EURASIP Journal on Advances
in Signal Processing

RESEARCH Open Access

Electrocardiogram prediction based on variational mode decomposition and a convolutional gated recurrent unit

HongBo Wang[1,2*], YiZhe Wang[1,2†], Yu Liu[1†] and YueJuan Yao[1†]

[†]YiZhe Wang, Yu Liu and YueJuan Yao have contributed equally to this work.

*Correspondence:
foreverwhb@126.com

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2] Beijing Key Laboratory of Knowledge Engineering for Materials Science, University of Science and Technology Beijing, Beijing 100083, China

Abstract

Electrocardiogram (ECG) prediction is highly important for detecting and storing heart signals and identifying potential health hazards. To improve the duration and accuracy of ECG prediction on the basis of noise filtering, a new algorithm based on variational mode decomposition (VMD) and a convolutional gated recurrent unit (ConvGRU) was proposed, named VMD-ConvGRU. VMD can directly remove noise, such as baseline drift noise, without manual intervention, greatly improving the model usability, and its combination with ConvGRU improves the prediction time and accuracy. The proposed algorithm was compared with three related algorithms (PSR-NN, VMD-NN and TS fuzzy) on MIT-BIH, an internationally recognized arrhythmia database. The experiments showed that the VMD-ConvGRU algorithm not only achieves better prediction accuracy than that of the other three algorithms but also has a considerable advantage in terms of prediction time. In addition, prediction experiments on both the MIT-BIH and European ST-T databases have shown that the VMD-ConvGRU algorithm has better generalizability than the other methods.

Keywords: Electrocardiogram prediction, Time series, Variational mode decomposition, Convolutional gated recurrent unit

1 Introduction

Time series data are generated during the process of regular observation and collection of certain phenomena. In the medical field, many biomedical data can be used as time series. An electrocardiogram (ECG) is a graphical representation of the electrical potential of the heart and is commonly used to detect the presence of cardiovascular disease (CVD). In general, an ECG has three main components: a P wave, a QRS composite wave and a T wave. The ECG in Fig. 1 was recorded at a rate of 25 mm/s (5 large cubes/s) [1]. In the vertical direction, 10 mm represents 1 mV, and each 1 mm square represents 0.04 s (40 ms) and 0.10 mV.

ECG prediction is a scientific analysis method based on time series data. If potential health hazards can be predicted in time, timely medical assistance can be provided, and the compressed storage of relevant signals can be facilitated. In the existing research on deep learning-based prediction of ECGs, there are notable limitations regarding

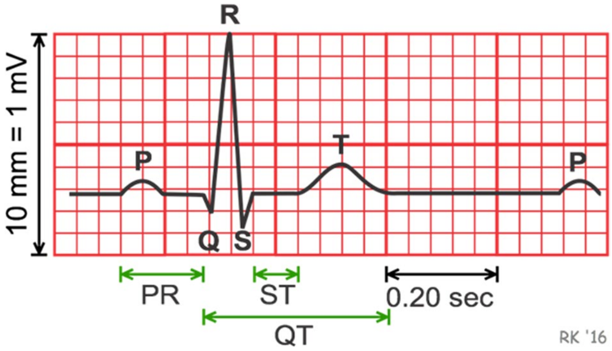


Fig. 1 Electrocardiogram

handling abnormal conditions, noise, and interference that can significantly affect the model performance. Furthermore, the majority of models tend to predict only a single point after the input signals, indicating that both the output time duration and the prediction accuracy require further optimization.

Therefore, to solve the key problems of noise filtering, such as baseline drift, and further improve the prediction time and accuracy, in this paper, the ECG prediction algorithm VMD-ConvGRU, which combines variational mode decomposition (VMD) and a convolutional gated recurrent unit (ConvGRU) was proposed, and a generalization test was carried out. Compared to other methods, VMD-ConvGRU is particularly well suited for ECG compression, variable-length signal restoration, and complex environment applications, such as real-world clinical settings. The following points summarize the contributions of this paper: (1) Using VMD to preprocess the data can effectively filter out noise such as baseline drift and capture different frequency domain characteristics of the actual signal through multiple adaptive segmentation component predictions. (2) The encoder–forecaster (EF) structure can freely change the input length and prediction data to realize continuous prediction for a period and effectively increase the prediction time. Moreover, multilayer information exchange can be realized between the stacked structure encoder and the predictor. Finally, combining VMD signal decomposition with EF-ConvGRU greatly improves the prediction accuracy.

The remainder of this article is structured as follows. Section 2 discusses related works. In Sect. 3, VMD and ConvGRU are theoretically analyzed and introduced. Section 4 proposes an improved ECG prediction algorithm. Section 5 presents some experimental analysis and comparisons with the relevant algorithms and their performance. Finally, the conclusion and future work are given in Section 6.

2 Related work

ECG signal preprocessing is an indispensable step in the prediction process. VMD is an adaptive and completely nonrecursive decomposition technique that has been gradually applied in various fields since it was first proposed [2].

Upadhyay et al. proposed a method based on VMD to detect sound or silent regions in speech signals [3]. Wang et al. combined the particle swarm optimization algorithm with VMD and applied it to the fault diagnosis of complex rotating machinery [4]. Lahmiri proposed a model combining VMD and the backpropagation neural network (BPNN), which decomposed a price sequence into a series of variational modes by VMD. Then, the BPNN was trained to predict the stock price of the day [5]. This approach has also been applied to ECG decomposition. The typical workflow in ECG signal prediction is as follows. Su et al. proposed a prediction method for ECG signals based on the combination of phase space reconstruction and the Takagi–Sugeno (TS) fuzzy model [6], which used only three data points to carry out experiments. Huang et al. proposed a prediction method for ECG signals based on the autoregressive integrated moving average (ARIMA) model and discrete wavelet transformation (DWT) [7]. Sun et al. proposed an ECG signal prediction method, VMD-NN, based on a combination of VMD and a backpropagation neural network [8], where an ECG signal was decomposed into n models by VMD and $n-k$ models other than k noise models were input into the backpropagation neural network. The network structure consists of three layers, the number of input layer nodes is 9, that is, $n-k=9$, the number of hidden nodes is 18, and the number of output nodes is 1. The simulation experiment only used one example (No. 100). The results showed that the waveform of the ECG signal is not affected by fault tolerance, verifying the effectiveness of the algorithm for the prediction of the ECG signal. In 2018, the team proposed a prediction model based on phase space reconstruction and a neural network [9]. The same example (No. 100) was used in the MIT-BIH database for the experiments to demonstrate the effectiveness of the algorithm. Huang et al. proposed an ECG signal prediction method combining VMD, the Cao method and a long short-term memory (LSTM) neural network, significantly improving the prediction accuracy [10].

Based on the analysis and summary of the relevant related research mentioned above, it is far from practical to predict an ECG signal at only one point each time, and there is still room for improvement in the accuracy of these algorithms. In addition, due to the limited data selection, systematic research on the generalizability of related algorithms has become highly important.

