Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 132 (2018) 581–594

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2018)

A Nonlinear Manifold Detection based Model for Software Defect Prediction

Soumi Ghosh[a*], Ajay Rana[b], Vineet Kansal[c]

[a*]Amity University Uttar Pradesh, Noida, Uttar Pradesh, India
[b]Amity University Uttar Pradesh, Noida, Uttar Pradesh, India
[c]Institute of Engineering and Technology, Lucknow, Uttar Pradesh, India

Abstract

Software defect prediction requires developing a new technique which aims at accurately predict defective modules in software system with minimum time and space complexity as well as lesser computational cost. As such, a new model based on Nonlinear Manifold Detection Techniques has been proposed to eliminate undesirable and irrelevant attributes of high dimensional datasets by dimension reduction with more prediction accuracy and improved software quality. In this paper, a novel step towards achieving the goal by developing a new model based on Nonlinear MDTs and comparing its effectiveness with existing Feature Selection techniques for identifying the most accurate defect prediction method. The performance of different classification methods with both new and existing techniques has been evaluated, compared and also tested statistically by using Friedman test and Post Hoc analysis. The result proved that new proposed model based on Nonlinear MDTs is better performance oriented compared to accuracy level of all other techniques.

Keywords:
Feature Selection techniques; Friedman test; Nonlinear Manifold Detection techniques; Paired two-tailed T-test; Software Defect Prediction

**Corresponding Author: soumighoshphd@gmail.com*

1. Introduction

In present days with more and more use of software and increasing dependence on software in every phase of life, demand for quality software product has gone up considerably. To obtain quality software is no doubt an ever increasing requirement but similarly a very challenging task, as software system is not at all free from defects or faults. The defects or faults in the system result failure of software and deviate the system from its actual functioning

10.1016/j.procs.2018.05.012

and desired behaviour pattern which in-turn reduces software quality and finally dissatisfaction of the customers. As such, prediction of defects in the system of software at the very early stage of software development is very crucial. Prediction of defects in software system is basically a process that involves identification of sensitive and defect prone zones in the system of software. Effective and in time prediction of defects by way of classification and testing of software modules may help in obtaining assured quality software thus satisfaction of customers and at the same time reduce expenses as well as consumption of time in software development. It helps qualitative improvement of software [23] its reusability and simultaneously saves resources or inputs for development of software by minimizing software rework. Moreover, software defect prediction at the initial stage gives an additional advantage of resource planning [16] and test planning in a much better way [32]. Defect prediction in software system is, therefore, regarded as a very crucial task as it helps identifying or locating the defective and defect prone modules by spending minimum time and resources and augments the process and activities of assurance quality software. With this objective, a large number of feature selection methods have been applied in most cases but the methods were found to be complex in nature and consuming more time and money without producing desired level of accuracy in defect prediction.

For this obvious reason, a new model based on techniques namely Nonlinear Manifold Detection Techniques (Nonlinear MDTs) has been applied in order to obtain better result with much higher accuracy in defect prediction by means of elimination of those datasets that are undesired and redundant and taking into consideration only those attributes of the datasets which are relevant and significant without altering the basic structure or geometry of the datasets. This paper aims at developing a new model based on Nonlinear MDTs with the idea to identify the optimum number of attributes that can accurately predict defects in software system and comparing the same with the efficacy of the already existing Feature Selection techniques for the sake of identifying the most effective technique which have maximum level of accuracy in case of classification for software defect prediction. Moreover, performances of all the classification methods with original datasets (all attributes) have also been compared with both new model based on Nonlinear MDTs and existing Feature Selection techniques so that the impact of the most effective technique can very well be identified and then results obtained have also been tested, validated statistically by applying Paired Two-Tailed T-test, Friedman test and Post Hoc analysis for the basic purpose of finding out an accurate prediction technique. Performance measures like Accuracy Percentage, F-measure, AUC have been used for evaluating and comparing the performance of different classification methods with both new and existing techniques. The research work is a recent one comprising of datasets collected from public repository and the level of accuracy obtained by use of already existing methods reveal the fact that as compared to those techniques, the new model based on Nonlinear MDTs has larger avenues for improvement, designing better performance oriented and more accurately detecting ability of software defects and thus enabling improvement in the quality of software.

