Improved Hybrid Feature Selection Framework

Weizhi LIAO^{†a)}, *Member*, Guanglei YE^{†b)}, Weijun YAN^{†c)}, *Nonmembers*, Yaheng MA^{†d)}, *Member*,
and Dongzhou ZUO^{†e)}, *Nonmember*

SUMMARY An efficient Feature selection strategy is important in the dimension reduction of data. Extensive existing research efforts could be summarized into three classes: Filter method, Wrapper method, and Embedded method. In this work, we propose an integrated two-stage feature extraction method, referred to as FWS, which combines Filter and Wrapper method to efficiently extract important features in an innovative hybrid mode. FWS conducts the first level of selection to filter out non-related features using correlation analysis and the second level selection to find out the near-optimal sub set that capturing valuable discrete features by evaluating the performance of predictive model trained on such sub set. Compared with the technologies such as mRMR and Relief-F, FWS significantly improves the detection performance through an integrated optimization strategy. Results show the performance superiority of the proposed solution over several well-known methods for feature selection.

key words: machine learning, feature selection, hybrid, filter mode, wrapper mode

1. Introduction

With the rapid development of artificial intelligent based applications such as sales prediction, Natural Language Processing (NLP), medical system and other domains [1]–[5], building a robust predictive model using noisy data in a huge volume is a challenging task and of great importance to various domains. In general, the quality of data and features determines the upper bound of the performance of corresponding model trained on it. Predictive models are designed to approximate such upper bound by tuning its hyper parameters. However, considering the curse of dimension, high dimensional noisy data with unknown unrelated and redundant features could pose unforeseeable impact on the performance of prediction models. Hence, design an efficient strategy to perform dimension reduction [6], [7] is of great importance to build a robust machine learning model. How to select out features (Feature selection) is one of the fundamental problems in machine learning which could be described as designing a methodology to extract most significant sub set of features [8], and in the same time, quickly

reduce the massive data by compressing the full feature spaces and improve the performance of corresponding models/algorithms trained on such subset.

Several feature selection methods exemplified by filter mode customize an evaluation metric for the relation of feature vector to response vector. In filter mode based strategy, the key idea is to define a generalizable score function, corresponding score of each feature vector against response vector is calculated, and make the final decision based on setting a threshold of the score. Another line of research efforts could be categorized as model dependent algorithms, including wrapper method and embedded method, which aiming at choosing a sub set of features that contributes the most to the predictive model. The wrapper method select features by evaluating the performance of predictive model trained on such sub set. Embedded methods, in general, for example LASSO based method, select features while the model is being created. These traditional methods have won a good reputation with a satisfactory performance but are generally causing huge information loss and very computational expensive due to the complexity of data structure and volume of feature size. For example, in the filter mode, influence of features are evaluated separately and the combined affects of features are ignored.

In this paper, we propose an integrated solution, referred to as FWS, to perform the feature selection through combining filter mode and wrapper mode in a novel way. FWS conducts the first level of selection, where the covariance within original feature space and feature space to response vector are evaluated, to filter out non-related features by thresholding the score of correlation evaluation metric, and the second level selection that aiming to find the approximate optimal sub set that capturing valuable features by evaluating the performance of predicting model based on such random selected sub set. Compared with traditional feature selection methods, FWS significantly reduces the computation complexity and improves the detection performance through an integrated optimization strategy. Extensive simulation results illustrate the performance superiority of the proposed integrated solution over several commonly used methods for feature selection in practice.

The rest of this paper is organized as follows. Section 2 conducts a survey of related work. Section 3 proposes the FWS detection method with the integration of sensor deployment and data fusion. Section 4 evaluates the performance of FWS through simulations. Section 5 concludes

Manuscript received November 11, 2020.

Manuscript revised January 31, 2021.

Manuscript publicized May 12, 2021.

[†]The authors are with the University of Electronic Science and Technology of China, ChengDu, China.

a) E-mail: liaoweizhi@126.com

b) E-mail: 201821040125@std.uestc.edu.cn

c) E-mail: 799110106@qq.com

d) E-mail: 201821040126@std.uestc.edu.cn

e) E-mail: zuodongzhou@outlook.com

DOI: 10.1587/transinf.2020BDP0017

our work.