3 Methods

3.1 Variational mode decomposition

The goal of VMD is to decompose the original input signal into multiple intrinsic mode functions (IMFs). In the process of obtaining the decomposed components, VMD determines the frequency center and bandwidth of each component by iteratively searching for the optimal solution of the variational model to adaptively subdivide the signal frequency domain and effectively separate each component. The VMD defines the IMF as the amplitude-modulated–frequency-modulated (AM-FM) signal shown as

$$u_k(t) = A_k(t)\cos\left(\phi_k(t)\right) \tag{1}$$

where $u_k(t)$ is the modal component, $A_k(t)$ is the instantaneous amplitude of the modal component signal, and $\phi_k(t)$ is the phase of the modal component signal. The instantaneous frequency of $u_k(t)$ is defined as

$$\omega_k(t) = \phi_k'(t) = \frac{d\phi_k(t)}{dt} \tag{2}$$

It is assumed that each mode $u_k(t)$ has a center frequency and limited bandwidth; the constraint condition is that the sum of all the modes is equal to the input original signal *f*, and the sum of the estimated bandwidths of each mode is minimal. The constrained variational model is expressed as

$$\begin{aligned} &\min_{\{u_k\}\{w_k\}}\left\{\sum_k \left\|\partial_t\left[\left(\delta(t)+\frac{j}{\pi t}\right)*u_k(t)\right]\mathrm{e}^{-\mathrm{jw_k t}}\right\|^2\right\} \\ &\text{s.t. } \sum_k u_k = f \end{aligned} \tag{3}$$

Here, f is decomposed into k finite IMF components, namely, $\{u_k\} := \{u_1, u_2, \ldots, u_k\}$, and $\{w_k\} := \{w_1, w_2, \ldots, w_k\}$ is the central frequency of each component. The problem of obtaining the decomposed IMF is transformed into solving the constrained variational model. The penalty factor α and the Lagrange multiplicative operator λ are introduced, and Eq. (3) is transformed into

$$\begin{aligned} L(\{u_k\},\{\omega_k\},\lambda) := \alpha\sum_k \left\|\partial_t\left[\left(\delta(t)+\frac{j}{\pi t}\right)*u_k(t)\right]e^{-j\omega_k t}\right\|_2^2 \\ + \left\|f(t)-\sum_k u_k(t)\right\|_2^2 + \left\langle \lambda(t), f(t)-\sum_k u_k(t)\right\rangle \end{aligned} \tag{4}$$

In the solution process, the alternate direction method of multipliers (ADMM) is used to update $u_k^{(n+1)}$, $\omega_k^{(n+1)}$, and $\lambda_k^{(n+1)}$ are used to find the saddle point, which can be written as

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{f}(\omega) - \sum_{i\neq k}\hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2}}{1+2\alpha(\omega-\omega_k)^2} \tag{5}$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega\left|\hat{u}_k(\omega)\right|^2 \mathrm{d}\omega}{\int_0^\infty \left|\hat{u}_k(\omega)\right|^2 \mathrm{d}\omega} \tag{6}$$

$$\hat{\lambda}^{n+1}(\omega) \leftarrow \hat{\lambda}^n(\omega) + \tau\left(\hat{f}(\omega) - \sum_k \hat{u}_k^{n+1}(\omega)\right) \tag{7}$$

until the update conditions are met with

$$\sum_k \left\|u_k^{n+1} - u_k^n\right\|_2^2 / \left\|u_k^n\right\|_2^2 < \epsilon \tag{8}$$

Finally, k IMF components are obtained according to the frequency domain characteristics of the actual signal, and adaptive segmentation of the signal frequency band is completed.

In our application scenario, according to the formula in pseudocode [2], u_k, w_k and other parameters converge with appropriate parameter initialization and convergence criteria. Therefore, VMD optimized with the ADMM can successfully decompose any preprocessed ECG signals in the datasets mentioned in Sect. 5.1. For complexity, the algorithm uses two *for* loops successively, with complexity $O(K)$. In one cycle, the minimum value is selected, and the optimal complexity is $O(KlogK)$; thus, the final complexity is $O(K^2logK)$.

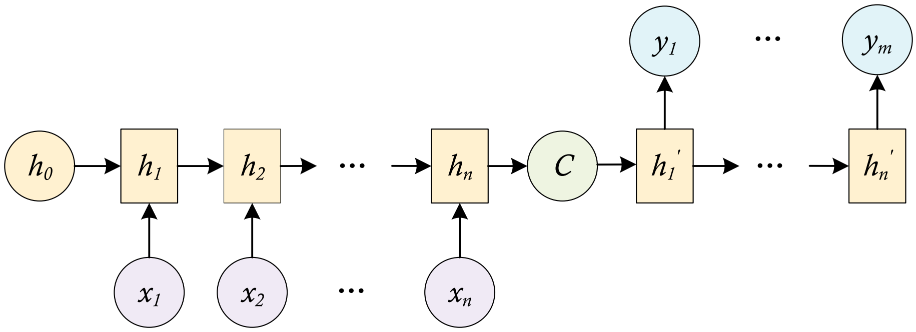


Fig. 2 Structure of the RNN sequence-to-sequence model

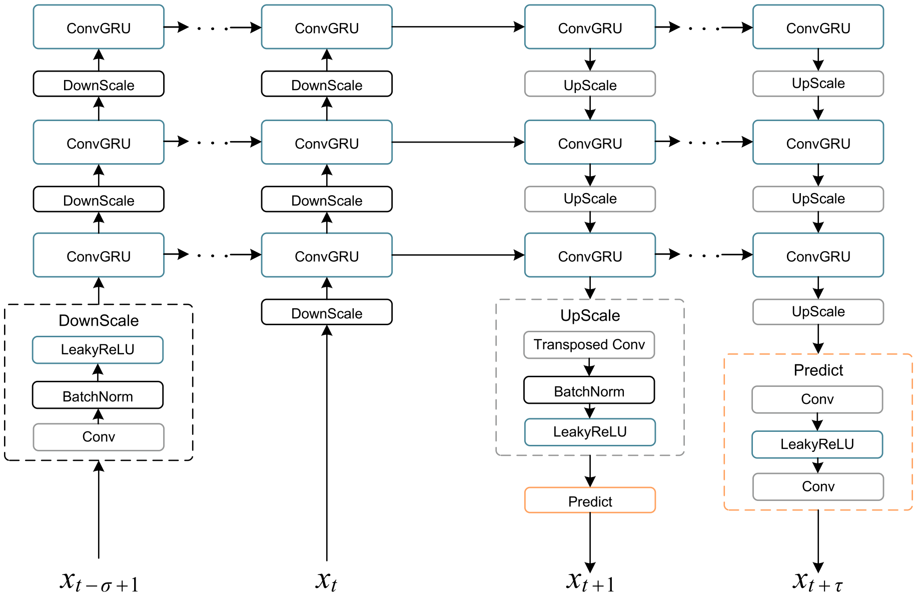


Fig. 3 EF-ConvGRU construction and sharing mechanism

3.2 Encoding–forecasting structure based on a convolutional gated recurrent unit

For a situation where the lengths of the input and output sections are inconsistent, a variant structure of the recurrent neural network (RNN) is a good choice; its specific implementation structure from sequence to sequence is shown in Fig. 2. Usually, this variant RNN structure first encodes an input sequence into a context vector C. C becomes a hidden initial variable of the decoding network, which is the forecaster module in the following content of this paper.

The encoding–forecasting structure based on a convolutional gated recurrent unit (EF-ConvGRU) can be regarded as a special encoding and decoding structure [11]. Instead of sharing only one set of information, in the coding prediction structure, the encoder and predictor share multiple sets of information, and there is information exchange between each layer. The structure is shown in Fig. 3. On the left, the information flowing from bottom to top is the encoder, and on the right, the information flowing from top to bottom is the predictor. The input of the encoder includes observations from $x_{t-\sigma+1}$ to x_t; that is, σ continuous states can be input, and a total of τ continuous states from x_{t+1} to $x_{t+\tau}$ can be predicted.

The encoder consists of two different modules, ConvGRU and DownScale, for reducing the data length. There are three different modules in the predictor, ConvGRU and UpScale for restoring the data length and Predict for converting the data into the final predicted value.

