

# The Application of Software Defect Types Prediction Based on Grey Entropy Absolute Relational Analysis

Dong LianJie\*, Shao HongBo and Zhou Jing

College of Science, Agriculture University of Hebei, Baoding, China

**Abstract:** In the software development process, reliability prediction can effectively improve its quality. But the small project data sets often restrain the predicting methods of the traditional software defect types, which causes the inaccuracy and unreliability of the predicting results. Grey relational analysis (GRA) is a method which is always used to describe the degree of influence among the factors and suitable for small project data sets. Therefore, on the basis of grey entropy absolute relational analysis (GEARA), we propose a software defect classification prediction method. Firstly, this method chooses an algorithm to carry on analysis and choice for feature attribute, which prediction result has greater relation degree. And then, it detects and removes anomaly project by using anomaly project detection algorithm and gets a set of engineering project. Finally, we use the grey entropy absolute relational analysis to predict software defect type. The simulation experiment indicated that proposed method owe a higher prediction precision than the traditional ones and predicted faster.

**Keywords:** Anomaly project, feature subset selection, grey system theory, grey entropy absolute relational analysis, software defect prediction.

## 1. INTRODUCTION

Software defect type prediction is a typical application of data mining methods and it is a much more important prediction method. In software development process, decision-makers can make necessary planning and adjustments according to the prediction results in order to optimize the development progress and reduce the loss. Despite that many researches and technology have been put forward accordingly, the software defect prediction models rarely consider small-scale software engineering data sets [1, 2]. However, the software item sets of prediction defect types are usually small, the relationship between the software defect types and data properties are often unclear. There would be some difficulties in selecting the predicting projects and the corresponding featured subsets(attribute sets) by the traditional statistical or machine learning methods, which usually require a large and perfect data set and a certain degree of distribution. The data set would be too small to make the prediction method sustained and effective, resulting in less accurate predictions. Most of the software engineering projects in each phase would eventually be overrun.

Grey system theory is a data theory based on small scale, poor information and uncertainties [3].

People often make use of the depth of color to describe the clarity of information, with "black" means unknown information, with "white" means the completely clear information, with the "gray" indicates a part of clear information and a part of unclear information. Accordingly,

the information which is completely clear is called white system, the completely unclear information is called black system, and the gray system is known as partly clear and partly unclear information.

GRA theory is a factor analysis, is a comparative analysis in quantification between the development trends of various factors, this method needs a small demand of sample size, does not require the typical distribution when analysis, and the analysis results are generally in consistent with the qualitative analysis results. Thus it is well suited for software defect type prediction. A distinct advantage of Grey system theory is that it uses a part of clear information to predict the system behavior. GRA is a method of gray system theory. GRA provides a quantitative measure for a system evolving situation, so it is suitable for dynamic analysis [4, 5]. Employed this method to predict software hours, and proposed related prediction algorithm.

Almost all of the software engineering data sets exit abnormalities, which could make unreasonable predictions. On the other hand, every software project has featured subsets, some of which can be used to predict the type of software defects, but some cannot [6]. So how to choose and identify some useful features, how to remove some useless or irrelevant features is of particular importance.

Grey relational entropy will mainly apply gray entropy theory to Deng's relation, whereby a new method of gray relational analysis occurs. The method overcomes the shortcomings associated with Deng's relation: local point relational coefficient controls the entire grey relational sequence tendency and causes loss of information. However, because the Deng's relation is not unique, the gray relational entropy is not unique too, so it is likely to cause inconsistent results between the evaluation and qualitative analysis. Therefore, this article is going to make improvement based

on gray entropy relational analysis method, and apply gray entropy theory to the gray absolute relation, then propose the absolute relational analysis model based on gray entropy, and apply it to the software defect type prediction.

**2. GRA METHOD**

Acting on the gray relational space, GRA is a theory built on system theory, space and control theory, which can be used to obtain relationship information between the elements [7]. Main principle of GRA: In the development process of the system, if the two factors are in consistent trends, namely a higher degree of simultaneous changes, then they can be described as a high degree of association; and vice versa. The association grade of factors between the two systems change with time or different objects, we called it relational grade.

Main features of GRA: You can use a relatively small part of the data analysis information to identify and analysis the information between the main elements. Grey relational grade (GRG) can quantify the degree of influence between the compare variables [8]. The measure of GRG does not necessarily make the assumptions of data distribution types, for it can use the relative distance between the elements directly to represent it. The distance between the elements is shorter, the impact is greater.

