Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 4, April 2019, pp. 1161-1170
DOI: 10.23940/ijpe.19.04.p11.11611170

Hybrid SVM and ARIMA Model for Failure Time Series Prediction based on EEMD

Haiyan Sun[a], Jing Wu[a,*], Ji Wu[b,*], and Haiyan Yang[b]

[a]*School of Mathematics and Systems Science, Beijing University of Aeronautics and Astronautics, Beijing, 100191, China*
[b]*School of Computer Science and Engineering, Beijing University of Aeronautics and Astronautics, Beijing, 100191, China*

Abstract

A more widely used hybrid model of support vector regression (SVR) and autoregressive integrated moving average (ARIMA) based on Ensemble Empirical Mode Decomposition (EEMD) is proposed for failure time series prediction by taking advantage of the SVR model to forecast the nonlinear part of failure time series and the ARIMA model to predict the linear basic part. It firstly uses EEMD to decompose the original failure sequence into several significant fluctuation components and a trend component, and then it utilizes SVR and ARIMA to forecast them separately. The performance of the presented model is measured against other unitary models such as Holt-Winters, autoregressive integrated moving average, multiple linear regression, and group method of data handling of seven published nonlinear non-stationary failure datasets. The comparison results indicate that the proposed model outperforms other techniques and can be utilized as a promising tool for failure data forecast applications.

Keywords: ensemble empirical mode decomposition; support vector machines regression; autoregressive integrated moving average; failure time series forecast; hybrid models

(Submitted on October 23, 2018; Revised on November 24, 2018; Accepted on December 26, 2018)

1. Introduction

Software technologies are widely used nowadays, and they play a key role in achieving highly complicated and safety-critical missions in many fields. The process of software development is increasingly complicated, and the difficulty of software quality management increases correspondingly, making the performance of software more difficult to be evaluated than ever before. The reliability is one of the important indicators to measure the quality of software. It refers to the possibility that the software can perform its intended functions during a period of running time without any failure [1]. Software reliability analysis involves using statistical methods to establish the corresponding model based on software failure data and then evaluate and predict the software's reliability. The software reliability model can be divided into two categories in accordance with the principle of prediction: analysis model and data-driven model [2]. The analysis model, such as software reliability growth models (SRGM), is often difficult to achieve ideal modeling and prediction results due to unsatisfactory assumptions in practical applications [3]. The data-driven model has good adaptive ability because it does not make any assumptions about software internal defects, failure, and troubleshooting procedures, and it has been gradually proven to be an effective method for studying software failure data [4].

Numerous single data-driven models, such as Auto-Regressive Integrated Moving Average (ARIMA) [5], Support Vector Regression (SVR) [4], and Artificial Neural Network (ANN) [6], have been applied effectively to study failure data [7]. These data are in regular patterns and have limited data points, but for more complex failure time series data with no obvious patterns, such as the nonlinear and non-stationary characteristics, the unitary model cannot always predict it effectively (seen in section 3). The reason for this result is that the unitary model is only suitable for data with certain characteristics. Therefore, we consider decomposing the failure time series into subsequences to obtain different features and fitting them respectively. For data decomposition, Ensemble Empirical Mode Decomposition (EEMD) is usually applied. Wang et al. utilized the ARIMA model based on EEMD decomposition to forecast annual runoff time series [8].

* Corresponding author.
E-mail address: sunhy@buaa.edu.cn

Plakandaras et al. used SVR technology based on EEMD decomposition and Elastic Net feature selection to forecast U.S. housing prices from 1890 to 2012 [9]. Žvokelj et al. developed multi-ICA technology based on EEMD decomposition to detect and diagnose rotary bearing failure [10]. These works commonly use EEMD as a preprocessing technique. Predictions of all components are completed by one model, and then the model result can be obtained by combining the multiple component predictions. Since the components obtained by EEMD have different characteristics, unified prediction of all components by a single model may lead to a loss of data information. To capture the different characteristics reflected by components, multiple models are considered for prediction and combination.

Based on this framework, a combined technique is proposed to focus on the non-stationary nonlinear characteristics of software failure sequence. In EEMD decomposition, two significant components recognized as the fluctuation and trend are extracted from the original failure time series data, and two models named trend-ARIMA and fluctuation-SVR are individually developed to fit and forecast the two components.

The rest of this paper is organized as follows. In Section 2, the EEMD combinational model is proposed. In Section 3, the experimental results are discussed. Finally, the conclusions are drawn.

2. Combined Model

2.1. Reliability Prediction Framework for Combined Models

Different models can be used for combinational prediction according to the results of EEMD decomposition to make full use of the information reflected by the original failure data. According to the characteristics of the failure interval series decomposed by EEMD, the framework of the proposed EEMD-SVR-ARIMA method is shown in Figure 1. Except for the data preparation, there are three steps: decomposition, modeling, and combination. The following focuses on the three core modules of decomposition, model calculation, and parameter optimization and combination model.