4 Proposed approach

4.1 Architecture of the VMD-ConvGRU model

The VMD and ConvGRU algorithms were integrated, and the VMD-ConvGRU algorithm framework was proposed. As shown in Fig. 4, the input is the original ECG signal of $\sigma \times len$ points, the output is the predicted $\tau \times len$ ECG signal and does not contain noise such as baseline drift, where σ represents the number of segments of the input ECG, τ represents the number of segments of the output ECG, and len represents the length of each ECG segment. The algorithm first decomposes the original ECG signal through VMD, removes some components to achieve noise reduction, then uses EF-ConvGRU to predict each IMF component, and finally combines the results to obtain the predicted ECG signal. On the one hand, it is convenient for simultaneous computation in a distributed system to achieve higher efficiency; on the other hand, pathology can be divided to carry out multidimensional analysis.

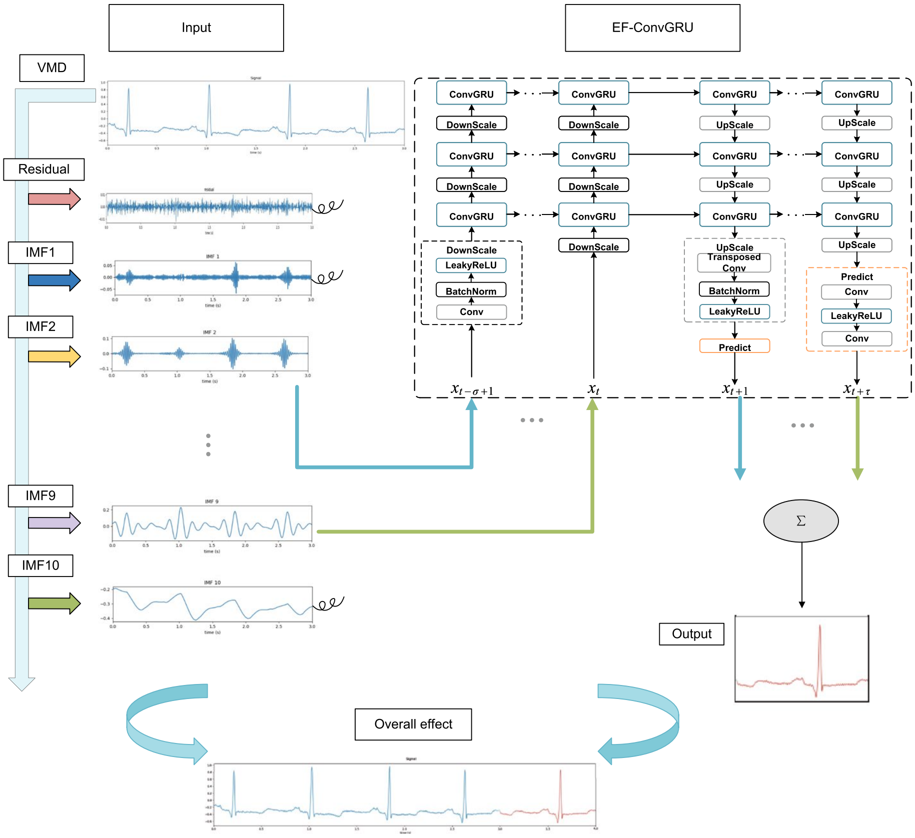


Fig. 4 Algorithm framework of VMD-ConvGRU

Different from image information, ECG signals are usually one-dimensional sequential signals. Therefore, the convolution operation in the ConvGRU is replaced with a one-dimensional convolution. Like two-dimensional convolution methods, one-dimensional convolution methods also involve parameters such as the kernel size, step size and padding. The difference is that the size of the one-dimensional convolution kernel is usually $n \times 1$, and it only moves along the direction of length n, which is the direction of step size movement. Padding is only applied at both ends in this direction. The parameters required for one-dimensional convolution are usually expressed as

$$parameters = in\_channels \times out\_channels \times kernal\_size \tag{9}$$

The classical GRU based on matrix multiplication can be denoted as

$$h_t = \mathrm{GRU}(x_t, h_{t-1}; W, b; \cdot) \tag{10}$$

The input parameters are the observed value at the current moment x_t and the hidden state at the previous moment h_{t-1}. The parameters in the cell mainly include the matrix W and bias b, while the operation is mainly matrix multiplication. The calculation of the ConvGRU is designed as

$$H_t = \mathrm{ConvGRU}\left(X_t, H_{t-1}; W, b;^{*}\right) \tag{11}$$

The input parameters X_t and H_{t-1} have the same meaning as those of the classical GRU, but the dimensions are changed to multichannel information. For the ConvGRU of the first layer, the input dimension changes from $\mathbb{R}^{L}$ to $\mathbb{R}^{1\times L}$, and 1 means that there is currently only one channel. The input dimension of the upper ConvGRU is $\mathbb{R}^{C_i\times L}$, where C_i indicates that there may be multiple channels. The dimension of H_{t-1} is $\mathbb{R}^{C_h\times L}$, and there may also be multiple channels. Moreover, the sizes of C_i and C_h can be different. The parameters in the cell mainly include the convolution kernel W and the bias b, and the operation is mainly a convolution operation. The final output $H_t \in \mathbb{R}^{C_h\times L}$ has the same dimension as that of H_{t-1}.

The Downscale and UpScale modules are also shown in Fig. 3. For DownScale, the data flow from the bottom to the top. First, a one-dimensional convolution module is used, and the stride of the convolution is set to 2 to reduce the length of the input data by half. Moreover, setting *out_channel* greater than *in_channel* can increase the understanding of more dimensional information while reducing the length. To speed up the training process, batch normalization (BN) operations are used after convolution to aggregate information from different samples, and the leaky rectified linear unit (LeakyReLU) is used as the activation function. For UpScale, the data flow from top to bottom. It is necessary to use transposed convolution to increase the length of the data while reducing *out_channel*. The operation is also followed by the BN and LeakyReLU operations.

Different from the ReLU calculation method [12], which directly zeroes parts less than 0, the LeakyReLU [13] activation function is shown as

$$LeakyReLU(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0.01 \times x, & \text{otherwise} \end{cases} \tag{12}$$

which uses a parameter to scale the part less than 0. In this paper, we set this value to the recommended value of 0.01.

Finally, after processing by the Predict module, the final predicted value can be obtained. The Predict module is shown in Fig. 3 and includes the convolution and LeakyReLU operations. In the prediction process, convolution is used to replace the fully connected layer. The advantage of this approach is that the number of parameters can be reduced while increasing the ability to gather information. By setting *out_channel* of the last convolutional layer to 1, an output with a channel number of 1 can be obtained, which can be used as the predicted value at a certain moment in the end.

4.2 Training process of the VMD-ConvGRU model

The ECG prediction algorithm based on VMD-ConvGRU includes various parameters, such as the length of the input data, the parameters of the convolution kernel in the hidden layer, the number of decomposition layers k, and the initialization of the parameters. In addition, the parameters in the neural network are continuously updated by learning the data in the training set until the number of iterations is reached.

The algorithm includes the following operations.

1. Determine the number of decomposition layers k. Considering the fairness of the comparison experiments with related algorithms, set $k = 10$.

2. Determine the length of the input data and the length to be predicted.

3. Set the EF-ConvGRU convolutional layer parameters, such as *kernal_size*, *stride*, *padding*, *in_channel*, and *out_channel*.