GRA method consists of two steps:

Step 1: Determine the reference number of columns reflecting system behavior characteristics and the compare number of columns affecting system behavior. Data sequence reflecting the characteristics of system behavior is called reference number of columns. Data sequence affecting system behavior is called compare number of columns (target number of columns).

Step 2: Calculate the gray relational grade. GRA makes use of gray relational coefficient to describe the changing trend between the reference number of columns and the compare number of columns. Specific methods are as follows:

Set  $X_0(k) = [X_0(1), X_0(2), X_0(3) \dots X_0(n)]$  as a reference sequence,  $X_i(k) = [X_i(1), X_i(2), X_i(3) \dots X_i(n)]$ ,  $i = 1, 2, \dots, n$  as compared sequences, the coefficient relation of the sequence between the reference sequence and comparison points as

$$r_{0i}(k) = \frac{\min_k |X_0(k) - X_i(k)| + \xi \max_i \max_k |X_0(k) - X_i(k)|}{|X_0(k) - X_i(k)| + \xi \max_i \max_k |X_0(k) - X_i(k)|} \quad (1)$$

In the formula,  $\xi \in (0,1)$  is distinguishing coefficient. Used to adjust the difference between the relational coefficient in the environment, and when  $\xi = 1$ , there is no change in the comparative environment. When  $\xi = 0$ , the comparative environment does not exist. When large variations appear in the data,  $\xi$  usually float between [0.1, 0.5], and reduce the impact deviation in  $\max_i \max_k |X_0(k) - X_i(k)|$ .

Due to the relational coefficient of the contrast is more complicated, it can be linked to grey relation coefficient and grey relation through further deduce calculation. Grey relational degree between the reference sequence and comparative sequence are as follows:

$$\Gamma(x_0(k), x_i(k)) = \frac{1}{m} \sum_{k=1}^m \gamma(x_0(k), x_i(k)) \quad (2)$$

However, in practice, the influence degree of each factor in the system can vary, the formula (2) can correct it as follow:

$$\Gamma(x_0(k), x_i(k)) = \sum_{k=1}^m \beta_k \gamma(x_0(k), x_i(k)) \quad (3)$$

Of which  $\beta_k$ ,  $k = \{1, 2, \dots, m\}$  is the weight of normalized factors. It can be used to show more comparative sequences  $X_i (i \in \{1, 2, \dots, n\})$  and each element of the influence degree of the reference sequences, as  $\sum_{k=1}^m \beta_k = 1$ .

The grey relational degree between the two elements is not the only one. A main purpose of GRA method is used to obtain more series of each element in the sequence of the degree in the sort, and they don't care about the exact numerical value, so comparative sequence is particular important for relational degree of each element reference grey.

**3. THE ALGORITHM OF GREY ENTROPY ABSOLUTE RELATION**

The method of grey entropy relational analysis is to carry on the quantitative description of similar and match degree between the reference sequence and the comparative data of other sequences, and complete the orders of influence factors through quantitative index.

**3.1. The Calculation of Gray Relation Entropy**

Map the relational coefficient between the reference sequence and comparative sequence, and get the grey relation entropy comparative sequence.

$$H(p_i) = - \sum_{k=1}^n p_{ik} \ln p_{ik} \quad (4)$$

In the formula,  $p_{ik}$  is the distribution of density value

$$p_{ik} = r_{0i}(k) / \sum_{k=1}^n r_{0i}(k) \quad (5)$$

**3.2. The Calculation of Gray Entropy Relation**

The Grey relational entropy  $E_r(x_i)$  of Calculation comparison sequence  $x_i$  is

$$E_r(x_i) = H(p_i) / H_m \quad (6)$$

In the formula ,  $H_m$  represent the maximum entropy composed of a property elements, as  $H_m = \ln(n)$ .