2. Related Work

Feature selection is one of the most important problems in machine learning. In dealing with data set with large number of features that containing meaningless or irrelevant features, it is, often times, advantageous to design a feature selection algorithm to extract a subset of features to improve the performance of the predictive models. Existing feature selection methodologies could be mapped into three classes: *Filter method*, *Wrapper method*, *Embedded method*. Under the category of filter mode, Kiral [9] proposed an euclidean distance based feature extraction method, referred as Relief and further carry out Relief-F [10] to address multi-classification problem. Robnik [11] summarize the effectiveness of hyper parameters to the performance of Relief based models by running extensive experiments. Despite of the high accuracy the relief based models have achieved, such models randomly sample data from original data set as training data which add randomness into model building and hence may result in a sub-optimal model. To address such drawback, Liu [12] integrated sampling algorithm with Relief and achieved comparable accuracy using partial data compared with using entire data set. Qian [13] proposed a consistency metric based theoretical framework, referred as Positive Approximation, which can accelerate the speed of extraction of feature subsets, but the performance of the model is not improved. Peng [14] proposed a minimum redundancy maximization correlation method (mRMR), which selects most relevant and least redundant features by evaluating the correlation between features and response vectors and the degree of redundancy between different features. In the category of Wrapper mode, Jarvis [4] employ the genetic algorithm to the decision of feature extraction over the Fourier transformed infrared spectral data, and the predictive model trained with such subset achieved high performance. Huang [15] and Cai [16] used information theory to extract feature subsets and significantly improved the performance of the predictive model. In the category of Embedded mode, Quilan [17] proposed a decision tree based algorithm. Feature selection and predictive model building will accomplish in the same time, since during the node splitting process of building decision tree, the information gain of each feature will be calculated and further used to select features. Guyon [18] designed a SVM based feature selection model, referred to as *SVM – REF*, which utilize the hyper parameters of SVM, such method has been widely used in the field of genetic representation learning. Besides from traditional model that focus on exploiting single methodology, another research line conforming to hybrid design has also been well studied. Li [17] proposed a method to concatenate Filter mode and Wrapper mode to filter out potential relevant features using information gain of each features and then use improved binary particle swarm optimization as a wrapper approach to select subset of features to train predictive model. Our work differs from the

aforementioned efforts in that we state the rational behind the integration and propose an innovative methodology to construct a robust framework.

3. Hybrid Feature Selection Framework

We propose a two-stage hybrid solution, referred to as FWS, which integrates filter mode and wrapper mode for the objective of selecting influential features. We first present the overall structure of *FWS*, then detail the design and rational behind the choice of methods for filter mode and wrapper mode.

3.1 Design of FWS

Filter mode based strategies are widely adopted to feature selection due to the calculation efficiency. However, it limits to measuring only the influence of continuous variables. On the other hand, wrapper mode based methodologies utilize the performance of predictive model at cost of high computational complexity. We proposed a new method that integrate the merits of filter mode and wrapper mode to take into account the affects of continuous features, discrete features, combined features in the same time.

Our feature selection framework can be divided into two stages as shown in Fig. 1. In the first stage: We utilize the filter mode to perform first level feature extraction to improve the calculation efficiency. More specifically, we measure both the standard deviation and the Pearson correlation coefficient to evaluate the divergence and relevance of single continuous features and redundancy within features, which in turn improve extraction efficiency and reduce the dimensionality. In the second stage: Based on the selection result of filter stage, we concatenate the wrapper mode to improve the extraction accuracy. In particular, we repeatedly select a sub set of features for predefined number of iterations, and comparing the performance of predictive models including K-nearest neighbour (KNN) and Linear regression that trained on each subset, and output the final decision with highest accuracy.

In this paper, the experimental results are compared with the existing models, and the results show that the algorithm is having a demonstrable effect on feature extraction. Therefore, the proposed algorithm is innovative.