4. Determine *batch_size* and *epoch*.

5. Using the mean square error (MSE) as the loss function of each prediction, express the final loss as

$$Loss = \sum_{i=2}^{9} MSE\left(\hat{IMF}_i\right) \tag{13}$$

6. Initialize the convolution kernel and bias value.

7. Decompose the signal with VMD and remove the decomposed IMF 1, IMF 10 and residual.

8. Train the model and use dense sampling for data augmentation. Each turn rearranges the ECG signals in the training set.

9. Use the loss function to train the parameters of the convolution kernel, and stop when the set epoch is reached.

The pseudocode of the ECG prediction algorithm based on VMD-ConvGRU is shown in Algorithm 1.

Algorithm 1 ECG prediction algorithm based on VMD-ConvGRU

```
Require: $epoch$, $batch\_size$, $\sigma$, $\tau$, $len$
Ensure: Predicted ECG signals
Set the decomposition layers $k = 10$
Determine parameters of each convolution layer
Use the Kaiming uniform to initialize $W$, and use the uniform to initialize $b$
Decompose the signal with VMD and remove the decomposed $IMF1$, $IMF10$
and $residual$
for $e = 1$ to $epoch$ do
    Rearrange the training set order
    while Data is not used up do
        Read a batch of data
        $loss = 0$
        for $i = 2$ to $9$ do
            $\widehat{IMF}_i = EF\text{-}ConvGRU\,(IMF_i)$
            $loss+ = MSE\left(\widehat{IMF}_i\right)$
        end for
        Propagate the loss backward, update convolution kernel $W$ and $b$
    end while
end for
```

5 Experimental results and analysis

5.1 Experimental settings

The MIT-BIH database, an internationally recognized ECG database that can be used as a standard, [14], was generated by the laboratories of the Beth Israel Hospital and the Massachusetts Institute of Technology. Each piece of data contains three files, namely, the header file (.hea), the data file (.dat) and the annotation file (.atr). The header file records the signal name, the number of leads, the name of the lead, the sampling frequency, the number of sampling points, etc.; the data file records all signal points; the annotation file contains the diagnostic information of the expert on the ECG signal; and a detailed record is made for each heartbeat type of the ECG.

The signal sampling rate in this database is 360 Hz, and there are 48 two-lead ECG signals (mostly from the MLII and V1) collected in approximately 30 min. We used the MLII lead data and screened out 19 data points for experiments based on the proportion of normal heartbeats exceeding 95%. The first 80% of each piece of data was used as the training set for a total of 24 min, and the remaining 20% was used as the test set for a total of 6 min.

When $\sigma = \tau = 2$ and $len = 36$, the parameter settings of the network structure are shown in Table 1.

The experiment was run on a computer with the Windows 10 64-bit operating system and an Intel Core i5-8250U CPU with 8 GB of memory and written in Python 3.8.5 using PyTorch 1.6.0.

Table 1 Parameter setting of the module encoder and forecaster

Module	Submodule	in_channel	out_channel	kernal_size	stride	padding
Encoder	Downscale	1	8	2	2	0
	ConvGRU	8	64	3	1	1
	Downscale	64	128	2	2	0
	ConvGRU	128	128	3	1	1
	Downscale	128	128	3	2	1
	ConvGRU	128	128	3	1	1
Forecaster	ConvGRU	128	128	3	1	1
	Upscale	128	128	3	2	1
	ConvGRU	128	128	3	1	1
	Upscale	128	64	2	2	0
	ConvGRU	64	64	3	1	1
	Upscale	64	8	2	2	0
	Predict-conv1	8	8	3	1	1
	Predict-conv2	8	1	1	1	0

5.2 Evaluation indicators

The evaluation metrics used in the simulation experiment include the mean absolute error (MAE), mean square error (MSE) and root mean square error (RMSE): which are, respectively defined as

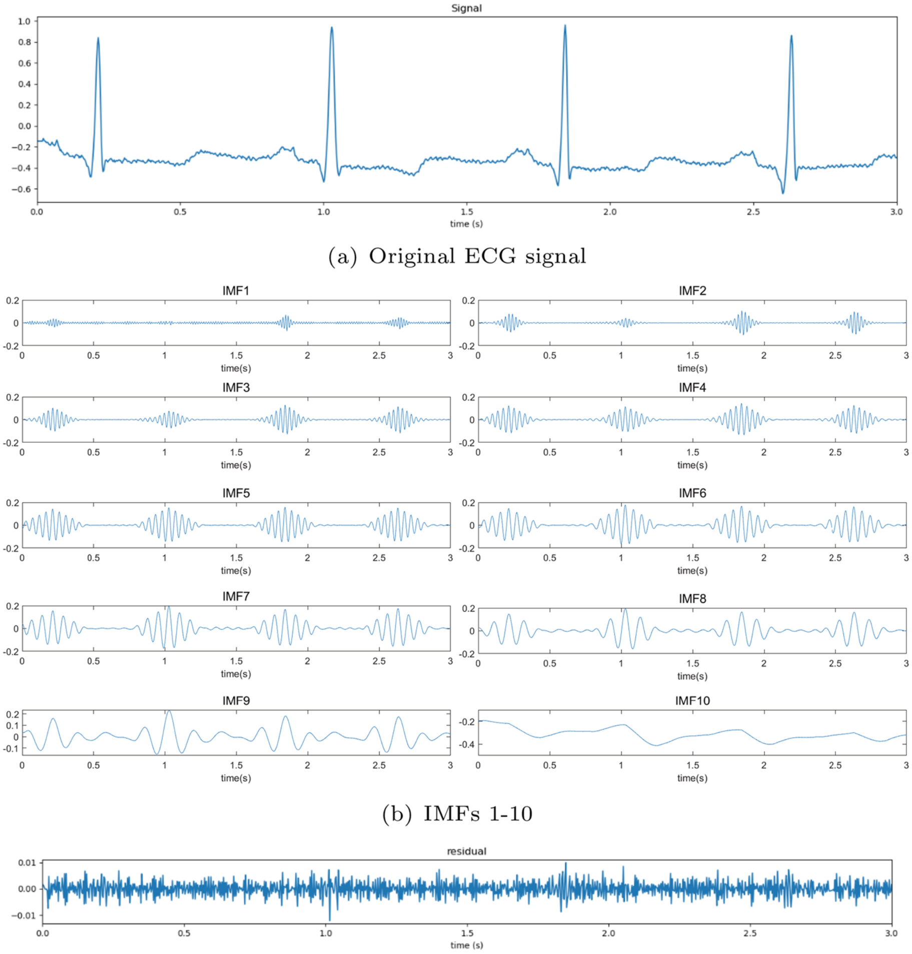
$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{14}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2} \tag{16}$$

5.3 VMD of the ECG signal

To show the effect of VMD on signal decomposition, we used the VMD tool [15] provided in MATLAB to decompose the first 3 s of the 100th signal in the MIT-BIH database. The decomposition results are shown in Fig. 5. The number of decomposition layers k was set to 10, and the other parameters were set to the default parameters provided by MATLAB. Figure 5a shows the original ECG signal. Figure 5b shows the decomposed components IMFs 1-10, where IMF 1 can be regarded as a high-frequency noise signal, IMF 10 can be regarded as baseline wander. and Fig. 5c is the undecomposed residual signal.



(a) Original ECG signal

(b) IMFs 1-10

(c) Residual signal

Fig. 5 Decompose the first 3 s original signal of example 100 with VMD

A more intuitive representation is shown in Fig. 6, where the x-axis is time (in seconds), the y-axis is the corresponding amplitude, and the z-axis is the residual signal, IMF, or original signal, respectively. The value of the residual signal is very small relative to the original signal, and the frequency gradually decreases from IMF 1 to IMF 10.