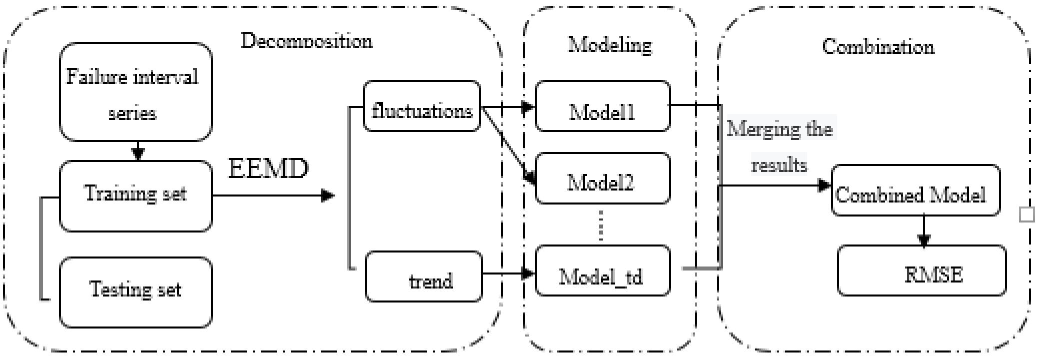


Figure 1. The framework of failure interval series analysis

2.2. Ensemble Empirical Mode Decomposition

Commonly used decomposition methods include short-time Fourier and wavelet analysis. This paper chooses EEMD as the decomposition method. EEMD utilizes the statistical characteristics of Gaussian white noise with uniform frequency distribution, adding auxiliary white noise to the original component to eliminate the intermittent phenomenon of the original component. It makes the original component have continuity at different scales. Furthermore, EEMD uses the characteristics of multiple uncorrelated random white noise tests to cancel the effects of noise in the decomposition results, effectively weakening the phenomenon of mode mixing in empirical mode decomposition [11]. The EEMD decomposition is adaptively acquired in an iterative manner according to the characteristics of the components, and the basis function is decomposed by the data itself. The basis function and the decomposition layer number are not required to be specified, which is a completely data-driven method. This method is intuitive, direct, and adaptive, which has obvious advantages in dealing with nonlinear data.

The failure interval series **TF** and the divided training set **TR** and test set **TE** are defined as Equations (1) to (3):

$$TF = \{x_1, x_2, \cdots, x_n\} \tag{1}$$

$$TR = \{x_{tr_1}, x_{tr_2}, \cdots, x_{tr_n}\} \tag{2}$$

$$TE = \{x_{te_1}, x_{te_2}, \cdots, x_{te_n}\} \tag{3}$$

The constraints are met as Equations (4) and (5):

$$Tr \subset TF,\ Te \subset TF,\ Tr \cap Te = \varnothing \tag{4}$$

$$tr_n + te_n \le t_n,\ tr_n < te_1,\ tr_n, te_n \ne 0 \ \ \forall t \in N \tag{5}$$

Where tr_n is the size of the training set, x_{tr_n} is the time interval between the occurrence of the $tr_n{}^{\text{th}}$ failure and the occurrence of the $(tr_n - 1)^{\text{th}}$ failure, te_n is the size of the testing set, and x_{te_n} is the time interval between the occurrence of the $te_n{}^{\text{th}}$ failure and the occurrence of the $(te_n - 1)^{\text{th}}$ failure.

The decomposition process takes *TR* as the input, and the decomposition steps can refer to literature [11]. An original time series is decomposed into the sum of wave components and a trend component by EEMD, and the decomposed expression can be written as Equation (6):

$$EEMD\!: TR \to TR = \sum_{j=1}^{J} c_j + r,\ J = \left[\log_2(tr_n)\right] \tag{6}$$

2.3. Model Calculation and Parameter Optimization

A series of sub-components with obvious features, different frequency, and different amplitude are obtained from the original failure sequence decomposed by EEMD. Support vector machine (SVM) [12] is established for the components with large frequency, small amplitude, and obvious fluctuation characteristics, and the ARIMA [13] model is produced for the components with small frequency, large amplitude, and obvious trend characteristics.