The paper has been framed in the following pattern – Background study of literature is given in section 2. Section 3 describes methodology which includes the proposed new model based on Nonlinear Manifold Detection Techniques and experimental setup. Section 4 shows comparative analysis of the experimental results and its statistical validation. Section 5 presents the concluding part with future work.

2. Background Study

A number of research work has been made in connection with prediction of software defects [1, 2, 7, 8, 18, 21, 22]. Studies based on Feature Selection Techniques [9, 24, 27] have also been discussed in this field in order to frame models for software defect prediction and quantify the software defects. Rodrigo *et al*. [3] used cluster methods using Particle Swarm Optimization (PSO) was applied for defect prediction by using Manhattan similarity measures and identified the modules that are defect prone and observed that they are better compared to other techniques. Fernando [28] conducted a study using fuzzy linear regression and statistical linear regression models and compared their performances in defect prediction of software. It was found that statistical linear regression model had better performance compared to other methods. Shulong *et al.* [19] used FEature Clustering And feature Ranking (FECAR) for selection of highly important features or attributes for defect prediction. Zhou *et al*. [31] proposed MICHAC technique whereby using Maximal Information Coefficient [MIC] for selection of relevant attributes and Hierarchical clustering technique for eliminating level of redundancy. Kehan *et al*. [10] used private

datasets for estimating the performance of seven feature ranking and four feature subset selection techniques. Huanjing *et al.* [29] evaluated the performances of six feature ranking and two ensemble techniques on a private datasets. As different datasets were used, the results obtained were rather inconsistent [25]. In our earlier research paper [11, 12, 13], existing techniques used for software defect prediction have been thoroughly reviewed and it has been observed that the problem relating to high dimensionality of software datasets resulting hike in cost of computation and reduction in desired performance of selected software models [20, 26]. A thorough review of literature from 1992 to 2016 in the area of defect prediction of software system has already been made. The review revealed the fact that out of various methods adopted Support Vector Machine (SVM), Neural Network (NN) and Advance Machine Learning (ML) methods are having more accuracy rate [12] and therefore, used widely by the researchers. Soumi *et al.* [11, 13] observed that Bayesian Network (BN) [11] and Support Vector Machine (SVM) [13] are better performing than others when used with or without Manifold Detection Techniques due to higher accuracy level. Moreover, this study has shown that for software defect prediction there is no full-proof and absolutely effective technique and so it necessities finding out a technique that can predict defects very accurately in case of large, complex datasets.

3. Methodology

3.1. New model based on Nonlinear Manifold Detection Techniques

In this research paper, endeavor has been made to frame a novel and empirical model based on Nonlinear MDTs having two main phases: (1) to identify the optimum number of attributes; and (2) to find out the most effective and accurate defect prediction technique. Practically, most of the traditional models for defect prediction applying feature selection techniques are complex, expensive and produced less accurate outcome. So, this new model has been developed for predicting software defects more accurately on high dimensional datasets.

The proposed new model based on Nonlinear MDTs consists of 10 steps:

1. Software datasets which are defective in nature have taken from open source repository;
2. The datasets have been trained and missing values eliminated.
3. Computed low dimensional embedding by performing Nonlinear MDTs on all datasets and plotted curve for extracting the point of lower dimensional embedding forming “an elbow curve”.
4. Reduced the attributes of datasets by using new model based on Nonlinear MDTs.
5. Obtained the reduced datasets and considered as new inputs for the subsequent steps.
6. All classification methods have been applied on the new datasets.
7. Performance measuring values have been computed, optimized with Ten-Fold-Cross-Validation.
8. The new model has been tested and only those better performed measuring values have been taken as output.
9. The results of new model have been validated statistically using Friedman test and Post Hoc analysis.
10. The defects in the software system have been predicted accurately applying new model based on Nonlinear MDTs and simultaneously compared with the existing techniques.