The signal decomposed by VMD can be used to reconstruct the original signal, and its corresponding relationship can be expressed as

$$Signal = \sum_{i=1}^{10} IMF_i + residual \tag{17}$$

During reconstruction, the high-frequency noise of IMF 1, the baseline drift of IMF 10, and the residual signal were removed, and the signals were synthesized by

$$Signal' = \sum_{i=2}^{9} IMF_i \tag{18}$$

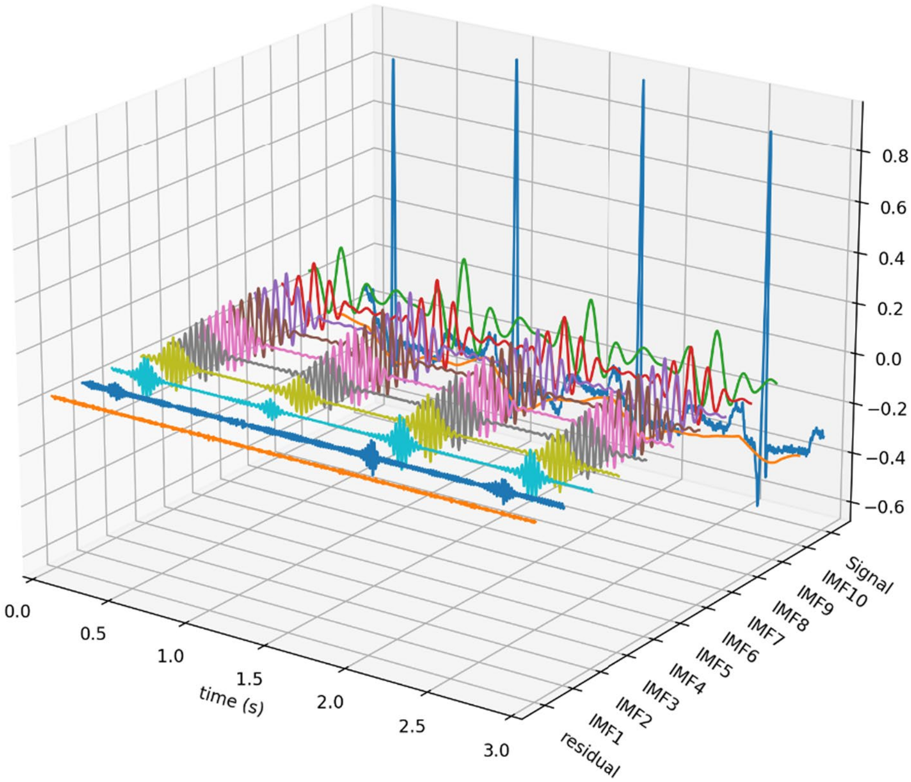


Fig. 6 3D display for the first 3 s of example 100 after VMD

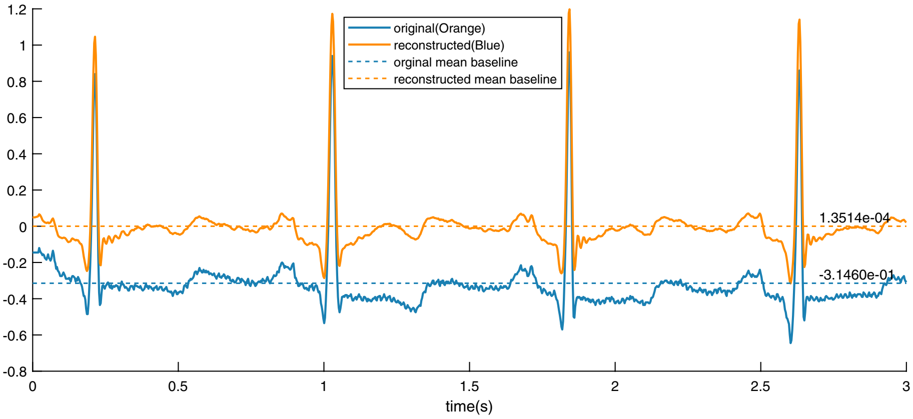


Fig. 7 Signal reconstruction for the first 3 s of the example 100

We reconstructed the first 3 s of signal 100, and the comparison is shown in Fig. 7. The reconstructed ECG signal is smoother in the time domain than in the other domains, and the quality of the original signal is preserved. Additionally, the baseline signal was evaluated using the mean. In Fig. 7, the reconstructed ECG baseline is determined to be $1.3514E-04$, which is very close to zero. This result demonstrates that the reconstructed signal effectively removes noise and corrects baseline drift. In subsequent experiments, the reconstructed signal was used as the predicted target value.

5.4 Parameter analysis

In the parameter analysis experiment, we used example 100 as training and test data. These data contain a total of 30 min of ECG signals. The first 80% was used as the training set, a total of 24 min, and the remaining 20% was used as the test set, a total of 6 min. For each predicted IMF component, the MSE was used as the loss function. In addition, the adaptive moment estimation (Adam) optimizer was used, the weight decay was set to 0.0001, the initial learning rate was 0.01, and the learning rate was reduced when the loss no longer decreased. In each experiment, 50 epochs were trained.

The parameters that were set in the EF-ConvGRU model include the input time length σ, the prediction time length τ, and the input length at each moment *len*. These three parameters are discussed separately below.

1) Input time length.

To explore the effect of σ on the result, we fixed $\tau = 2$ and $len = 36$ and then set $\sigma =$ 2, 4, 6, 8, and 10; that is, the inputs were 0.2 s, 0.4 s, 0.6 s, 0.8 s, and 1 s, respectively. The effect of different input lengths on the result was obtained. The change in loss is shown in Fig. 8. When the input duration is 0.2 s, 0.4 s, or 0.8 s, the convergence trend is relatively stable. For an input duration = 0.8 s, the VMD-ConvGRU converges the earliest, i.e., at epoch 25. For an input duration = 0.2 s, the VMD-ConvGRU converges at epoch 33, which is better than the corresponding value of 0.4 s. The other models with different input durations converge after 50 epochs. As the input duration increases, the model exhibits improved performance after convergence. The convergence loss values are very close when the input duration is 0.2 s and 0.4 s. In addition, as the input duration increases, such as with values of 0.6 s or 1 s, the model clearly exhibits significant fluctuations in performance during the initial training stages. This result can be attributed to the larger amount of information the model receives at once compared to that at shorter input durations, resulting in greater signal variations. Consequently, additional

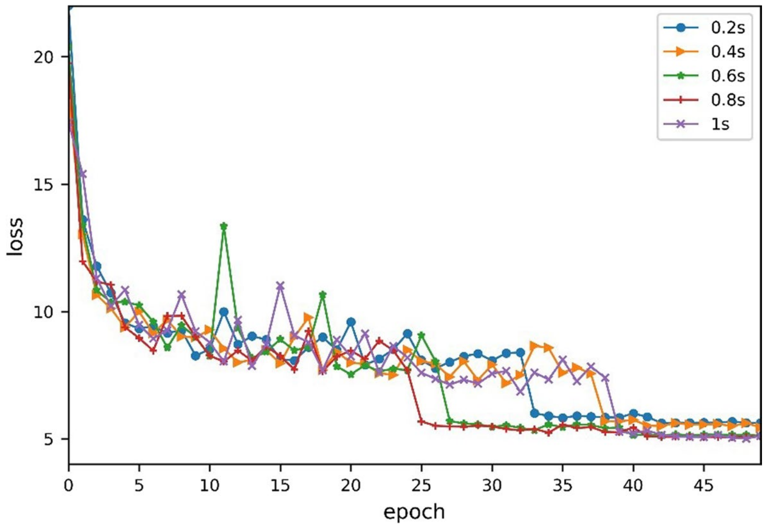


Fig. 8 Changes in the loss of the VMD-ConvGRU algorithm for example 100 for different input durations