To fully reflect the correlation characteristics inside the failure series, it is necessary to construct the series data into phase space. According to Takens' theory, the key of phase space construction is to select the appropriate embedded dimension m and delay time τ to reconstruct phase space. In this paper, the observation interval ΔN is taken as the delay time τ and is the unit. That is, ΔN =1, with one failure difference existing. A mapping relationship between the sliding time window $[x_{t-1}, x_{t-2}, \cdots, x_{t-m}]$ and the output $x_t = f(x_{t-1}, x_{t-2}, \cdots, x_{t-m})$ is established. The fluctuation components $\mathbf{x} = \{x_1, x_2, \cdots, x_n\}$ is converted to Equation (7):

$$\mathbf{x} = \{x_1, x_2, \cdots, x_n\} = \begin{pmatrix} x_1 & x_2 & \cdots & x_m \\ x_2 & x_3 & & x_{m+1} \\ \vdots & & \ddots & \vdots \\ x_{n-m+1} & x_{n-m+2} & \cdots & x_n \end{pmatrix} \tag{7}$$

Where the single-variable failure sequence components used for the SVR model is converted to matrix form to obtain the correlation between its data.

After the single variable failure data component is transformed into matrix form by formula (7), it can be further modelled according to the support vector regression learning and prediction framework of Figure 2. The former m data is used to predict the $(m+1)^{\text{th}}$ data, and then it is iterated successively until the end.

It can be seen from Equations (7) to (10) that by controlling the embedding dimension parameter m and the hyper-parameters c, γ, and ε of the SVR model, the generalization ability of the SVR model can be controlled. Therefore, how to select the four parameters reasonably and quickly has a great impact on application effectiveness and scope of the SVM.

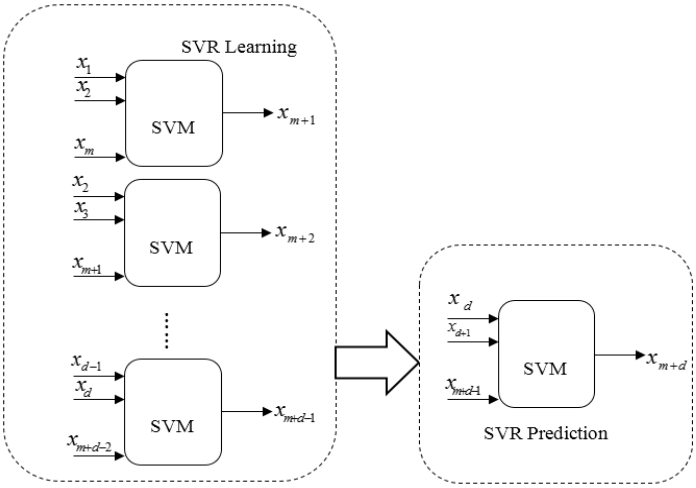


Figure 2. The framework of support vector regression learning and prediction

The determination of the embedding dimension parameter m follows two principles: 1) it cannot be greater than half the length of the time components, 2) it can divide exactly the known period of the time components. For non-periodic failure time interval data, Lou and Jiang indicated that the best prediction effect can be obtained when m is 5-25 [14], and the text is determined by 5 to 25 according to the criterion of the smallest final prediction error. In this paper, m is determined in 5-25 according to the criterion of the smallest final prediction error.

For the hyper-parameters c, γ, and ε of the SVR model, commonly used existing mature parameter optimization algorithms include the grid search method [15], genetic algorithm [16], particle swarm algorithm [17], and simulated annealing algorithm [18]. In this paper, a practical heuristic method [19] is used to determine the coarse value of the parameter directly from the training data, a “good area” of the parameter is set by reducing and enlarging the coarse value of the parameter, and finally the grid search is used. The method selects the optimal parameters in the “good zone” according to the principle that the final empirical error is the smallest.

An ARIMA (p,d,q) model is established for the trend component, where p,d and q can be determined by taking the difference and calculating the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) value as its minimum.

2.4. Combination

The core idea of model prediction is to decompose the original failure time interval sequence into components of different strong features by EEMD, including the fluctuations and the trend, respectively establish a prediction model for the components, and then combine the results to obtain the prediction result. The model combination process is shown in Figure 1.

An original failure sequence is decomposed into the sum of several wave components and one trend component by EEMD, and the modeling process uses two models, SVR and ARIMA. Due to the additivity of the EEMD decomposition, the models created by the subcomponents can be added together to obtain the final model. The expression for the model is given by Equation (8):

$$\textit{Combined forecast}:\ \mathrm{M}_{com} = \sum_{j=1}^{J} \mathrm{M}_{SVR_j}(c_j) + \mathrm{M}_{ARIMA}(r) \tag{8}$$

3. Experimental Result

In this paper, R is the development tool, which models and forecasts the failure time series by using several R packages, including parameter estimation, model establishment, model fitting, and prediction of failure data. The comparison results of EEMD decomposition, subcomponent modeling, and combination model are presented in the form of Table. At the same time, Excel is used to present the prediction contrast effect of the model.

