511*. [Online]. Available: <https://arxiv.org/abs/1402.3511>
- [26] C. Li, B. Xu, G. Wu, S. He, G. Tian, and Y. Zhou, "Parallel recursive deep model for sentiment analysis," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2015, pp. 15–26.
- [27] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 241–248.
- [28] M. Wang, L. Song, X. Yang, and C. Luo, "A parallel-fusion RNN-LSTM architecture for image caption generation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 4448–4452.
- [29] V. Turchenko and L. Grandinetti, "Parallel batch pattern BP training algorithm of recurrent neural network," in *Proc. IEEE 14th Int. Conf. Intell. Eng. Syst.*, May 2010, pp. 25–30.
- [30] A. Larhlimi and M. Mestari, "GPU parallel neural hierarchical multi objective solver for burst routing and wavelength assignment," *Eng. Appl. Artif. Intell.*, vol. 75, pp. 48–63, Oct. 2018.
- [31] D. Połap, K. Kęsik, M. Woźniak, and R. Damaševičius, "Parallel technique for the metaheuristic algorithms using devoted local search and manipulating the solutions space," *Appl. Sci.*, vol. 8, no. 2, p. 293, Feb. 2018.
- [32] R. J. Martis, U. R. Acharya, K. M. Mandana, A. K. Ray, and C. Chakraborty, "Application of principal component analysis to ECG signals for automated diagnosis of cardiac health," *Expert Syst. Appl.*, vol. 39, no. 14, pp. 11792–11800, Oct. 2012.
- [33] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1027–1035.

- [34] C. Saritha, V. Sukanya, and Y. N. Murthy, "ECG signal analysis using wavelet transforms," *Bulg. J. Phys.*, vol. 35, no. 1, pp. 68–77, Feb. 2008.
- [35] J.-W. Zhang, X. Liu, and J. Dong, "CCDD: An enhanced standard ecg database with its management and annotation tools," *Int. J. Artif. Intell. Tools*, vol. 21, no. 5, Oct. 2012, Art. no. 1240020.
- [36] S. Zhang, H. Jiang, M. Xu, J. Hou, and L. Dai, "A fixed-size encoding method for variable-length sequences with its application to neural network language models," 2015, *arXiv:1505.01504*. [Online]. Available: <https://arxiv.org/abs/1505.01504>
- [37] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *Proc. Int. Conf. Web-Age Inf. Manage.* Cham, Switzerland: Springer, 2014, pp. 298–310.
- [38] L. D. Sharma and R. K. Sunkaria, "Inferior myocardial infarction detection using stationary wavelet transform and machine learning approach," *Signal, Image Video Process.*, vol. 12, no. 2, pp. 199–206, 2018.
- [39] D. Rajan and J. J. Thiagarajan, "A generative modeling approach to limited channel ECG classification," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 2571–2574.



KUN LIU was born in Shandong, China, in 1990. He received the B.Sc. and M.Sc. degrees in mathematics from Qufu Normal University, in 2013 and 2017, respectively. He is currently pursuing the Ph.D. degree with the College of Computer Science and Engineering, Shandong University of Science and Technology. His current research interests include machine learning, fuzzy sets, and neural networks.



SHAOHUA XU received the B.Sc. degree in mathematics from the Northeastern University of Petroleum, in 1983, the M.Sc. degree in applied mathematics from the Harbin Institute of Technology, in 1986, and the Ph.D. degree in computer software and theoretical engineering from Beihang University, in 2004. His current research interests include large data analysis, intelligent information processing technology, and 3D visualization data modeling technology.



JINGJING LI was born in Ningxia, China, in 1993. She is currently pursuing the M.Sc. degree with the College of Computer Science and Engineering, Shandong University of Science and Technology. Her current research interests include deep learning, machine learning, and neural networks.



LU WU received the B.Sc. degree from the College of Computer Science and Engineering, Northwest Normal University, in 2017, and the master's degree from Northwest Polytechnic University, in 2006. He is currently pursuing the Ph.D. degree with the College of Computer Science and Engineering, Shandong University of Science and Technology. He is currently an Associate Professor with the Shandong Academy of Sciences and the Deputy Director of National Supercomputer Center, Jinan. His research interests include the applications of AI and bigdata technology in medicine.

• • •