3.2 Designation of First Stage Feature Selection

We propose to utilize standard deviation and Pearson correlation coefficient as our filter methods as shown in Fig. 2. Implementation details are list as follows:

3.2.1 Standard Deviation Measurements

In this step, we filter out less informative features by thresholding the standard deviation of each feature. The standard deviation is calculated as:

Fig. 1 Integrated frame work

Fig. 2 Design of first stage

$$u_x = \frac{\sum_{i=1}^n X_i}{n},$$

$$S^2 = \frac{\sum_{i=1}^n (X_i - u_x)^2}{n},$$

(1)

(2)

olding the Pearson value between features and response vector P_{ff} . We then eliminate the redundant features by thresholding the Pearson value of each pair of feature P_{ff} . The P value is calculated as:

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - u_x)(Y_i - u_y)}{n - 1}, \quad (3)$$

$$P_{X,Y} = \frac{cov(X, Y)}{\theta_X \theta_Y}, \quad (4)$$

3.2.2 Pearson Correlation Coefficient Measurements

In this step, we first filter out less relevant features by thresh-

Fig. 3 Design of second stage

3.2.3 Normalization

In this step, we perform normalization to balance the feature importance.

3.3 Designation of Second Stage Feature Selection

In this stage, we further prune potential unrelated features by deploying wrapper mode. As shown in Fig. 3, we first set the upper bound for size of feature sub set as F_{max} , where $F_{max} = 30$ if number of overall features FS is greater than 100, otherwise F_{max} equals to $\alpha * FS$, $1/3 \leq \alpha \leq 1/2$. Instead of performing brute force search, we then set the termination condition of iteration trail as a finite number T and adopt heuristic algorithm to select subset for each iteration, which help us find the near optima result. We finally propose to use KNN and linear regression to server as classifier and regressor in the wrapper mode correspondingly. The merits of such designation are two-fold: i) over fitting avoidance: we limited the number of features selected out from original feature space. ii) computation efficiency: instead of brute force search, we set the termination condition of iteration trail as a finite number T and adopt heuristic algorithm to select subset, which help us find the near optima subset in limited iterations.

3.4 Merged Framework

We then propose an innovative method to merge filter mode and wrapper mode as shown in Fig. 4. The key steps of the proposed FWS framework are provides in Alg. 2

In step 1, we initialize the standard deviation SD threshold as V , constant v to determine the increasing step size of SD and p to control the increasing step size of Pearson value. And initialize Pearson thresholds P_{ff} and P_{ft} , a constant T as iteration number to limit the sampling times.

In step 2, we filter out less important features by measuring the value of SD .

In step 3, we drop out less related features by calculating the Pearson value of each features and response vector, and comparing it with predefined threshold.

In step 4, we further remove redundant features by calculating the Pearson value for each pair of features and comparing it with predefined threshold.

In step 5, we normalize the data set to avoid imbalance feature weight, and initialize iteration index t , optimal subset H^* till current iteration, S to store the score returned by regressor/classifier and N to store the feature size.

In step 6, we perform stop-criteria check, if number of iterations exceeds our predefined value, we stop iterative searching and jump to step 11.

In step 7, we random sample feature subset H , and evaluate the score returned by predictive model.

Fig. 4 Overall design

Algorithm 1 FWS

Input: data set with features set $\{X\}$ and response vector Y
Output: reduced data set with feature set $\{X'\}$.

- 1: Initialize the SD threshold as V , step size v and p , Pearson thresholds P_{ff} and P_{ft} , constant T , $X' = \phi$.
- 2: For each feature $k \in X$, compute standard deviation SD_k if $SD_k < V$, $X = X - k$.
- 3: Calculate the P value P_{ft}^i for each pair of features (i, Y) , $i \in X$, if $P_{ft}^i < P_{ft}$, $X = X - i$.
- 4: Calculate the P value for each pair of features (j, m) , $j, m \in X$, if $P_{ff}^{jm} < P_{ff}$, calculate P_{ft}^j and P_{ft}^m . if $P_{ft}^j > P_{ft}$, $X = X - j$, else $X = X - m$.
- 5: Normalize X and set $t=0$, $S=0$, $N=0$, $H^* = \phi$.
- 6: if $t > T$ go to step 11.
- 7: Random sample feature sub set H with size N , compute the performance S_H of predictive model trained on H .
- 8: if $S_H > S \wedge n < F_{max}$, go to step 10.
- 9: if $S_H == S \wedge n < N$: go to step 10, else: $t = t + 1$, go to step 6.
- 10: Update: $t=0$, $H^* = H$, $S = S_H$, $N = n$.
- 11: if $H^* = \phi$: $V = V + v$, $P_{ft} = P_{ff} + p$, $P_{ft} = P_{ff} - p$, go to step 2.
- 12: return H^*

In step 8 and 9, we compare the score of current subset H with H^* , if predictor score of H outperforms H^* and the size of H is less than threshold F_{max} or if score of H and H^* are equal and feature size of H is smaller than H^* , we continue to next step. Otherwise, we increment the timer t with 1 and go to step 6.

