Proceedings of the 2018 International Conference on Machine Learning and Cybernetics, Chengdu, China, 15-18 July, 2018

IMBALANCED DATA CLASSIFICATION BASED ON EXTREME LEARNING MACHINE AUTOENCODER

CHU SHEN[1], SU-FANG ZHANG[2,*], JUN-HAI ZHAI[1], DING-SHENG LUO[3], JUN-FEN CHEN[1]

[1]Key Lab. of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding, 071002, Hebei, China
[2]Hebei Branch of China Meteorological Administration Training Center, China Meteorological Administration，Baoding 071000, China
[3]Key Lab. of Machine Perception (Ministry of Education), Speech and Hearing Research Center Department of Machine Intelligence, School of EECS, Peking University, Beijing 100871, China
E-MAIL: mczsf@126.com

Abstract:

In practice, there are many imbalanced data classification problems, for example, spam filtering, credit card fraud detection and software defect prediction etc. it is important in theory as well as in application for investigating the problem of imbalanced data classification. In order to deal with this problem, based on extreme learning machine autoencoder, this paper proposed an approach for addressing the problem of binary imbalanced data classification. The proposed method includes 3 steps. (1) the positive instances are used as seeds, new samples are generated for increasing the number of positive instances by extreme learning machine autoencoder, the generated new samples are similar with the positive instances but not same. (2) step (1) is repeated several times, and a balanced data set is obtained. (3) a classifier is trained with the balanced data set and used to classify unseen samples. The experimental results demonstrate that the proposed approach is feasible and effective.

Keywords:

Imbalanced data classification; Extreme learning machine; Autoencoder; Generative model

1. Introduction

In practice, there are many imbalanced data classification problems, such as the problem of spam filtering, the problem of credit card fraud detection and the problem of software defect prediction etc. The problem of learning from imbalanced data is very challenging, which has attracted growing attention from fields of machine learning and data mining [1-4]. In this paper, we investigate the imbalanced learning problem in the framework of binary classification. In binary imbalanced classification, the proportion of instances belonging to one class (negative class) is very much higher than another class (positive class) [5, 6]. The positive class and negative class are denoted by S^+ and S^- respectively in this paper. Balancing is a simple and effective method to solve the problem of imbalanced learning, the balancing approaches can be roughly classified into two categories: undersampling and oversampling. In the undersampling, a subset S_u^- with size of $|S^+|$ is randomly selected from S^-, a balanced data set can be obtained by merging S_u^- and S^+. In the oversampling, some randomly generated instances are added to the set S^+, so that the size of S^+ is increased. The SMOTE (Synthetic Minority Oversampling TEchnique) [7] is the most representative oversampling method.

Relatively, compared with random undersampling, the random oversampling is more widely used in binary imbalanced learning. A novel class imbalance metric called GIR (generalized imbalance ratio) was defined by Tang and He [3], and based on this new metric, they proposed two sampling approaches which adaptively split the imbalanced learning problem into multiple balanced learning subproblems in a probabilistic way. Based on self-organizing map (SOM), an oversampling method was proposed by Douzas and Bacao [8]. The proposed method uses SOM to generate a two-dimensional representation of the input space, and based on this representation, artificial data points can be effectively generated. In [9], based on support vector machine (SVM), two oversampling methods were proposed by Piri et al. The one is called SIMO (synthetic informative minority over-sampling), the other is called W-SIMO (Weight SIMO). In the SIMO, SVM is firstly applied to the original imbalanced dataset, and then, positive class examples which are close to the decision boundary of SVM as the informative minority examples are over-sampled. In W-SIMO, incorrectly classified informative positive examples are over-sampled with a