### 3.3. The Defects of Traditional Grey Relational Entropy

Traditional grey relational entropy based on representing relational degree, but there are two disadvantages in representing relation:

- (1) Resolution ratio  $\xi$  Cause the relational coefficient,  $\xi$  can increase the reference sequence and comparing the difference between the sequence relational coefficient, different  $\xi$  can conclude two conflicting result  $r_{0i}(k) > r_{0j}(k)$  or  $r_{0i}(k) < r_{0j}(k)$  . Obviously, resolution ratio  $\xi$  influenced the objectivity of analysis, even lead to relational sequence miscalculation.
- (2) The reference sequence and compare sequence influence the relational coefficient. Set reference sequence  $x_0$  , compare sequence  $x_i = x_0 + a$  ,  $x_j = x_0 + b$  ,  $a > b$  . On the one hand, the two is not equal to the distance of the reference sequence and compare sequence,

According to the Representing relation, it can be concluded that  $r_{0i}(k) < r_{0j}(k)$ ; on the other hand, the tendency of the two comparative sequences is in full accord. Obviously, because the representing relation exists in this one, it makes this distance between sequence relational coefficients.

### 3.4. Grey Entropy Absolute Relational Degree

Considering the defects of the representing relation, this paper use grey absolute relational degree instead of the representing relational entropy. Get the improved grey relational entropy-- grey entropy absolute relational degree.

Grey entropy of absolute relational degree based on two main time series curve of the slope on the corresponding period of time to measure the geometric relationships between the sequences. If two curve slopes is equal or smaller difference all the time, the relational coefficient is large; Vice is small. The definition of Grey entropy of absolute relational degree as follows (Ke Hongfa, *et al.*, 2014):

$$r_{0i} = \frac{1}{n-1} \sum_{k=1}^{n-1} r_{0i}(k) \tag{7}$$

In the formula,  $r_{0i}(k)$  is Grey entropy absolute relational coefficient,

$$r_{0i}(k) = \frac{1}{1 + |(x_0(k+1) - x_0(k)) - (x_i(k+1) - x_i(k))|} \tag{8}$$

Apparently, grey absolute relational degree with relational coefficient of entropy and uniqueness, which can ensure the accuracy of the evaluation.

### 3.5. The Prediction of Software Defecttypes

The development of software engineering is a continuous process and the software engineering may have many period of development which will have different aspects of the improvement, but the development projects are not many of the same project. According to a large number of data collection and relative information in [9], which analyze the software defects classification mining. Classification of software defect prediction technologies is shown as follow.

In this paper, the prediction of defect types on the grey entropy absolute relational software is important useful information that unearthed in the limited data set, and finally predicts the type of software defects in late development. In reference [10], it was summarized the most frequent type of ten defect types in black box testing. This article will take the most frequently five types as the research object: ① Uncaught exception handling, ② The graphical interface anomaly, ③ Abnormal state of control, ④ control abnormality, ⑤ End/collapse. The ultimate goal is to predict the number of each defect type contains the defects on the prediction of the software defect types.

Definition 1: quintuple  $\langle t, s, d, g, o \rangle$  used to show five types of defects and its corresponding number above, use  $t, s, d, g, o$  respectively to show the number of the five defect types.

The prediction on software defect types translated into the prediction and measure on the project engineering of quintuple  $\langle t, s, d, g, o \rangle$  . In the paper, the method of software defect types consists of three types:

- (1) Feature subset selection algorithm based on grey entropy absolute relational analysis (FSS@GEARA). Retrieval and analysis associated with the result of predicting larger feature attribute, to ensure the accuracy of the data to predict.
- (2) Anomaly project detection algorithm based on grey entropy absolute relational analysis (APD@GEARA). Detect and remove abnormal engineering, thus to ensure the quality of forecast data.
- (3) Software defect type's prediction algorithm based on grey entropy absolute relational analysis (SDTP@GEARA). Using the above two algorithms to screen out the set of the project and predict the type of software defect.

### 3.6. Selection of Feature Subset

Set  $D = \{p_1, p_2, \dots, p_i, \dots, p_n\}$  is a software project,  $p_i = \{x_i(1), x_i(2), \dots, x_i(m), c_i\}$ , in the formula  $x_i(1), x_i(2), \dots, x_i(m)$  show the data of  $m$  attribute value, among  $c_i = \langle t_i, s_i, d_i, g_i, o_i \rangle$  signify five software defect types and the correspond number of defects in large. The corresponding software project set of matrix can be described as the following.

$$D = \begin{pmatrix} x_1(1) & x_1(2) & L & x_1(m) & c_1 \\ x_2(1) & x_2(2) & L & x_2(m) & c_2 \\ M & M & O & M & M \\ x_i(1) & x_i(2) & K & x_i(m) & c_i \\ M & M & O & M & M \\ x_n(1) & x_n(2) & L & x_n(m) & c_n \end{pmatrix}$$

The selection of data feature subset is to be in the property of  $m$ , and the purpose is to find some properties of relational prediction result greatly. Use these attributes, and then to predict a software defect types. Algorithm description is as follows:

(1) Preliminarily, finding common attributes in this project, make each project at the same properties. Suppose there are  $m$  common property.