In step 10, we reset the timer and replace H^* with H .

In step 11, if the optima set is empty we adjust the threshold and go to step 2. Otherwise, return the result.

4. Experimental Settings and Results

In this section, we first describe the experimental settings, then we derive a customized evaluation metric and further present prediction results using various machine learning models provided by scikit-learn library.

4.1 Experimental Settings

We set the threshold of standard deviation $V = 10$, P value threshold P_{ft} as 0.1 and P_{ff} as 0.9, step size $v=1$, $F_{max} = 7$, iterations $T = 100$, classifier model as KNN and regression model as Linear Regression. We use machine learning models provided by scikit-learn with default setting. We deploy a 5-fold cross validation strategy that divide training dataset into five equal size dataset, and conducted 5 runs, for each run of the experiments, we select one part out as testing and remaining as training. We present the average performance of 5 runs.

4.2 Evaluation Metric

According to the existing research literature, the performance of the feature extraction algorithm is evaluated by the classification accuracy of the feature subset extracted by the feature extraction algorithm in the classification model. In order to systematically compare the performance of feature extraction algorithm, this paper defines ave, AvF and AEV as the evaluation criteria of feature extraction algorithm, and designs a customizable evaluation index as follow:

$$AEV = \frac{Ave}{AvF}, \quad (5)$$

Table 1 Data set selected from UCI

Name	samples	features	categories
Glass	214	9	7
Lymphography	148	18	4
Spectf	267	44	2
Muli-feature zer	2000	47	10
Musk (Version 2)	6598	166	2
Muli-feature fac	2000	240	10
Muli-feature pixel	2000	240	10
Arrhythmia	452	279	16

where AEV represents the extraction efficiency of our framework, Ave is the average prediction performance of our framework on several benchmark datasets and AvF is the average size of features subtracted.

4.3 Data Set

To evaluate the robust of our integrated method, we sampled several datasets containing diversity categories of features from UCI [19] as shown in Table 1, a well known benchmark machine learning repository.

4.4 Model in Comparison

In the experiment, two classical feature extraction algorithms based on filter model, mRMR and Relief-F, are selected to compare with this algorithm. That is, the core idea of mRMR algorithm is to measure the maximum correlation between feature variables and predicted targets, and to predict the minimum redundancy between different features, while Relief-F is the representative of feature extraction algorithm in multi-classification problem, so this paper chooses to compare with these two algorithms.

4.4.1 mRMR

The core idea of mRMR is to maximize the correlation between features and classification variables while minimizing the correlation between features. Metrics use Mutual information.

Definition: given the probability density function of two random variables X and Y are $p(x)$, $p(y)$, and $p(x,y)$, then the mutual information is:

$$I(x; y) = \int \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy \quad (6)$$

Finally, the feature subset S with m features was found, which should meet the following two conditions:

(1). Maximum relevance:

$$\max D(S, c), D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i; c) \quad (7)$$

(2). Minimum redundancy:

$$\min R(S), R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i; x_j) \quad (8)$$

4.4.2 Relief-F

Relief-F is a feature weighting algorithm that assigns features different weights according to the relevance of each feature and category. When dealing with multiple types of problems, randomly takes a sample R from the training sample set, then finds out the near Hits of R from the sample set of the same category as R , and the near Misses of R from the sample set of different categories of R . It updates the weight of each feature as follow:

$$W(A) = W(A) - \sum_{j=1}^k \text{diff}(A, R, H_j) / (mk) + \sum_{C \in \text{class}(R)} \left[\frac{p(C)}{1 - p(\text{Class}(R))} \sum_{j=1}^k \text{diff}(A, R, M_j(C)) \right] / (mk) \quad (9)$$

In the above formula, $\text{diff}(A, R_1, R_2)$ represents the difference between sample R_1 and sample R_2 on feature A . M represents the j th nearest neighbor sample in class C . As shown in the following formula:

$$\text{diff}(A, R_1, R_2) = \begin{cases} \frac{|R_1[A] - R_2[A]|}{\max(A) - \min(A)} & A \text{ is continuous} \\ 0 & A \text{ is discrete } R_1[A] = R_2[A] \\ 1 & A \text{ is discrete } R_1[A] \neq R_2[A] \end{cases} \quad (10)$$

The Relief-F algorithm execution process is as Algorithm 2.