978-1-5386-5214-5/18/$31.00 ©2018 IEEE

higher degree compared to the correctly classified informative positive examples. Based on Wiener process, an oversampling approach which brings the physics phenomena into sample synthetization was proposed by Li et al. [10], the proposed method constructs a robust decision region by expanding the attribute ranges in training set while keeping the same normal distribution. Motivated by the localized generalization error model [11], an imbalanced data classification method based on ensemble learning was proposed by Chen et al. [12], the proposed approach generates some synthetic samples located within some local area of the training samples and trains the base classifiers with the union of original training samples and synthetic neighborhoods samples. Zhai et al. [5] proposed an oversampling method which generates positive samples within their enemy nearest neighbor hyperspheres. Vanhoeyveld and Martens [13] experimentally compared the performance of the commonly used imbalanced learning strategies on sparse and large behavior datasets, including oversampling strategy, and obtained some valuable conclusions. For example, they found that oversampling techniques show a good overall performance and do not suffer from overfitting, and the EasyEnsemble technique [14] outperforms all others on sparse and large behavior datasets. Amin et al. [15] viewed the customer churn problem as an imbalanced learning problem and made a comparative study on various oversampling techniques by the customer churn prediction case study. Douzas and Bacao [16] applied the conditional generative adversarial networks [17-20] to approximate the true data distribution and generate data for the positive class of various imbalanced datasets and obtained promising performance.

Motivated by the idea of autoencoder [21-23], based on extreme learning machine autoencoder [24], this paper proposed an approach for classifying binary imbalanced data. The proposed method uses positive instances as seeds, and uses the extreme learning machine autoencoder to generate positive instances which are different from the seeds. The paper is organized as follows. The preliminaries used in this paper are presented in Section 2. The proposed method is presented in Section 3, the experimental results are given in Section 4. Section 5 concludes this paper.

2. Preliminaries

In this section, we will briefly review the extreme learning machine (ELM) which will be in this paper.

ELM [25] is a simple and effective random algorithm which is tailored for training single hidden layer feed-forward neural networks (SLFNs). In ELM, the weights between input layer and hidden layer and the biases of hidden nods are randomly assigned, while the weights between hidden layer and output layer are analytically determined. The architecture of a SLFN can be described by a triple (d, m, k), where d is the number of input layer nodes, i.e. the dimension of input data, m is the number of hidden nodes and k is the number of output layer nodes, i.e. the number of classes of input data. Given a training set $D=\{(x_i, y_i) \mid x_i \in R^d, y_i \in R^k\},\ 1 \le i \le n$, the SLFN with structure (d, m, k) can be modeled by the following equation (1).

$$f(x_i)=\sum_{j=1}^{m}\beta_j g\left(w_j \cdot x_i + b_j\right) \tag{1}$$

where β_j is the weight vector connecting the j^{th} hidden node with the output nodes, $g(\cdot)$ is an activation function, w_j is the weight vector connecting the j^{th} hidden node with the input nodes, b_j is the bias of the j^{th} hidden node. In Eq. (1), the w_j and b_j are randomly generated, the β_j may be obtained by solving the following linear system (2).

$$\sum_{j=1}^{m}\beta_j g\left(w_j \cdot x_i + b_j\right)=y_i \tag{2}$$

the Eq. (2) can be written in a matrix format as

$$H\beta = Y \tag{3}$$

where,

$$H=\begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_m \cdot x_1 + b_m) \\ \vdots & \vdots & \vdots \\ g(w_1 \cdot x_n + b_1) & \cdots & g(w_m \cdot x_n + b_m) \end{bmatrix}$$

$$\beta = \left[\beta_1^T, \beta_2^T, \cdots, \beta_m^T\right]^T$$

$$Y = \left[y_1^T, y_2^T, \cdots, y_n^T\right]^T$$

H is the output matrix of the input layer of SLFN, it is usually a non-square matrix. The approximated solution of (3) can be obtained by solving the following optimization problem [26-28].

$$\min_{\beta} \| H\beta - Y \| \tag{4}$$

The approximated solution of (4) is given by

$$\hat{\beta} = H^{\dagger} Y \tag{5}$$

$H^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H. The pseudo code of algorithm ELM [25-27] is given in Figure 1.

Algorithm 1: ELM Algorithm

Input: Training data set $D = \{(x_i, y_i) | x_i \in R^d, y_i \in R^k, i = 1, 2, \cdots, n\}$, an activation function g, and the number of hidden nodes m

Output: weights matrix β.

for $(j = 1; j \le m; j = j + 1)$ **do**

  Randomly assign input weights w_j and b_j;

end

Calculate the hidden layer output matrix H;

Calculate output weights matrix $\beta = H^{\dagger} Y$.