3.1. Experimental Data

To verify the effectiveness of the proposed method, this paper selects seven datasets collected by Bell Labs [20]: 5, SS1A, SS1B, SS1C, SS2, SS3, and SS4. They are from the real-time business system, operating system, time sharing system, and word processing system respectively. These datasets are all the failure time interval data of each software system. This paper compares and analyses the seven failure datasets and compares them with classic models such as Holt-Winters, ARIMA, MLR, GMDH, RVM, and SVR. In particular, the dataset SS3 is selected for EEMD decomposition, sub-component modeling, and combined model component reconstruction prediction, which are presented in the form of figures.

3.2. Model Evaluation

There are many standards for evaluating the fitting effect of software reliability model on failure interval data. In this paper, RMSE is selected as the evaluation criterion for models. The formula is shown as Equation (9):

$$RMSE = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{9}$$

Where x_i represents the observed value of the time interval between the i^{th} failure and the $(i\text{-}1)^{\text{th}}$ failure, and $\hat{x}_i$ represents the prediction of the corresponding failure interval.

3.3. Model Prediction

Dataset SS3 contains 278 failure interval data. Seven IMF sub-sequences and one residual trend term are obtained after decomposition. The decomposition results as shown in Figure 3:

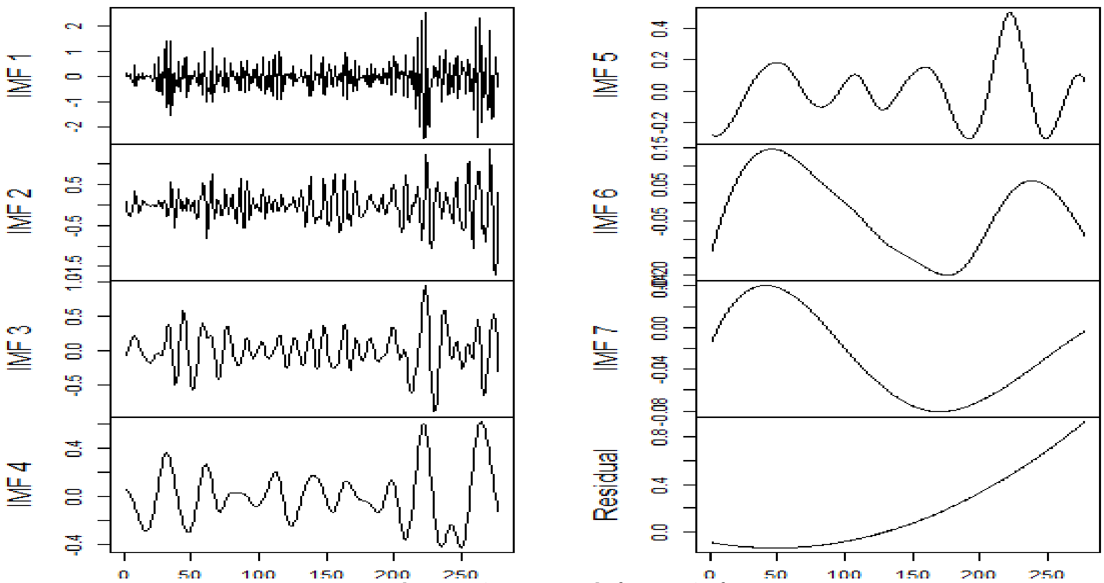


Figure 3. The IMFs (intrinsic mode functions) of EEMD

Each component in Figure 3 is treated as a component. The first 80% dataset of the component is used as the training set for modeling, and the corresponding model is established. The 20% dataset of the component is taken as the test set to verify the effectiveness of the proposed model. The SVR model is established for IMF1-IMF7, and the ARIMA model is established for the residual. Then, the predictions of IMFs subcomponents and the residual subcomponent are combined to obtain the predicted results of the original dataset SS3, as shown in Figure 4.

In Figure 4, the left figure shows the prediction results and residual schematic diagram of the original dataset SS3, and the right figure demonstrates the software accumulative running time and failure number schematic diagram. The predicted and observed values of the original failure interval component of the EEMD-SVR-ARIMA model proposed in this paper show that the proposed method has a good prediction effect.