Table 2 Performance comparison of VMD-ConvGRU on example 100 with different σ values

σ	MSE	MAE	RMSE
2 (0.2s)	4.2173E−04	9.7310E−03	0.0205
4 (0.4s)	4.1936E−04	9.7199E−03	0.0204
6 (0.6s)	4.0401E−04	9.2763E−03	0.0201
8 (0.8s)	3.9777E−04	9.1632E−03	0.0199
10 (1s)	3.8782E−04	9.4672E−03	0.0196

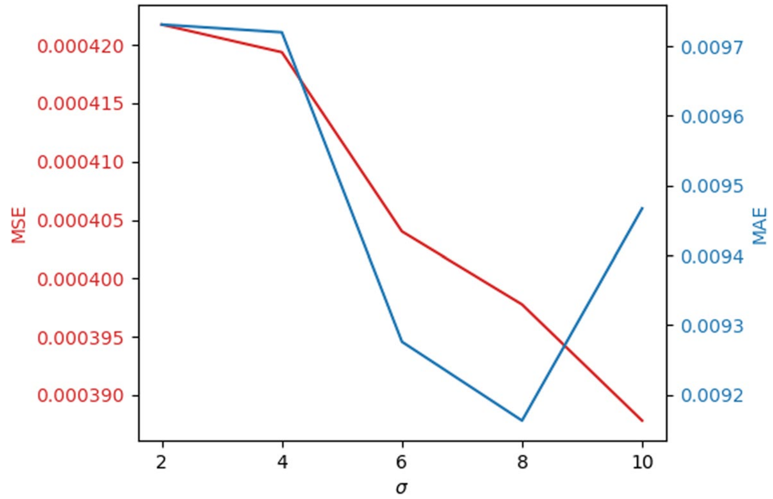


Fig. 9 MSE and MAE line charts for VMD-ConvGRU on example 100 with different input durations

adjustments are needed to effectively extract relevant information, ultimately leading to improved predictive outcomes.

To verify the prediction effect of the model with different input durations, we conducted a test, and the results are shown in Table 2. When the input $\sigma = 10$, the effect is the best, and the performance is similar to that on the training set. Therefore, the greater the number of input periods is, the better the performance, but the greater the additional computational overhead. Figure 9 shows the comparison line charts of the MSE and MAE for different input durations.

2) Forecast length.

To explore the impact of τ on the results, we fixed $\sigma = 10$ and $len = 36$ and then set $\tau = 1, 2, 3, 4$, and 5; that is, the outputs were 0.1 s, 0.2 s, 0.3 s, 0.4 s, and 0.5 s, respectively; and the influence of different output lengths on the prediction results was obtained. The change in loss is shown in Fig. 10. The output duration has a greater impact on the loss, and the shorter the prediction time is, the better the training effect.

To verify the prediction effect, we tested the trained model. The experimental results are shown in Table 3. As the output duration increases, the prediction performance of the model decreases. A possible reason for this result is that as the length of the data transfer increases, some information is lost, reducing the prediction accuracy. In addition, an increase in τ reduces the inference speed. Figure 11 shows that as the output duration increases, the MSE gradually increases, and the model fitting ability gradually decreases.

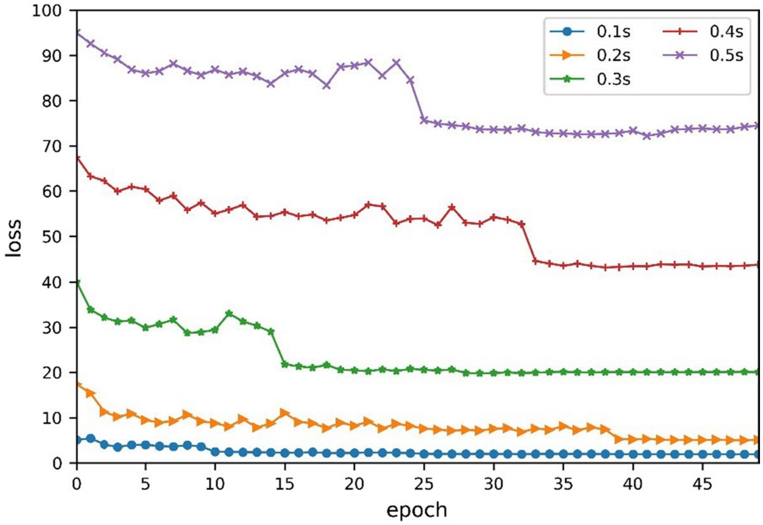


Fig. 10 Changes in the loss of the VMD-ConvGRU algorithm for example 100 with different prediction durations

Table 3 Performance comparison of VMD-ConvGRU on example 100 with different prediction durations

τ	MSE	MAE	RMSE
1	2.8695E−04	5.1096E−03	0.0169
2	3.8782E−04	9.4672E−03	0.0196
3	2.1028E−03	1.8909E−02	0.0458
4	4.7850E−03	2.8002E−02	0.0691
5	9.2696E−03	3.6550E−02	0.0962

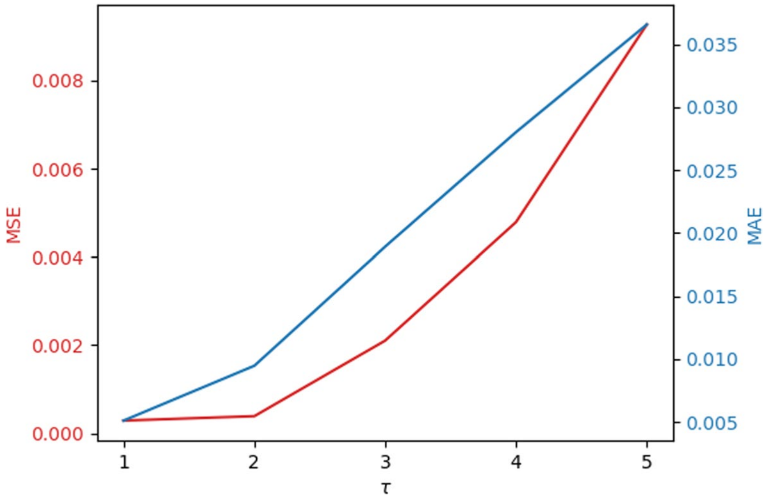


Fig. 11 MSE and MAE line charts for VMD-ConvGRU on example 100 with different prediction durations

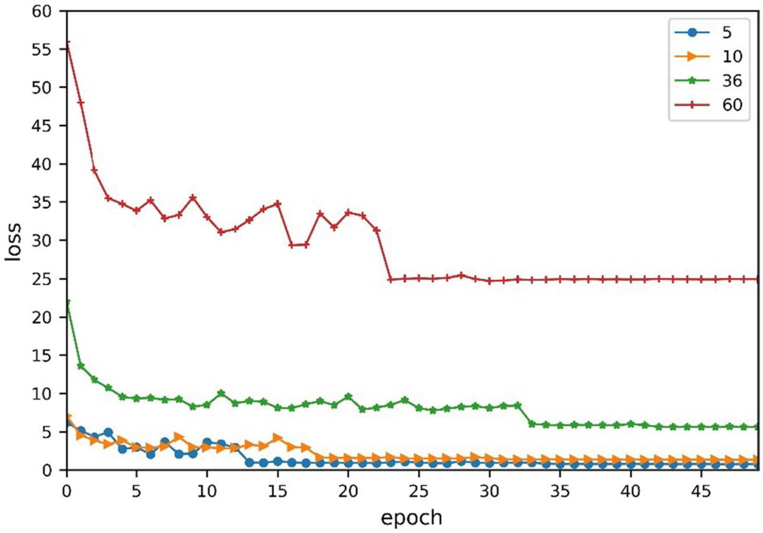


Fig. 12 Loss change line chart for VMD-ConvGRU on example 100 with different *len* values

Table 4 Performance comparison of VMD-ConvGRU on example 100 with different *len* values

len	MSE	MAE	RMSE
5	1.3756E−04	5.1124E−03	0.0117
10	2.0580E−04	5.7412E−03	0.0143
36	4.2173E−04	9.7310E−03	0.0205
60	2.4239E−03	2.0607E−02	0.0492

3) Input length at each moment.