FIGURE 1. The pseudo code of algorithm ELM

3. Imbalanced data classification based on extreme learning machine autoencoder

Extreme learning machine autoencoder (ELM-AE) [24] is a generative model, it's architecture can be described by the following Figure 2.

From Figure 2, we can find that the ELM-AE is a two-layer feed-forward neural networks, the first layer is a encoder, and the second layer is an decoder. The ELM-AE has the following two characteristics:

(1) The weights and bias of the first layer (encoder) are randomly assigned, and the weights of the second layer (decoder) are analytically determined.

(2) The inputs of the ELM-AE are equal to the outputs of the ELM-AE.

The output of the j^{th} node of the first layer (i.e. the encoder) is given as follows.

$$g(w_j \cdot x_i + b_j) = g\left(\sum_{s=1}^{d} w_{js} x_{is} + b_j\right) \tag{6}$$

where $g(\cdot)$ is an activation function, usually it is a sigmoid function.

The Eq. (6) can be written as the following compact format.

$$H = g(XW + b) \tag{7}$$

In Eq. (7), X is the input data matrix, W is the weight matrix of the first layer (i.e. encoder), b is the bias vector.

The output of the t^{th} node of the second layer (i.e. the decoder) is given as follows.

$$o_t = \sum_{j=1}^{m} \beta_{jt} g(w_j \cdot x_i + b_j) \tag{8}$$

The Eq. (8) can be written as the following compact format.

$$O = H\beta \tag{9}$$

In Eq. (9), β is the weight matrix of the decoder, which will be analytically determined as in ELM done.

The ELM-AE is a generative model, one can use the trained ELM-AE to generate samples which are similar to or same as the input instances. Motivated by this idea, this paper proposed an imbalanced data classification algorithm. The proposed algorithm includes three steps:

(1) We use positive instances as seeds and apply ELM-AE to generate samples which are similar to the seeds, but different from the positive instances.

(2) Repeat step (1) several times and obtain a balanced data set.

(3) Train a classifier on the balanced data set, the trained classifier is used to classify the new samples.

It is obvious that the proposed algorithm is an iterative algorithm which will be terminated when the number of the augmented positive instances is equal to the number of the negative instances. In the iteration, we used MMD (Maximum Mean Discrepancy) [29] to measure the difference between the probability distributions of the input data and the output data.

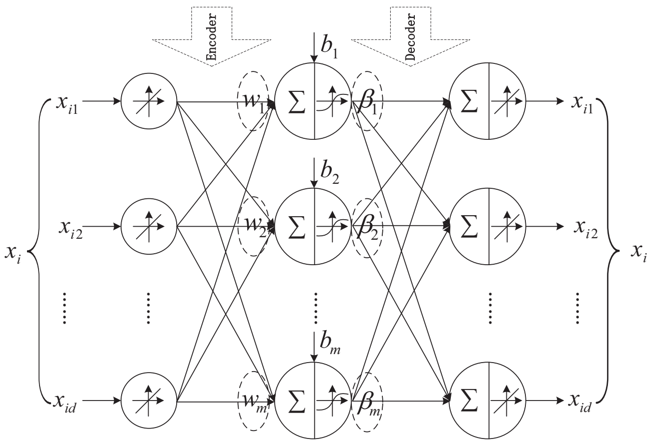

FIGURE 2. The structure of ELM-AE

4. Experimental results

In order to verify the effectiveness of the proposed algorithm, we experiment the proposed algorithm on 6 data sets in the environment of Eclipse 4.7.0 and Weka 3.9 on a PC platform with 16GB memory, Intel(R) Core(TM) i5-6600K 3.50GHz CPU, and Windows 10 operation system. The basic information of the selected data sets is given in table 1.

TABLE 1. The basic information of data sets used in our experiments

Data sets	Number of instances	Number of attributes	Imbalance ratio
MC2	125	40	1.84
KC2	522	22	3.88
Abalone	731	8	16.40
Glass	214	10	3.20
Pima	768	9	1.87

In Table 1, the MC2 and KC2 are two software defect prediction data sets [30], the Abalone, Glass, Pima are three UCI data sets [31]. Because Abalone and Glass are not binary imbalanced data sets, we transform them into TWO binary imbalanced data sets. Specifically, for data set Abalone, we select it's eighteenth class as positive class and ninth class as negative class. For data set Glass, we merge four classes (i.e. build wind float, build wind non-float, vehicle wind float and vehicle wind non-float) as negative class and merge three classes (containers, tableware and headlamps) as positive class.