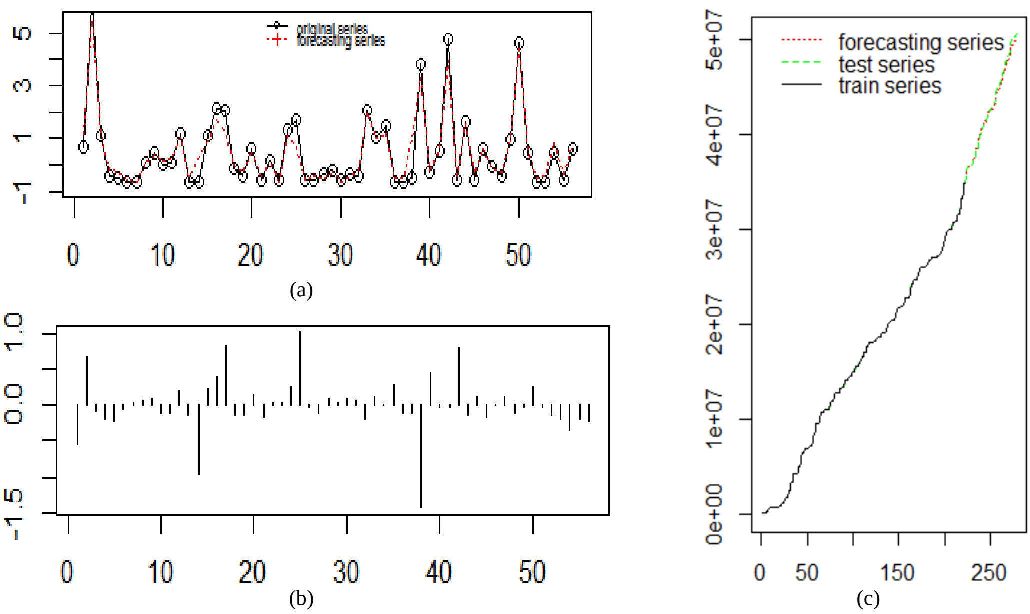


Figure 4. (a) Figure of forecasting; (b) Residual of forecasting result; (c) Comparison of original sequence and predicted result

3.4. Model Comparison

To further illustrate the effectiveness of the proposed method, the classical time series models such as Holt-Winters, ARIMA, MLR, and SVR are selected for comparison with the model proposed in this paper. The comparative analysis is divided into two parts: one compares the fitting and predicting effects of the same failure time interval dataset, and the other compares the fitting and prediction effects of the multiple sets of failure time interval datasets.

Table 1 demonstrates the RMSE comparison results fitted by each model, where RMSE_M represents the minimum of a single model, RMSE_C denotes the combined model, and Relative Error Reduced (RER) signifies the percentage reduction of ratio [21], indicating the rate of improvement of fitting or prediction accuracy. The formula is shown as Equation (10):

$$RER = \frac{RMSE_M - RMSE_C}{RMSE_M} \times 100\% \tag{10}$$

The comparative analysis results are shown in Table 1:

Table 2. The RMSE comparison results fitted by each model

Dataset	H-W	ARIMA	MLR	SVR	RMSE_M	RMSE_C	RER
P5	0.037	0.036	0.036	0.024	0.024	0.01	58.33%
SS1A	0.164	0.08	0.098	0.105	0.08	0.033	58.75%
SS1B	0.06	0.059	0.06	0.043	0.043	0.016	62.79%
SS1C	0.067	0.061	0.061	0.038	0.038	0.019	50.00%
SS2	0.084	0.076	0.073	0.059	0.059	0.026	55.93%
SS3	0.058	0.043	0.054	0.053	0.043	0.02	53.49%
SS4	0.085	0.07	0.07	0.056	0.056	0.025	55.36%

Table 1 shows that the RMSE of the combined model fit for the same failure dataset is significantly smaller than the other models. Overall, it also can be seen that the fitting effect of the combined model is significantly better than other models for the seven different failure time datasets. Figure 5 visually compares the RMSE size of each set of data fitted to each model. It shows that the combined model has the smallest RMSE in each set of data.

Similarly, Table 2 presents the RMSE comparison results predicted by each model. Table 2 indicates that the RMSE predicted by the combined model is significantly smaller than that predicted by other models when the same dataset of failure interval is predicted. As for the datasets with seven different failure time intervals, the RMSE predicted by the combined model is smaller than that of other models, and it can also be seen that the prediction effect of the combined model is significantly better than that of other models.