(2) According to the corresponding matrix to attribute column into a collection of  $m+1$  :  
 $x(1) = \{x_1(1), x_2(1), L, x_n(1)\}$   
 $x(2) = \{x_1(2), x_2(2), L, x_n(2)\}$   
 $x(m) = \{x_1(m), x_2(m), L, x_n(m)\}$

make  
 $x(0) = \{c_1, c_2, L, c_n\} = (\langle t_1, s_1, d_1, g_1, o_1 \rangle, \langle t_2, s_2, d_2, g_2, o_2 \rangle, \dots, \langle t_n, s_n, d_n, g_n, o_n \rangle)$   
 can be concluded that:  
 $x(0) = \{\langle t_1, t_2, \dots, t_n \rangle, \langle s_1, s_2, \dots, s_n \rangle, \langle d_1, d_2, \dots, d_n \rangle, \langle g_1, g_2, \dots, g_n \rangle, \langle o_1, o_2, \dots, o_n \rangle\}$

$x(0)$  as the reference sequence,  $x(1), x(2), \dots, x(m)$  is comparative sequence, calculated separately each defect types and grey target set and each absolute relational degree:

$$\Gamma(x_t(0), x(i)), \Gamma(x_s(0), x(i)), \Gamma(x_d(0), x(i)), \Gamma(x_g(0), x(i)), \Gamma(x_o(0), x(i))$$

(3) According to the result of (2), get grey relational entropy formula in turn, sorting the corresponding attributes, deposit in the 5 set  $L_t, L_s, L_d, L_g, L_o$ .

(4) According to the result of (3), take respectively  $m*5/6$  attributes from the five set and make a property manual comparison in turn and select the most influential attributes. Making arrangement from big to small according to the size of the impact in the predict result to get the final feature subset S.

**3.7. Detection on Abnormal Engineer**

Definition 2: abnormal engineer refers to, both engineering characteristics of attributes and other engineering characteristics of the grey relational degree are greater than the fixed threshold  $\theta$ , but software defect types and other software defect types of relative error is not less than a certain threshold  $\varepsilon$ . The core idea of abnormal engineering detect based on using GRA method is: pick out the most influential engineering estimates  $I \subset D$  to predict a

specific project  $p_i \in D(i = \{1, 2, \dots, n\})$ . Specific approach is to select candidate set of abnormal engineering  $O$  according to a certain method firstly, and then pick out the anomaly in abnormal engineering project set. Finally, it concluded that the resulting abnormal project set  $O = O - U\{O_f\}$ . Abnormal engineering detection algorithm based on grey relational entropy method is described as follows:

(1) The retrieval of candidate abnormal engineering. Firstly, for a project  $p_i \in D(i \in \{1, 2, \dots, n\})$ , calculate project  $p_i$  and  $\Gamma(p_i - \{c_i\}, p_j - \{c_j\})$  of any other project  $p_j \in D(j \in \{1, 2, \dots, n\} \wedge j \neq i)$ , and then sort this project  $n-1$  according to grey relation descending, and get the most influential and first few engineering under the order, as the set  $I \subset D$ . Secondly, observe every project  $p_j \in I$  in the set. Take the corresponding  $c_j$  as the predicting outcomes of project  $p_i$ , and compare with the actual result  $c_i$ . If the difference is greater than the given threshold  $\theta_c^1$ , and the project  $p_j \in I$  as a candidate exceptions of predicted engineering, and put the project  $p_j$  into the set  $O$ . We can modify appropriately threshold  $\theta_c$  and repeat the above process, and get the final candidate set  $O$  of abnormal engineering.