In our experiments, for different data set, the number of the hidden node of ELM-AE are different, the configuration of the number of the hidden node is given in table 2.

TABLE 2. The configuration of the number of the hidden node of ELM-AE

Data sets	The number of the hidden node
MC2	55
KC2	25
Abalone	10
Glass	10
Pima	10

The commonly used assessment metrics for imbalanced data classification algorithms include Precision, Recall, F-measure, G-mean, ROC curve and AUC area, in this paper we use F-measure, G-mean and AUC area as the assessment metrics [1]. The experimental results on 5 data sets are illustrated in figure 3 to figure 7. In the figures, the horizontal axis represents the number of iterations, the vertical axis represents the assessment metrics.

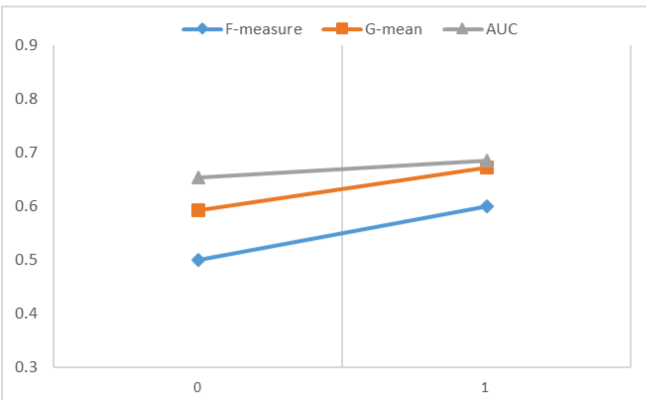

FIGURE 3. The experimental results on data set MC2

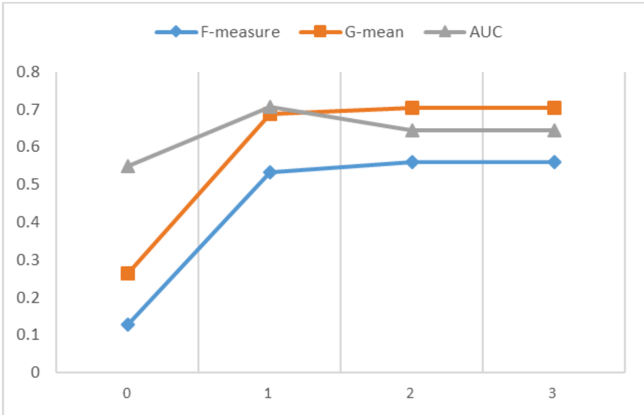

FIGURE 4. The experimental results on data set KC2

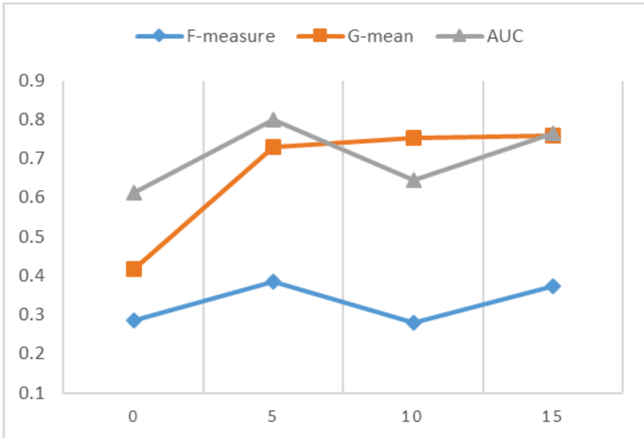

FIGURE 5. The experimental results on data set Abalone

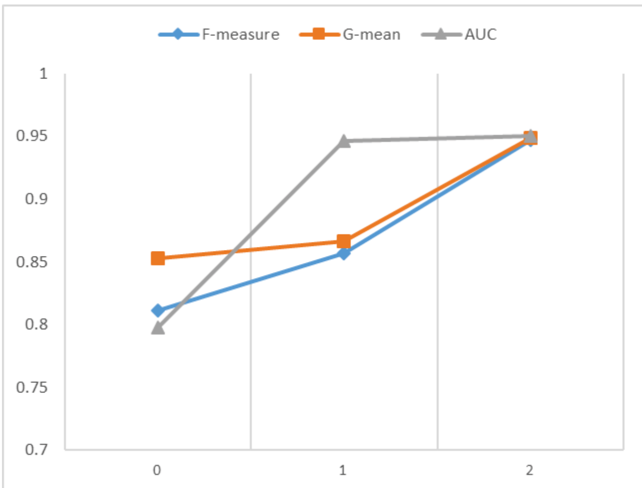

FIGURE 6. The experimental results on data set Glass

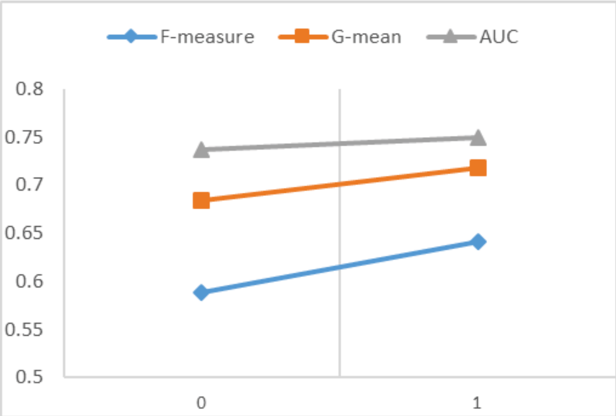

FIGURE 7. The experimental results on data set Pima

From Figures 3-7, we can find that as the number of iterations increases, the performance of the algorithm increases on all three metrics. Furthermore, in some data sets, such as MC2, Pima and Glass, the algorithm obtains it's very good results after 2-3 iterations, while the performance of the proposed algorithm will be stable after 4 iterations on data set KC2. The results on data set Abalone are similar. In short, the experimental results demonstrate that the proposed algorithm is feasible and effective.

5. Conclusions