3.2. Experimental Setup

Software taken from open source repository, Relink [30] includes four datasets such as Apache, Safe, Zxing and Eclipse34_debug that have been used to predict the defect-prone modules. The number of modules in Apache, Safe, Zxing datasets were 194, 56, 399 modules respectively. Each of these three datasets has 27 attributes whereas Eclipse34_debug dataset consists of 1065 modules with 18 attributes only. The classification methods like **Bayesian Belief Network (BBN), J48**, **Naive Bayes (NB), IBK** have been used in this research work**.** The performance of these methods has been evaluated using measuring values like Accuracy Percentage, F-measure and AUC.

3.3. Nonlinear Manifold Detection Techniques

The proposed new model based on Nonlinear Manifold Detection Techniques (Nonlinear MDTs) is basically having the objective of reduction of dimensions of the datasets by way of eliminating those attributes that are undesired and redundant and select those attributes which are significant and help in enhancing the accuracy level in defect prediction of software without changing the original properties of the dataset. Nonlinear MDTs are in numbers and in the earlier paper they have been described in details [11, 13]. However, in brief we may explain different (eight) Nonlinear MDTs.

ISOMAP (Isometric Feature Mapping) – It is basically a method nonlinear in nature and applied for computation of all datasets that are reduced in dimension.

LLE (Locally Linear Embedding) – It is a technique used for dimensionality reduction computing low dimensional datasets but at the same time preserving the high dimensional features of the datasets.

Diffusion Maps – It is having some unique feature of a nonlinear algorithm which is applied basically for reducing datasets that are high dimensional.

Laplacian Eigen Maps – It represents low dimensional datasets keeping intact manifold's properties. It also computes the low dimensional dataset in such a way that the distance between a dataset and its nearest neighbors comes to a minimal [15].

NPE (Neighborhood Preserving Embedding) – It is used as a local or nonlinear technique for dimensionality reduction of datasets and helps in minimizing the cost.

SPE (Stochastic Proximity Embedding) – It is a nonlinear technique and to a good extent similar to ISOMAP and it is used for reducing dimensionality of datasets by retaining distance of the neighborhood graph as present in the graph already.

LPP (Linearity Preserving Projection) – It is used with an objective of combining the advantages of both linear and nonlinear techniques for reduction of dimensions of datasets by way of linear mapping which can minimize the Laplacian Eigenmaps' cost function.

L-ISOMAP (Landmark ISOMAP) – It is a version of ISOMAP and it works at a faster pace compared to ISOMAP. This technique is more effective provided the Landmarks that are selected as data-points for construction of maps are actually chosen in a meaningful manner and not on random basis.

3.4. Feature Selection Techniques

The Feature Selection techniques are used for selection or identification of features which are significant, useful and best fitted for a model. In this paper, we have selected three feature selection techniques such as ReliefF, Correlation based Feature selection subset evaluator (CFs), Consistency based Feature subset evaluator (ConFs) that have been analyzed using different classification methods for predicting defects in software systems.

ReliefF – This technique is actually instance based and an extended version of Relief method. The method Relief is abbreviated to RF and it is available in the Weka suite [17].

Correlation based Feature selection subset evaluator (CFs) – This technique aims at selection of attributes on the basis of prediction ability of each feature with the level of redundancy between the features. In this technique, preference is given on low correlated features compared to highly correlated features as they may be redundant basically [14].

Consistency based Feature subset evaluator (ConFs) – In order to measure the value of a feature subset, this technique uses an indicator which is termed as consistency. It aims at searching the minimal subset having equal consistency of all other features [5, 6].