(2) The remove of abnormal engineering.  $o_1 \in O$ , for every abnormal  $o_1 \in O$  in the candidate engineering set  $O$ , calculating the distance between  $o_2 \in O - \{o_1\}$  and  $d_{o_1, o_2}$ , by using  $\varepsilon_1$  and  $\varepsilon_2$  to show respectively candidate abnormal  $o_1$  and  $o_2$  about the relative error of the predicting result  $c_i$ . If the relative distance  $d_{o_1, o_2}^2$  is greater than the given threshold  $\theta_d$ , and the relative error  $\varepsilon_1$  and  $o_2$  is greater than the given threshold  $\theta_c$ , and  $o_1$  and  $o_2$  are abnormal engineering. Define a temporary collection  $O_f$  to signify the abnormal engineering collection  $O_f = O_f \cup \{o_1, o_2\}$ , and get  $O = O - O_f$ . Repeat the process until get the final exception error collection, so the final abnormal engineering collection is  $O = O - O_f$ . Introductions:

1. The comparison methods of relative error and the corresponding threshold calculation as follows: a project  $p_j$  used to predict another project  $p_i$ , use  $\varepsilon = c_j - c_i = \langle t_j, s_j, d_j, g_j, o_j \rangle - \langle t_i, s_i, d_i, g_i, o_i \rangle$  to show the relative error, the definition of  $\varepsilon > \theta_c$ , every value is greater than the threshold value  $\theta_c$  corresponding to each value of the quintuple.

- Relative distance is large enough refers to the relative distance between two project is greater than a certain threshold  $d_{o_1, o_2}$ , and set this threshold  $2 * k$ , Which signifies the number of the most influential engineering the result of the forecast.

#### 4. THE PREDICTION ON SOFTWARE DEFECT TYPES

The prediction algorithm based on grey relational entropy analysis software defect type whose core idea is: after feature subset selection and abnormal project detection treatment, the data of the engineering project set got optimized. Then select the results of the most influential works from the project. Then the actual software defect types are weighted average in this project. The result is the final forecast predicted results of the project. Algorithm is described as follows:

- Suppose you have a project  $n$ , character subset as  $I$ ; set the reference sequence as  $p_1$ , and compare sequence as  $p_2, p_3, \dots, p_n$ . Use the formula (3) to calculate the engineering of reference sequence and compare sequence  $p_2, p_3, \dots, p_n$  the grey relational entropy is  $\Gamma(p_1 - \{c_1\}, p_i - \{c_i\}) (i \in \{2, 3, \dots, n\})$ , of which  $\beta_i$  receive from the data in the formula (3). It shows that the target sequence effect the weight in the prediction of reference, in the same way  $\sum_{i=2}^n \beta_i = 1$

- The grey entropy relational degree in engineering

Based on the grey entropy relational degree in engineering  $P_1$ , use quick-sort (descending) to sort those  $n-1$  project effect size, get sorted subseries  $L_p$ . Choose  $k$  projects and the corresponding software to protect the defect type. The project impact factors of forecasting purpose project are defined as follows:

$$w_i = \frac{\Gamma(p_1 - \{c_1\}, p_i - \{c_i\})}{\sum_{j=1}^k \Gamma(p_1 - \{c_1\}, p_j - \{c_j\})} \quad (9)$$

$w_i$  shows the influence degree of each project for reference sequence. So the final predict result is:

$$\hat{c}(p_1) = \sum_{i=1}^k w_i * c(p_i) \quad (10)$$

Of which  $c(p_i)$  signify  $k$  projects, the  $i$  project correspond to the type of software defects. The final result is:  $\hat{c}(p_1) = \langle \hat{t}_{p_1}, \hat{s}_{p_1}, \hat{d}_{p_1}, \hat{g}_{p_1}, \hat{o}_{p_1} \rangle$ . The prediction correspond every forecast of software defect types.

#### 5. EXPERIMENT AND ANALYSIS

Small project data set has two characteristics: Poor information and Irregularity. Poor information, refers to the project only have different versions of the data and information, but there was no information associated with the project. And irregularity refers to the distribution of data and information does not conform to certain rules. In practice, many prediction methods have ignored the project of small data sets, Many classification is integrated in the field of machine learning algorithm of the Weka software, Training data needs to be large and rich set of project data, And the proposed algorithm can overcome the problem of the small data sets in the system.

The paper through two experiments to verify the above present reliability and practicability of the proposed prediction method. The first set of experiments, using the Weka software neighbor classification algorithm and C4.5 classification algorithm to classify the target data [10], Classification results and using the relational analysis method to predict the results based on grey entropy defect types through the comparative analysis prediction method. The second set of experiments, the pretreatment of the data set of FSS@GEARA and APD@GEARA respectively do use and not to use a combination, and according to algorithm SDTP@GEARA make the final prediction.