Inspired by the idea of autoencoder, based on extreme learning machine autoencoder, this paper proposed an algorithm for addressing the problem of binary imbalanced data classification. The proposed algorithm uses ELM-AE to generate positive examples, so as to increase the number of positive examples and achieve the goal of balancing. The idea of the proposed algorithm is simple and it is easy to implement. From the experimental results on 5 data sets on three performance measures, one can find that the proposed algorithm is effective and efficient. In the further works, (1) we will conduct more experiments on more data sets with higher imbalance ratio to further prove that the proposed algorithm has good scalability. (2) we will conduct a comparative study on two data balancing methods, i.e. the method based on generative model and the method base on oversampling and attempt to figure out whether there exists essential difference between the two methods.

Acknowledgments

This research is supported by the natural science foundation of Hebei Province (F2017201026, F2016201161), by the natural science foundation of Hebei University (799207217071), and by the graduate innovation foundation of Hebei University (hbu2018ss47).

*SU-FANG ZHANG is the corresponding author.

References

[1] H. B. He, E. A. Garcia. Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering, 2009, 21(9): 1263-1284.

[2] Y. M. Sun, A. K. C. Wong, M. S. Kamel. Classification of imbalanced data: a review. International Journal of Pattern Recognition and Artificial Intelligence, 2009, 23(04):687-719.

[3] B. Tang, H. B. He. GIR-based ensemble sampling approaches for imbalanced learning. Pattern Recognition, 2017, 71:306-319.

[4] B. Krawczyk. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 2016, 5:221-232.

[5] J. H Zhai, S. F. Zhang, M. Y. Zhang, et al. Fuzzy integral-based ELM ensemble for imbalanced big data classification. Soft Computing, 22(11):3519-3531.

[6] J. H Zhai, S. F. Zhang, C. X. Wang. The Classification of Imbalanced Large Data Sets Based on MapReduce and Ensemble of ELM Classifiers. Journal of Machine Learning and Cybernetics, 2017, 8(3):1009-1017.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, et al. SMOTE: Synthetic minority over-sampling technique. Journal Artificial Intelligence Research, 2002, 16:321-357.