As shown above, the larger σ is, the better the prediction result, and the larger τ is, the worse the result. When $\sigma = \tau = 2$, a model with more accurate prediction results and better inference performance can be obtained. To explore the impact of the input value at each moment *len* on the result, we fixed $\sigma = \tau = 2$, and then set $len = 5$, 10, 36, 60, namely, 1/72 s, 1/36 s, 0.1 s, 1/6 s. The change of Loss is shown in Fig. 12. As *len* increases, the training effect worsens, and the model effect worsens when $len = 60$.

The performance on the test set is shown in Table 4. With the continuous increase in *len*, the prediction effect of the model decreases rapidly. When the length continues to increase, the model can no longer predict normally.

5.5 Experimental results and analysis

5.5.1 Performance comparison with related methods

To verify the prediction effect of VMD-ConvGRU, we compared our method with the three methods proposed in three related papers, and the results are shown in Table 5. When $\sigma = \tau = 2$ and $len = 5$, the RMSE (0.0117) and MAE (0.0051) of the proposed VMD-ConvGRU model are both smaller than the corresponding values of the VMD-NN [8], PSR-NN [9], and TS fuzzy models [6]. When $len = 36$, 0.2 s can be predicted at a time, which significantly increases the prediction time length, and the prediction performance does not decrease too much; in particular, the MAE indicator still has a great advantage.

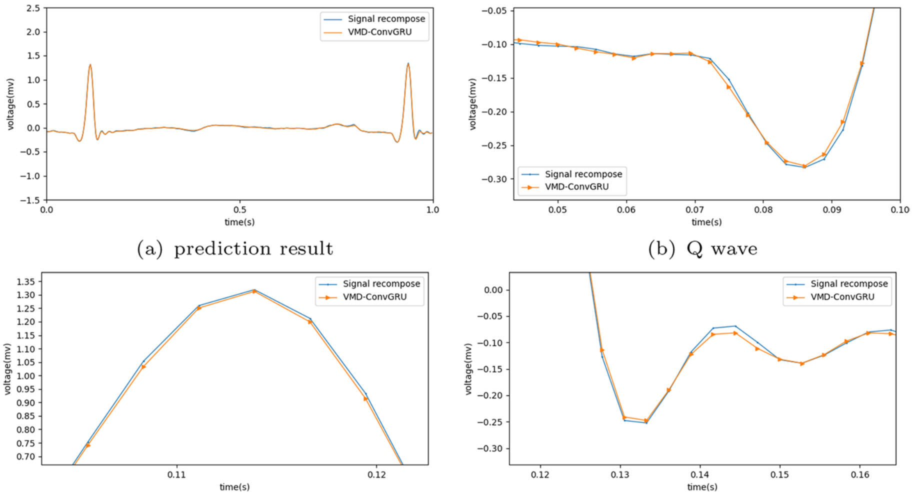
Table 5 Performance comparison of different algorithms on example 100 (VMD-NN, PSR-NN, TS fuzzy model and VMD-ConvGRU) and example 113 (TS fuzzy model and VMD-ConvGRU)

Data	Model	RMSE	MAE
100	VMD-NN	0.0233	0.0157
	PSR-NN	0.0423	0.0240
	TS Fuzzy Model	0.0146	0.0106
	Proposed (*len*= 5)	**0.0117**	**0.0051**
	Proposed (*len*= 36)	0.0205	0.0097
113	TS Fuzzy Model	0.0281	0.0172
	Proposed (*len*= 36)	**0.0259**	**0.0147**

The best values for RMSE and MAE on the 18 selected examples

Notably, the above three methods can predict only one point at a time, namely 1/360 s. However, VMD-ConvGRU can predict at least 1/72 s. VMD-ConvGRU increases the prediction duration while ensuring that the prediction accuracy is still improved. In addition, the experimental results of the above three methods are all part of the extraction from example 100, i.e., not all the data were used for training and testing. Our proposed method directly uses the first 24 min as training and the last 6 min as testing. Furthermore, we selected example 113 for comparison with the TS fuzzy model method. Our model achieves better performance, as shown in Table 5. Therefore, our model has better generalizability.

Figure 13a shows the prediction result at 1 s for example 100. The prediction results of the VMD-ConvGRU model proposed in this paper fit the reconstructed signal well. Figure 13b–d are the prediction results of the Q, R, and S waves, respectively, corresponding to the first heartbeats in Fig. 13a. The trends of the three waveforms are well predicted, and most of the predicted curves overlap with the original curves. Figure 14 shows the predicted difference. The error remains within a small range, and most of the differences are within 0.01.



(a) prediction result

(b) Q wave

(c) R wave

(d) S wave

Fig. 13 1 s prediction result using VMD-CONVGRU compared with true recombination signal on example 100

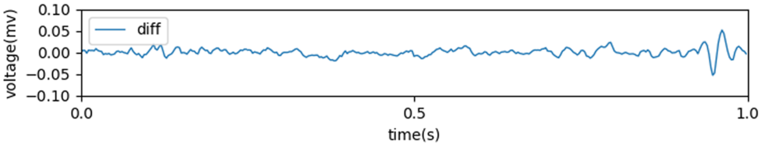


Fig. 14 Prediction bias of VMD-CONVGRU on 1 s of example 100

5.5.2 Generalizability of VMD-ConvGRU

We also conducted experiments on 18 examples selected from MIT-BIH and 18 other examples from a new database named the European ST-T Database (EDB) [16] to verify the model generalizability. The EDB contains 90 two-hour dual-channel (V4 and MLIII) examples, from which MLIII data were selected. Each example was truncated to the same length as the examples selected from MIT-BIH. We set $len = 36$ and $\sigma = \tau = 2$ for examples from MIT-BIH. Since the sampling frequency of the new database is 250 Hz, we set $len = 25$, and the other parameters were kept unchanged. All the results are shown in Table 6.

The best values for RMSE and MAE on the 18 selected examples in MIT-BIH are 0.0093 and 0.0067, respectively, while the corresponding values for EDB are 0.0088 and 0.0057, respectively. To evaluate the overall performance of VMD-ConvGRU, we selected two evaluation metrics: the mean and variance. The mean and variance of the RMSE and MAE are 0.0095 and 0.005, respectively, on 18 examples in the EDB. In general, the prediction effect and stability of VMD-ConvGRU are excellent, and its generalizability has been verified.