Because this article needs to estimate the number under the classification of defects, few existing public data set and indicate good classification data set in reality. So we have used the real data of the test team from a famous IT company. And we get it from the same project version 20, a total of 20 software engineering, which is item set  $\{p_1, p_2, \dots, p_{20}\}$ , each project  $p_i (i \in \{1, 2, \dots, 20\})$  has 13 character attributes that is  $m=13$ , the last attribute is the final software defect type corresponding to the number of defects quintuple. In the use of relational analysis method based on grey entropy type of software defect prediction, set the reference sequence as  $p_1$ , Fig. (1), and compare sequence as  $\{p_2, p_3, \dots, p_{20}\}$ ,  $k=4$ , The prediction should reserves two significant figures.

Results on the first experiments is shown in the Fig. (2). ① is the actual result, ② is the Weka classification result, ③ is the predicting result by using authors method, use  $t, s, d, g, o$  respectively to show the number of the five defect types.

Results on the second experiments is shown in the Fig. (3) ① is the actual result, ② is the raw result, ③ is the result by using FSS@GEARA, ④ is the result by using APD@GEARA, ⑤ is the result by using FSS@GEARA and APD@GEARA [11].

The second experiments of predicted results are shown in Table 1 below.

By calculating relative error  $\varepsilon$  and variance  $S$  between the predicted results and actual results in Fig. (1) can be

Table 1. The predicted results of the second experiments.

FSS@GEARA	APD@GEARA	Project Impact on the Prediction Results	SDTP@GEARA Predicting Result	Actual Result
Nonuse	Nonuse	0.43, 0.20, 0.24, 0.13	< 15.9, 14.3, 10.7, 6.2, 4.1>	< 14, 13, 8, 7, 3>
Nonuse	Nonuse	0.37, 0.27, 0.19, 0.17	< 15.2, 13.9, 10.4, 6.8, 4.3>	< 14, 13, 8, 7, 3>
Nonuse	Use	0.35, 0.28, 0.19, 0.18	< 12.9, 11.7, 6.9, 8.4, 2.4>	< 14, 13, 8, 7, 3>
Use	Use	0.33, 0.26, 0.21, 0.20	< 13.6, 12.8, 7.3, 6.9, 3.1>	< 14, 13, 8, 7, 3>