[8] G. Douzas, F. Bacao. Self-Organizing Map Oversampling (SOMO) for Imbalanced Data Set Learning. Expert Systems with Applications, 2017, 82:40-52.

[9] S. Piri, D. Delen, T. Liu. A synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets. Decision Support Systems, 2018, 106:15-29.

[10] Q. Li, G. Li, W. J. Niu, et al. Boosting imbalanced data learning with Wiener process oversampling. Frontiers of Computer Science, 2017, 11(5):836-851.

[11] X. Z. Wang, Q. Y. Shao, Q. Miao, et al. Architecture selection for networks trained with extreme learning machine using localized generalization error model. Neurocomputing, 2013,102:3-9.

[12] Z. Chen, T. Lin, X. Xia, et al. A synthetic neighborhood generation based ensemble learning for the imbalanced data classification. Applied Intelligence, 2017, https://doi.org/10.1007/s10489-017-1088-8.

[13] J. Vanhoeyveld, D. Martens. Imbalanced classification in sparse and large behavior datasets. Data Mining and Knowledge Discovery, 2018, 32(1):25-82.

[14] X. Y. Liu, J. Wu, Z. H. Zhou. Exploratory

undersampling for class-imbalance learning. IEEE Transactions on Systems, Man and Cybernetics- Part B, 2009, 39(2):539-550.
[15] A. Amin, S. Anwar, A. Adnan, et al. Comparing Oversampling Techniques to Handle the Class Imbalance Problem: A Customer Churn Prediction Case Study. IEEE Access, 2016, 4:7940-7957.
[16] G. Douzas, F. Bacao. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with Applications, 2018, 91:464-471.
[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al. Generative adversarial nets. Advances in Neural Information Processing Systems, 2014, vol. 1, pp. 2672-2680.
[18] I. Goodfellow. NIPS 2016 Tutorial: Generative adversarial networks. December, 2016. https://arxiv.org/abs/1701.00160.
[19] M. Mirza, S. Osindero. Conditional Generative Adversarial Nets. https://arxiv.org/abs/1411.1784v1.
[20] A. Creswell, T. White, V. Dumoulin, et al. Generative adversarial networks: an overview. IEEE Signal Processing Magazine, 2018, 35(1):53-65.
[21] G. E. Hinton, R. S. Zemel. Autoencoders, minimum description length, and Helmholtz free energy. Advances in Neural Information Processing Systems 6. J. D. Cowan, G. Tesauro and J. Alspector (Eds.), Morgan Kaufmann: San Mateo, CA.
[22] H. Bourlard, Y. Kamp. Auto-association by multiplayer perceptrons and singular value decomposition. Biological Cybernetics, 1988, 59:291-294.
[23] I. Goodfellow, Y. Bengio, A. Courville. Deep Learning. MIT Press, 2016.
[24] L. L. C. Kasun, H. Zhou, G. B. Huang, et al. Representational Learning with Extreme Learning Machine for Big Data. IEEE Intelligent Systems, 2013, 28(6):31-34.
[25] G. B. Huang, Q. Y. Zhu, C. K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. Proceedings of International Joint Conference on Neural Networks (IJCNN2004), 2004, vol. 2, pp. 985-990.
[26] G. B. Huang, Q. Y. Zhu, C. K. Siew. Extreme learning machine: Theory and applications, Neurocomputing, 2006, 70:489-501.
[27] G. B. Huang, L. Chen, and C. K. Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Transactions on Neural Networks, 2006, 17(4):879-892.
[28] G. B. Huang, H. M. Zhou, X. J. Ding, R. Zhang. Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 2012, 42(2), 513-529.
[29] B. Schölkopf, J. Platt, T. Hofmann. A Kernel Method for the Two-Sample-Problem. Proceedings of the 2006 Conference on Neural Information Processing Systems, 2007, Pages:513-520.
[30] J. Sayyad Shirabad, T. J. Menzies. The PROMISE Repository of Software Engineering Databases [http://promise.site.uottawa.ca/SERepository]. School of Information Technology and Engineering, University of Ottawa, Canada, 2005.
[31] M. Lichman. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.