Table 6 Generalizability results of VMD-ConvGRU on 36 examples from MIT-BIH and EDB

Dataset	MIT-BIH			EDB			
#	MSE	MAE	RMSE	#	MSE	MAE	RMSE
100	4.22E−04	9.73E−03	0.0205	e0103	1.09E−04	7.46E−03	0.0104
101	1.65E−04	8.03E−03	0.0128	e0104	3.01E−04	1.02E−02	0.0174
103	2.71E−04	9.84E−03	0.0165	e0105	5.34E−04	1.42E−02	0.0231
105	5.23E−03	3.37E−02	0.0724	e0106	1.92E−04	8.01E−03	0.0138
108	1.39E−03	2.35E−02	0.0373	e0108	8.73E−04	1.72E−02	0.0295
112	1.35E−04	7.82E−03	0.0116	e0110	2.60E−04	9.18E−03	0.0161
114	1.11E−03	2.10E−02	0.0333	e0111	1.46E−03	1.97E−02	0.0382
115	3.38E−04	1.06E−02	0.0184	e0114	1.42E−04	7.72E−03	0.0119
116	3.20E−03	3.38E−02	0.0566	e0115	1.65E−03	2.46E−02	0.0406
117	**8.66E−05**	**6.69E−03**	**0.0093**	e0116	3.27E−04	1.25E−02	0.0181
121	1.92E−03	2.78E−02	0.0438	e0118	1.15E−04	7.39E−03	0.0107
122	2.31E−04	1.03E−02	0.0152	e0123	3.49E−04	1.04E−02	0.0187
123	3.42E−04	9.60E−03	0.0185	e0124	5.41E−04	1.16E−02	0.0233
202	9.83E−04	2.13E−02	0.0313	e0125	2.07E−04	8.11E−03	0.0144
205	1.25E−03	1.30E−02	0.0354	e0126	**7.69E−05**	**5.71E−03**	**0.0088**
220	6.13E−04	1.50E−02	0.0248	e0133	1.24E−04	7.38E−03	0.0111
230	2.33E−03	2.81E−02	0.0483	e0147	1.02E−04	6.89E−03	0.0101
234	2.24E−04	1.05E−02	0.0150	e0154	8.29E−05	6.02E−03	0.0091
Average	1.12E−03	1.67E−02	0.0289		4.22E−04	1.10E−02	0.0184
Std	1.34E−03	9.25E−03	0.0174		4.61E−04	5.10E−03	0.0095

The best values for RMSE and MAE on the 18 selected examples

According to the fitting degrees of the predicted signals and comparisons with other ECG prediction algorithms, the proposed VMD-ConvGRU model not only improves the prediction duration but also ensures the accuracy of the prediction results and has excellent generalizability.

The method proposed by Huang et al. [10] improved the accuracy of ECG prediction, with an RMSE and MAE of 0.001326 and 0.001044, respectively, for 100 data points. Although our method cannot be compared in the same experimental environment, VMD-ConvGRU can predict data continuously for a period, and we have carried out generalization experiments on 36 ECG samples, which is more widely applicable.

6 Conclusions

In this paper, a novel network model, VMD-ConvGRU, was proposed. Based on the concept of signal decomposition, the VMD method was utilized to decompose and filter the electrocardiogram signal into ten dimensions. The convolutional GRU (ConvGRU) was then integrated with VMD, enabling signal prediction and evaluation in nine dimensions. The performance of the proposed VMD-ConvGRU was compared to that of the PSR-NN, VMD-NN, and TS fuzzy model using example 100 from the MIT-BIH dataset. The proposed VMD-ConvGRU demonstrates a 5% improvement in performance, as measured by the MAE, compared to that of the second best performing model. Additionally, the model generalizability was evaluated on 18 examples from both the MIT-BIH dataset and the European ST-T database. The results indicate that VMD-ConvGRU effectively filters out noise caused by baseline drift and enhances the prediction accuracy.

VMD-ConvGRU currently focuses on predicting electrocardiograms from a single healthcare provider at different time points. However, further research is needed to expand the study in terms of dataset sample size and the inclusion of data from multiple healthcare providers. Future research will aim to concentrate on predicting electrocardiograms from multiple individuals across different age groups, sexes, and even nationalities. Additionally, the focus will be on predicting unhealthy electrocardiograms and identifying potential health risks.

Acknowledgements
Not applicable.

Author contributions
Hong-Bo Wang and Yue-Juan Yao analyzed the electrocardiogram data and guided the experiment. Yi-Zhe Wang and Yu Liu processed the dataset and implemented a predictive model for training, and Yi-Zhe Wang was a major contributor to the writing of the manuscript. All the authors read and approved the final manuscript.

Funding
This work is partly supported by the Key Technologies Research and Development Program of China (No. 2020YFB1712104) and National Natural Science Foundation of China (No. 61572074).

Availability of data and materials
The datasets used and analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 19 September 2023 Accepted: 15 January 2024
Published online: 25 January 2024

References

1. R. Klabunde, *Cardiovascular physiology concepts* (Lippincott Williams & Wilkins, USA, 2011)
2. K. Dragomiretskiy, D. Zosso, Variational mode decomposition. IEEE Trans. Signal Process. **62**(3), 531–544 (2014)
3. A. Upadhyay, R.B. Pachori, Instantaneous voiced/non-voiced detection in speech signals based on variational mode decomposition. J. Franklin Inst. **352**(7), 2679–2707 (2015)
4. X.B. Wang, Z.X. Yang, X.A. Yan, Novel particle swarm optimization-based variational model e-composition method for the fault diagnosis of complex rotating machinery. IEEE/ASME Trans. Mechatron. **23**(1), 68–79 (2017)
5. S. Lahmiri, Intraday stock price forecasting based on variational mode decomposition. J. Comput. Sci. **12**, 23–27 (2016)
6. F. Su, H. Dong et al., Prediction of ecg signal based on ts fuzzy model of phase space reconstruction, in *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI 2019)*, Suzhou, China, p. 1–6 (2019)
7. F. Huang, T. Qin, L. Wang et al., An ecg signal prediction method based on arima model and dwt, in *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. Chengdu, China, p. 1298–1304 (2019)
8. Z.G. Sun, Y. Lei, J. Wang et al., An ecg signal analysis and prediction method combined with vmd and neural network, in *2017 IEEE 7th International Conference on Electronics Information and Emergency Communication (ICEIEC 2017)*, Shenzhen, China, p. 199–202 (2017)
9. Z.G. Sun, Q. Wang, Q. Xue et al., Data prediction of ecg based on phase space reconstruction and neural network, in *2018 IEEE 8th International Conference on Electronics Information and Emergency Communication (ICEIEC 2018)*, Beijing, China, p. 162–165 (2018)
10. F. Huang, T. Qin, L. Wang et al., A Deep Learning Method for ECG Signal Prediction Based on VMD, Cao Method, and LSTM Neural Network. https://doi.org/10.21203/rs.3.rs-139350/v1 (2021)
11. X. Shi, Z. Gao, L. Lausen et al., Deep learning for precipitation nowcasting: A benchmark and a new model, in *Advances in neural information processing systems (NIPS 2017)*, Long Beach, USA, p. 5617–5627 (2021)
12. V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, Haifa, Israel, p. 807–814 (2010)
13. A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, Atlanta, USA, p. 1152–1160 (2013)
14. G.B. Moody, R.G. Mark, The impact of the mit-bih arrhythmia database. IEEE Eng. Med. Biol. Mag. **20**(3), 45–50 (2001)
15. MathWorks, Documentation-Variational mode decomposition. MathWorks Help Center. https://ww2.mathworks.cn/help/signal/ref/vmd.html. Accessed 26 Oct 2020 (2020)
16. A. Taddei, G. Distante, M. Emdin, P. Pisani, G.B. Moody, C. Zeelenberg, C. Marchesi, The European ST-T database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography. Eur. Heart J. **13**, 1164–1172 (1992)

Publisher's Note