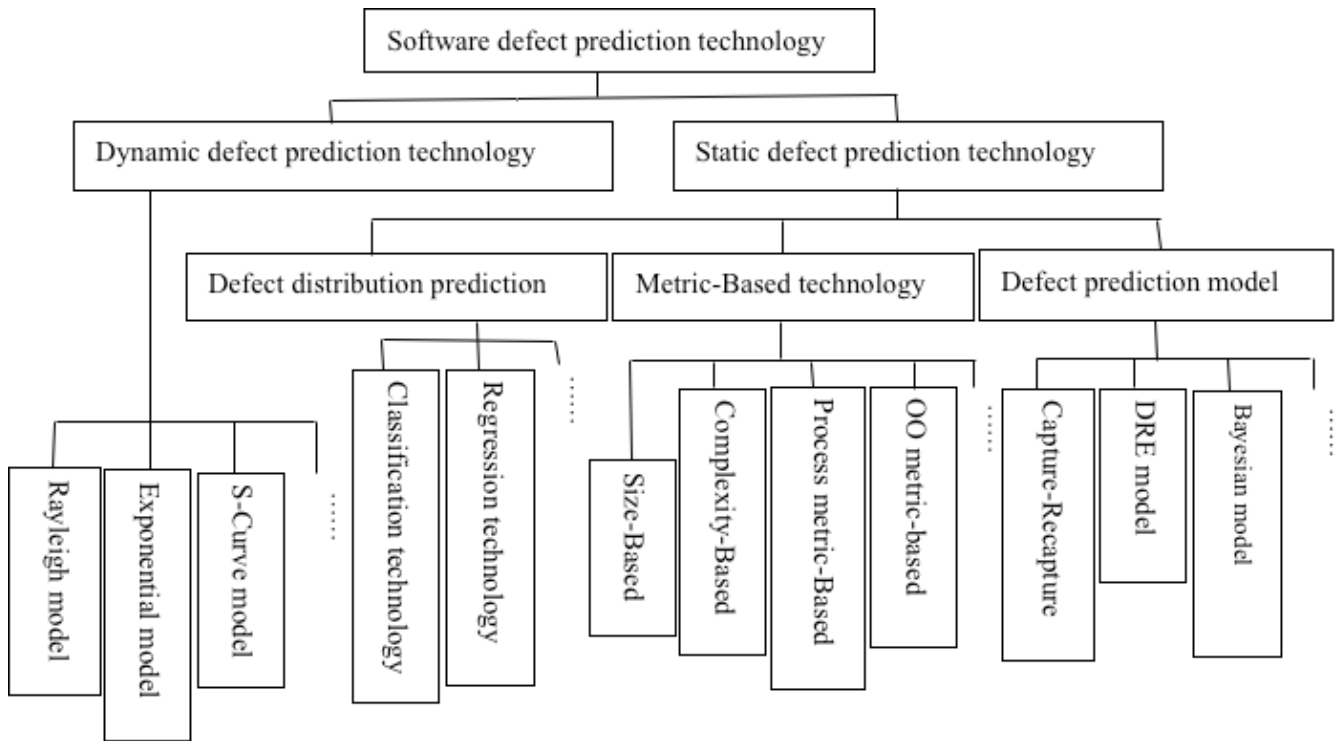


Fig. (1). Classification of software defect prediction technologies.

seen: For a small set of data, the prediction results of GRA method are more accurate and reliable than using the k neighbor classification algorithm and C4.5 classification algorithm of Weka software predicted results. After calculating the predicted results and actual results in Table 1 and the relative error  $\epsilon$  and variance  $S$  can be seen: Forecasting results of the data preprocessing is reliable than ignore data preprocessing of the predicted also should be accurate. And only use APD@GEARA method are more reliable and accurate results than using only the FCS method.

**CONCLUSION**

Absolute relational analysis method、feature subset selection algorithm and abnormal detection algorithm of engineering are proposed in this paper, and also the types of software defect prediction algorithm. The author verifies this method through two groups of different experiments by using two different algorithms. One experiment uses FSS@GEARA and APD@GEARA, another experiment uses

SDTP@GEARA. It has been analyzed and compared the data preprocessing and neglect of the predicted data preprocessing.

For a small project data set, In this paper, based on grey entropy absolute relational analysis on the predictions of a software defect type is more reliable and accurate conclusions. The proposed approach can be to use them in the software development, inspired by the decision makers make the reasonable projections for different aspects of development and deployment before the software development, So as to achieve the ultimate goal of reduce cost and risk in the software development.

**CONFLICT OF INTEREST**

The authors confirm that this article content has no conflict of interest.

**ACKNOWLEDGEMENTS**

None declared.

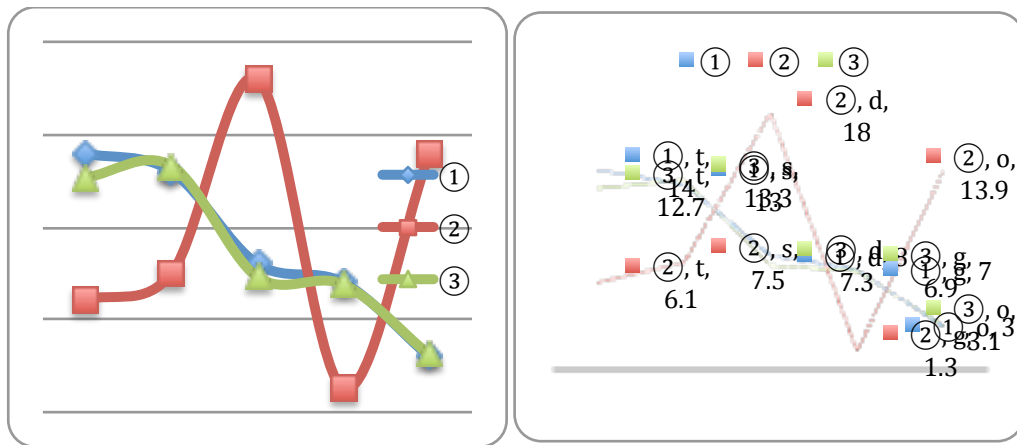


Fig. (2). Results on the first experiments.