Algorithm 2 Relief-F algorithm

Input and parameters: The training data set $\{D\}$, the number of sample sampling $\{m\}$, the threshold value of feature weight $\{o\}$, the number of nearest neighbor samples $\{k\}$.

Output: The feature weight $\{T\}$ of each feature.

- 1: Let all the feature weights be 0. T is the empty set.
 - 2: For $i=1$ to m do
 - 1) Pick a random sample R from D .
 - 2) Find k nearest neighbors $H_j (j = 1, 2, \dots, k)$ of R from the homogeneous sample set of R , and k nearest neighbors $M_j(C)$ from each heterogeneous sample set.
 - 3: For $A=1$ to N All feature do

$$W(A) = W(A) - \sum_{j=1}^k \text{diff}(A, R, H_j) / (mk) + \sum_{C \in \text{class}(R)} \left[\frac{p(C)}{1 - p(\text{Class}(R))} \sum_{j=1}^k \text{diff}(A, R, M_j(C)) \right] / (mk)$$
-

4.5 Results

Our performance evaluation includes two parts: i) compare the performance of the predictive models that adopting different feature selection strategies. ii) compare the performance of different features selection methods using custom evaluation metric.

Fig. 5 Performance comparison when adopting KNN as classifier

Fig. 7 Performance comparison when adopting Support Vector Machine (SVM) as classifier

Fig. 6 Performance comparison when adopting Naive Bayes as classifier

Table 2 Comparing feature selection methods

classifier	feature method	KNN			NB			SVM		
		FWS	mRMR	Relief-F	FWS	mRMR	Relief-F	FWS	mRMR	Relief-F
Ave		82.58%	80.77%	78.98%	83.84%	79.54%	75.34%	84.15%	82.72%	82.11%
AvF		14	15	15	12	13	12	15	16	15
AEV		5.90%	5.38%	5.27%	6.99%	6.12%	6.28%	5.61%	5.17%	5.47%

4.5.2 Comparing Using Custom Metric

To further illustrate the superior of our framework, we compared the performance of different feature selection methods on our custom evaluation metric. As shown in Table 2, our framework has the maximum *Ave* and *AEV* compared to other methods.

As can be seen from the Table 2, in the process of feature selection, the *AEV* index of FWS is higher than that of mRMR and Relief-F methods, indicating that FWS can effectively obtain data features.

5. Conclusion

In this work, we summarize the pros and cons of wrapper mode and filter mode based feature selection strategy. And we then propose an integrated solution to merge the merits of these two modes. Experimental results show that our framework achieved significantly higher accuracy in comparison with several other state-of-the-art methods in terms of various performance evaluation criteria. This method can obtain the important features of the data more accurately, reduce the dimension of the data to a large extent, and facilitate the later data analysis. In the future work, feature selection methods other than mRMR and Relief-F methods will be considered to better be used for data dimensionality reduction.

Acknowledgments

This work was supported by the National Key R & D Program of China (2017YFB1400301 & 2017YFB1400300).

4.5.1 Comparing Predict Accuracy

We first select Naive Bayes, K-Nearest Neighbours, Support Vector Machine (SVM) as our candidate evaluation predictive model and then compare the performance of these models that trained on subset features selected by our framework and other state-of-art methods. As we can see from Fig. 5, Fig. 6, Fig. 7, our framework obviously outperforms the traditional mRMR and Relief-F in all of the datasets and machine learning models.

As can be seen from figure 5, when the KNN algorithm classifier is used for performance comparison, the feature subset extracted by our framework has the highest classification accuracy in the corresponding data set.

Figure 6 shows the classification performance comparison using Naive Bayes classifier. It is found that in each different data set, the classification accuracy of the feature subset extracted by our framework in the corresponding data set is significantly higher than that of the other two feature extraction algorithms.