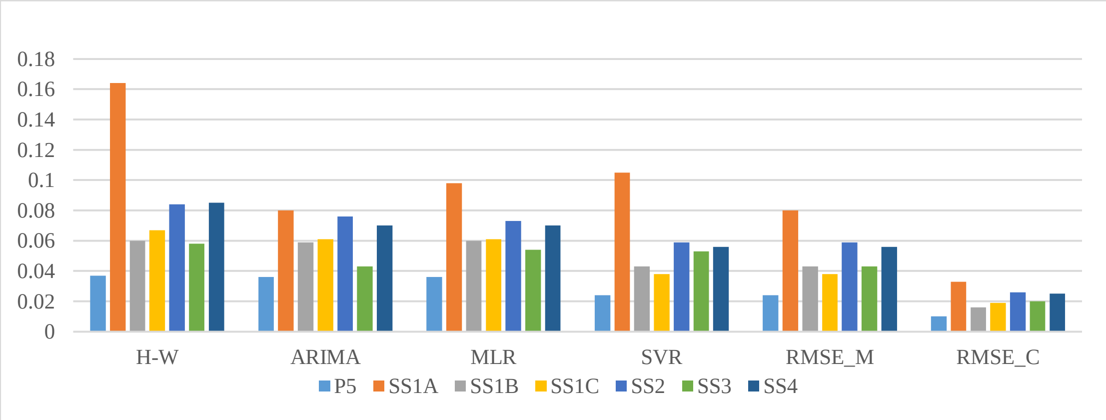


Figure 5. RMSE comparison for fitting results of each datasets

Table 3. The RMSE comparison results predicted by each model

Datasets	H-W	ARIMA	MLR	SVR	RMSE_M	RMSE_C	RER
P5	0.13	0.065	0.091	0.093	0.065	0.023	64.62%
SS1A	1.248	0.213	0.215	0.143	0.143	0.087	39.16%
SS1B	0.098	0.098	0.098	0.059	0.059	0.026	55.93%
SS1C	0.269	0.11	0.156	0.162	0.11	0.028	74.55%
SS2	0.239	0.116	0.197	0.208	0.116	0.07	39.66%
SS3	0.578	0.212	0.21	0.116	0.116	0.052	55.17%
SS4	0.381	0.176	0.223	0.221	0.176	0.073	58.52%

Figure 6 intuitively compares the RMSE size in each group of data predicted by each model. It displays that the RMSE of the combined model is the smallest in each group of data.

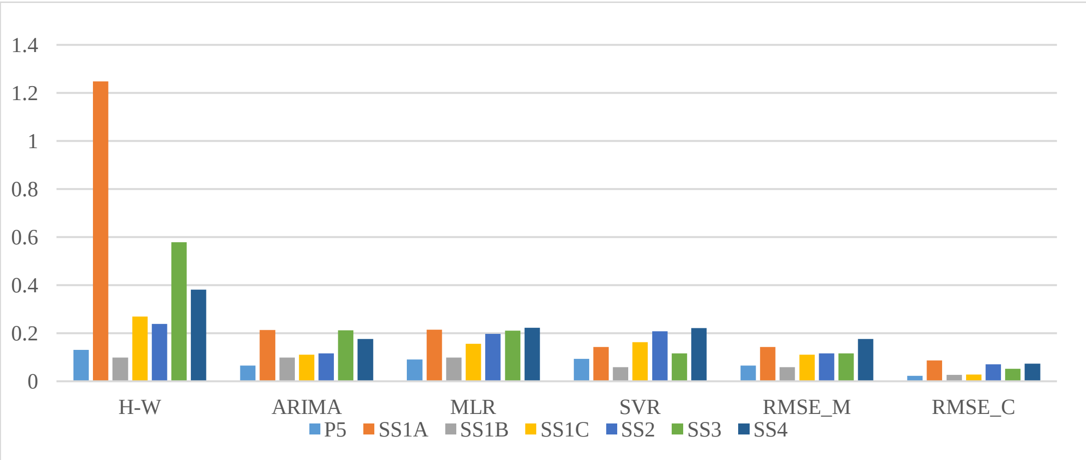
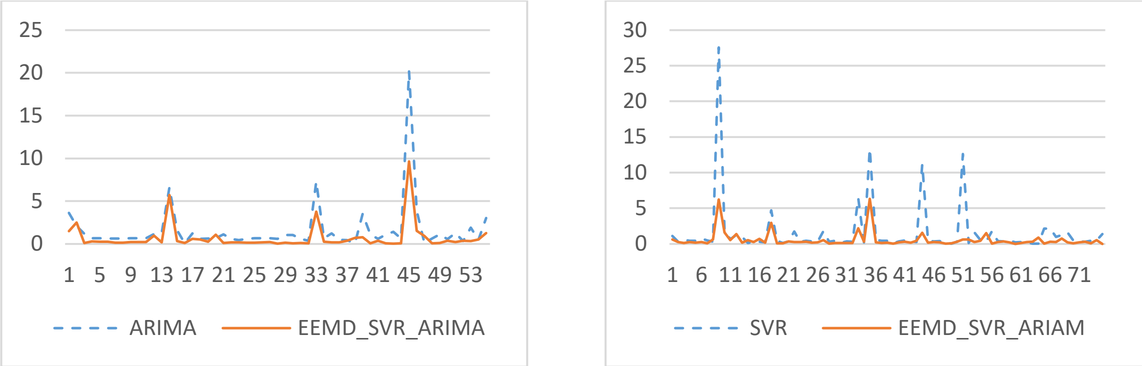


Figure 6. RMSE comparison for forecasting results of each datasets

To facilitate the further observation of the prediction effect of the model, Figure 7 shows the comparison of the prediction effect of the combined model and other optimal models on the failure data of each group. Relative error of failure (REF) is used to reflect the fitting effect of the model at each data point, and its expression is shown as Equation (11):

$$REF = \frac{|x(t) - \hat{x}(t)|}{x(t)} \tag{11}$$

In Figure 7, the REF of the combined model is stable around REF = 0 for datasets SSIC and SS1B, that is, the prediction effect is relatively stable. Moreover, the REF of the combined model is lower than that of other optimal models at most of the failure points, and this indicates the prediction effect is better.