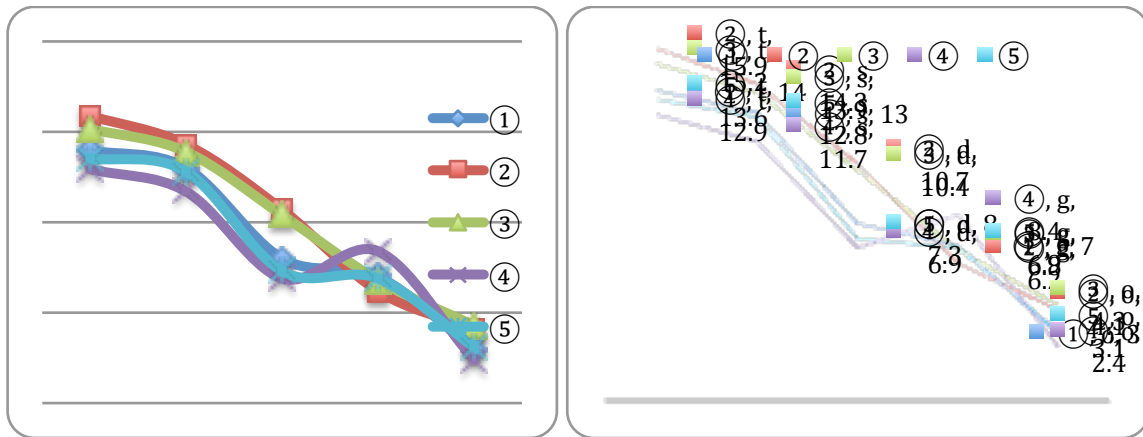


Fig. (3). Results on the second experiments.

REFERENCES

[1] J.L. Deng, "Properties of relational space for grey system," In: (J.L. Deng, Ed.), *Essential Topics on Grey System Theory and Applications [M]*, Beijing Sports University Press, Beijing, 2013, pp. 1-13.

[2] C. He, J. Xing, R. Zhu, and A. Lijuelong, "New model for software defect prediction using particle swarm optimization and support vector machine," *25<sup>th</sup> Chinese Control and Decision Conference*, 2013, pp. 4106-4110.

[3] H.I. Witten, E. Fraut, *Witten Eibe Frank Data Mining Practical Machine Learning Tools and Techniques*, Morgan Kaufman Publishers, 2011, pp. 28-39.

[4] H. Ke, H. Du, and Y. Chen, "solution method of electronic information equipment sosbased on grey system theory," In: *25<sup>th</sup> Workshop on Grey System Theory and its Applications*, 2014.

[5] P. S. Lu, E. Seo, and Y. Zhou, "Learning from mis-takes:A aomprehensive study on real word concurrency bug characteristics," In: *International Conference on Architectural Support for Programming Languages and Operating Systems-ASPLOS*, 2008, pp. 329-339.

[6] N. Li, Z. Li, and X. Sun, "Classification of Software Defect Detected by Black-Box Testing:An Empirical Study[C]," In: *Proceedings of 2010 2<sup>nd</sup> Word Congress on Software Engineering*, Wuhan, China, 2010, pp. 234-240.

[7] S. Qinbao, and S. Martin, "Predicting software project effort: A grey relational analysis based method," *Expert System with Applications*, vol. 38, no. 2011, pp. 7302-7316, 2011.

[8] A. Verikas, Z. Kalsyte, M. Bacauskiene, "Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: a survey: Soft Computing, a fusion of foundations," *Methodologies and Applications*, vol. 14, no. 9, pp. 995-1010, 2010.

[9] G. Xu, L.V. Yuejin, "Method for multiple attribute group decision making based on grey incidence," *Computer Engineering and Applications*, vol. 48, no. 9. Pp. 235-237, 2012.

[10] Y-N. Zhou, and Y. Zhu, "New and better algorithm for evaluation of overall performance of embedded computer through combining grey entropy with absolute correlation degree," *Computer Science*, vol. 38, no. 11, pp. 123-130, 2011.

[11] G-Y. Zhu, and X-B. Chen, "Flow shop multi-objective scheduling optimization research based on grey entropy relation analysis and the algorithm realization," *Control and Decision*, vol. 29, no. 1, pp. 220-230, 2014.

Received: September 16, 2014

Revised: December 23, 2014

Accepted: December 31, 2014

© LianJie et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the (<https://creativecommons.org/licenses/by/4.0/legalcode>), which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.