Figure 7 shows the classification performance comparison using SVM classifier. We can find that in Multi-feature zer data set and Multi-feature zer data set, the classification accuracy of the feature subset extracted by the two-stage extraction algorithm is more higher than the other two feature extraction algorithms.

References

- [1] D. Sisiaridis and O. Markowitch, "Reducing data complexity in feature extraction and feature selection for big data security analytics," 2018 1st International Conference on Data Intelligence and Security (ICDIS), pp.43–48, IEEE, 2018.
- [2] H. Ogura, H. Amano, and M. Kondo, "Comparison of metrics for feature selection in imbalanced text classification," *Expert Systems with Applications*, vol.38, no.5, pp.4978–4989, 2011.
- [3] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol.23, no.19, pp.2507–2517, 2007.
- [4] R.M. Jarvis and R. Goodacre, "Genetic algorithm optimization for pre-processing and variable selection of spectroscopic data," *Bioinformatics*, vol.21, no.7, pp.860–868, 2004.
- [5] F.P. Shah and V. Patel, "A review on feature selection and feature extraction for text classification," 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp.2264–2268, IEEE, 2016.
- [6] G.A. Susto and A. Beghi, "Dealing with time-series data in predictive maintenance problems," 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), pp.1–4, IEEE, 2017.
- [7] G.A. Susto, A. Schirru, S. Pampuri, and S. McLoone, "Supervised aggregative feature extraction for big data time series regression," *IEEE Transactions on Industrial Informatics*, vol.12, no.3, pp.1243–1252, 2016.
- [8] R.P.S. Manikandan and A. Kalpana, "A study on feature selection in big data," 2017 International Conference on Computer Communication and Informatics (ICCCI), pp.1–5, IEEE, 2017.
- [9] K. Kira and L.A. Rendell, "A practical approach to feature selection," *Machine Learning Proceedings 1992*, pp.249–256, Elsevier, 1992.
- [10] I. Kononenko, "Estimating attributes: analysis and extensions of relief," *European conference on machine learning*, vol.784, pp.171–182, Springer, 1994.
- [11] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relief and rrelief," *Machine learning*, vol.53, no.1-2, pp.23–69, 2003.
- [12] H. Liu, H. Motoda, and L. Yu, "A selective sampling approach to active feature selection," *Artificial Intelligence*, vol.159, no.1-2, pp.49–74, 2004.
- [13] Y. Qian, J. Liang, W. Pedrycz, and C. Dang, "An efficient accelerator for attribute reduction from incomplete data in rough set framework," *Pattern Recognition*, vol.44, no.8, pp.1658–1670, 2011.
- [14] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol.27, no.8, pp.1226–1238, 2005.
- [15] J. Huang, Y. Cai, and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," *Pattern Recognition Letters*, vol.28, no.13, pp.1825–1844, 2007.
- [16] R. Cai, Z. Hao, X. Yang, and W. Wen, "An efficient gene selection algorithm based on mutual information," *Neurocomputing*, vol.72, no.4-6, pp.991–999, 2009.
- [17] J.R. Quinlan, C4. 5: programs for machine learning, Elsevier, 2014.
- [18] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol.46, no.1-3, pp.389–422, 2002.
- [19] A. Frank, "Uci machine learning repository," <http://archive.ics.uci.edu/ml>, 2010.

Weizhi Liao is an Associate Professor in the Industrial Engineering Department of the School of Mechanical and Electrical Engineering at the University of Electronic Science and Technology of China. She was awarded her Ph.D. at the School of Information Science and Technology from Southwest Jiaotong University in 2010. She has a core background on algorithms and data structures and her main research line is focused on large-scale data driven and service.

Guanglei Ye is currently a postgraduate student in the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, ChengDu, China. His current research interests include deep learning technology and Natural Language Processing.

Weijun Yan is currently a postgraduate student in the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, ChengDu, China. His current research interests include deep learning technology and Natural Language Processing.

Yaheng Ma is currently a postgraduate student in the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, ChengDu, China. His current research interests include deep learning technology and Natural Language Processing.

Dongzhou Zuo is currently a postgraduate student in the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, ChengDu, China. His current research interests include deep learning technology and Natural Language Processing.