(a) (b)

Figure 7. (a) Comparison figure of REF (SS1C); (b) Comparison figure of REF (SS1B)

The concentration trend and dispersion degree of the model REF are respectively reflected by mean value and variance, and the comparison results are shown in Table 3. Mean_c and var_c represent the mean and variance of the EEMD-SVR-ARIMA combination model REF respectively, and mean_m and var_m respectively represent the mean and variance of other optimal models.

Table 4. The central tendency and dispersion degree of REF of two datasets

Dataset	mean_c	mean_m	var_c	var_m
SS1C	0.278	0.869	1.543	2.881
SS1B	0.247	0.531	1.072	3.913

It can be seen from Table 3 that for the datasets SS1C and SS1B, the mean value of the combined model REF is slightly less than that of other optimal models, and the variance of the combined model REF is less than that of other optimal models. Therefore, through the graph analysis, the fitting effect of the combined model is better than that of the other optimal models.

As can be seen from Figure 8, for the dataset SS1B, the fitting results of the combined model overlap with the original failure data at most points.

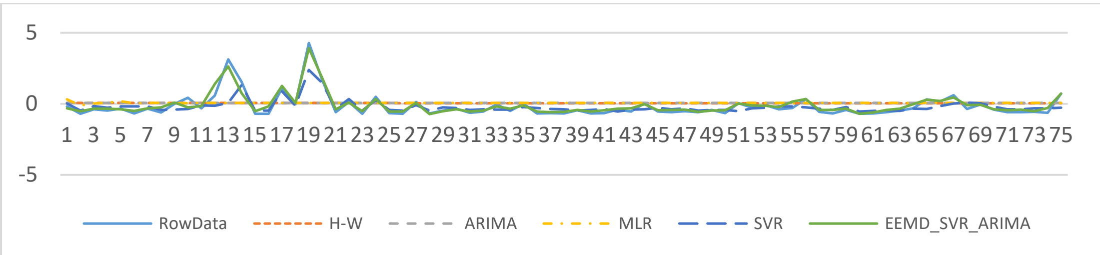


Figure 8. Schematic diagram of model prediction（SS1B)

4. Conclusions

This paper proposes a more comprehensive combination model based on EEMD and gives a model combination algorithm. A combined model based on EEMD decomposition is proposed for nonlinear and non-stationary series. This model decomposes the original complex failure sequence into components with different strong characteristics. Different forecasting models are established for sub-sequences with different characteristics, which overcomes the defect that a single model cannot fully capture the data’s characteristics. In addition, since the model has no obvious requirements for data characteristics and does not require pre-given basis functions, it is highly adaptive, so it is more widely applicable. Using this method for exposed datasets further proves its superiority. Most of all, this method is first used in the field of software reliability, which enriches the processing method of software reliability failure data.

Since the component prediction results are directly superimposed to obtain the final prediction results, the weighted combination of component prediction results can be considered in the future. In addition, when using the SVR model, the Gaussian basis function is used to transform nonlinear data into linear data. In the future, other basis functions, such as the spline basis function and trigonometric basis function, can be considered.

Acknowledgements

This research is supported by the Technology Foundation Program (No. JSZL2014601B008) of the National Defense Technology Industry Ministry.

References

1. M. Xie, K. L. Poh, and Y. S. Dai, “Computing System Reliability: Models and Analysis,” Springer Publishing Company, Incorporated, 2014
2. B. Yang, X. Li, M. Xie, and F. Tan, “A Generic Data-Driven Software Reliability Model with Model Mining Technique,” *Reliability Engineering and System Safety*, Vol. 95, No. 6, pp. 671-678, 2010
3. M. D. C. Moura, E. Zio, I. D. Lins, and E. Droguett, “Failure and Reliability Prediction by Support Vector Machines Regression of Time Series Data,” *Reliability Engineering and System Safety*, Vol. 96, No. 11, pp. 1527-1534, 2011
4. H. Li, M. Zeng, M. Lu, X. Hu, and Z. Li, “Adaboosting-based Dynamic Weighted Combination of Software Reliability Growth Models,” *Quality and Reliability Engineering International*, Vol. 28, No.1, pp. 67-84, 2012
5. R. Li and R. Kang. “Research on Failure Rate Forecasting Method based on ARMA Model,” *Systems Engineering and Electronics*, Vol. 30, No. 8, pp. 1588-1591, 2008
6. K. Xu, M. Xie, L. C. Tang, and S. L. Ho, “Application of Neural Networks in Forecasting Engine Systems Reliability,” *Applied Soft Computing*, Vol. 2, No. 4, pp. 255-268, 2003
7. J. L. Harvill and B. K. Ray, “A Note on Multi-Step Forecasting with Functional Coefficient Autoregressive Models,” *International Journal of Forecasting*, Vol. 21, No. 4, pp. 717-727, 2005
8. W. C. Wang, C. Kwokwing, D. M. Xu, and X. Y. Chen, “Improving Forecasting Accuracy of Annual Runoff Time Series using Arima based on Eemd decomposition,” *Water Resources Management*, Vol. 29, No. 8, pp. 2655-2675, 2015
9. V. Plakandaras, R. Gupta, and P. Gogas, “Forecasting the U. S. Real House Price Index,” *Economic Modelling*, pp. 259-267, 2015
10. M. Žvokelj, S. Zupan, and I. Prebil, “EEMD-Based Multiscale ICA Method for Slewing Bearing Fault Detection and Diagnosis,” *Journal of Sound and Vibration*, pp. 394-423, 2016
11. W. U. Zhaohua and N. E. Huang, “Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method,” *Advances in Adaptive Data Analysis*, Vol. 1, No. 1, pp. 1-41, 2011
12. V. Vapnik, “The Nature of Statistical Learning Theory,” in *Proceedings of Conference on Artificial Intelligence*, Springer-Verlag, pp. 988-999, 1995
13. G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, “Time Series Analysis: Forecasting and Control (Revised Edition),” *Journal of Marketing Research*, Vol. 14, No. 2, pp. 199-201, 1994
14. J. G. Lou and J. H. Jiang. “Correlation Analysis of Software Failure Time Data: Correlation Analysis of Software Failure Time Data,” *Journal of Computer Applications*, Vol. 30, No. 3, pp. 600-602, 2010
15. X. L. Liu, D. X. Jia, L. I. Hui, and J. Y. Jiang, “Research on Kernel Parameter Optimization of Support Vector Machine in Speaker Recognition,” *Science Technology and Engineering*, 2010
16. C. L. Wu, K. W. Chau, and Y. S. Li, “River Stage Prediction based on a Distributed Support Vector Regression,” *Journal of Hydrology*, Vol. 358, No. 1, pp. 96-111, 2008
17. D. Bratton and J. Kennedy, “Defining a Standard for Particle Swarm Optimization,” in *Proceedings of IEEE Swarm Intelligence Symposium*, pp. 120-12, IEEE Computer Society, 2007
18. P. F. Pai and W. C. Hong, “Software Reliability Forecasting by Support Vector Machines with Simulated Annealing Algorithms,” *Journal of Systems and Software*, Vol. 79, No. 6, pp. 747-755, 2006
19. V. Cherkassky and Y. Ma, “Practical Selection of SVM Parameters and Noise Estimation for SVM Regression,” *Neural Networks the Official Journal of the International Neural Network Society*, Vol. 17, No. 1, pp. 113-126, 2004
20. J. D. Musa, “Software Reliability Data,” Technical Report in Rome Air Development Center, 1979
21. H. Sun, L. Zhang, J. Wu, J. Wu, and H. Yang, “A New Method of Model Combination based on the NHPP Software Reliability Models,” in *Proceedings of International Conference*, pp. 153-158, 2018

Sun Haiyan is an associate professor of the School of Mathematics and Systems Science at Beijing University of Aeronautics and Astronautics. He received his Ph.D. from Nankai University in 1997, and he worked as a postdoctoral researcher in the Department of Statistics at Renmin University of China from 1997 to 2000. He focused on probability statistics. His research interests include statistical techniques, statistical model diagnosis, and statistical data analysis in software reliability analysis.

Wu Jing is currently pursuing her Master’s degree at Beijing University of Aeronautics and Astronautics. She received her

B.S. degree from Capital Normal University in 2016. Her research is focused on software reliability analysis based on time series.

Wu Ji is an associate professor and assistant dean of the School of Computer Science and Engineering (SCSE) at Beijing University of Aeronautics and Astronautics. He received his Ph.D. from Beijing University of Aeronautics and Astronautics in 2003 and his M.S. degree from the Second Research Institute of the China Aerospace Science and Industry Group in 1999. He focuses on industry-oriented research. His research interests include embedded systems and software modeling and verification, software requirements and architecture modeling and verification, safety and reliability assessment, and software testing.

Yang Haiyan is a lecturer at Beijing University of Aeronautics and Astronautics. Her main research interests include software engineering, software safety, and software testing.