3.5. Ten-Fold-Cross-Validation

The bias in case of random sampling of datasets is reduced by the method used for validation. Datasets are divided into equal size n parts and n-1 parts, the classifiers are trained. In order to evaluate the effectiveness of the

proposed new model based on Nonlinear MDTs; those parts which were eliminated are used in the test part of the dataset. Thereafter, the average of performances of all the n parts is evaluated.

3.6. Statistical Tests

Paired Two-Tailed T-Test basically comprises of samples of pairs of attributes similar in manner or attributes of a single group that have repeatedly been tested twice. This particular test has been applied for testing statistically the impact and effectiveness of all the classification methods by means of comparison with original datasets (all attributes) as well as new model based on Nonlinear MDTs and existing Feature Selection techniques with the idea of identifying the most effective technique that is having statistical significance in software defect prediction.

Friedman Test is a statistical test nonparametric in nature. It is done on the basis of assigning ranks to performance values than the actual values. In this test, the basic idea is to find out as to whether the difference of performance of classification methods with new model based on Nonlinear MDTs is at all significant or not [4].

4. Experimental Results and Comparative Analysis

4.1. With Original Datasets (All attributes)

The four datasets with all their original attributes have been used for evaluating the performance of different classification methods in order to predict the defects. The original structure of all four datasets has been shown in Fig 1,2,3,4. Fig 5, 6, 7, 8 represents the comparison of performances of all the classification methods in terms of measuring values like Accuracy Percentage, F-measure, AUC on Apache, Eclipse34_debug, Safe, Zxing datasets.

4.2. With New Model based on Nonlinear Manifold Detection Techniques

In this research work, use of new model based on Nonlinear MDTs such as ISOMAP, LLE, Diffusion Maps, Laplacian Eigen Maps, NPE, SPE, LPP, L-ISOMAP on all four datasets i.e. Apache, Eclipse34_debug, Safe, Zxing have been made for the purpose of reduction of dimensions of datasets by eliminating those attributes that are redundant and unwanted. The datasets with high dimensions have actually been reduced to three dimensional datasets and they have been considered as inputs in order to make different classification. The dimensional reduced datasets having relevant attributes after application of new model based on Nonlinear MDTs on Apache, Eclipse34_debug, Safe, Zxing have been shown in Fig 1,2,3,4. In-fact, comparison of performances of all the classification methods in terms of measuring values like Accuracy Percentage, F-measure, AUC on Apache, Eclipse34_debug, Safe, Zxing datasets have been made and presented in Fig 5,6,7,8. In case of Apache dataset, BN, J48, NB, IBK classification methods with new model based on Nonlinear MDTs showed higher percentage of accuracy (BN–Diffusion Maps 73.7113%, J48-L-ISOMAP 74.2268%, NB-Laplacian Eigen Maps 74.2268%, IBK-Laplacian Eigen Maps 68.5567% respectively). In regard to Eclipse34_Debug dataset, all classification methods with ISOMAP resulted higher accuracy level (78.6854%, 79.1549%, 77.2770%, 74.5540% respectively). So far as Safe dataset is concerned, BN, J48 , NB , IBK produced higher level of accuracy (BN-Diffusion Maps-SPE 80.3571%, J48-NPE 78.5714%, NB-SPE 76.7857%, IBK-LLE 78.5714% respectively). Further, in Zxing dataset, higher accuracy percentage has been shown by all classifiers with SPE (80.5371%, 76.7857%, 76.7857%, 75%).

4.3. With Feature Selection Techniques

In this case, same original datasets have been used with selection of attributes for evaluating the performance of various classification methods for software defect prediction with feature selection. The evaluation and comparison of performances in terms of Accuracy Percentage, F-measure, AUC in prediction of defects have been made after application of all the classifiers on these datasets and presented in Fig 5,6,7,8. For Apache dataset, classification methods showed accuracy rate (BN-ConFs 72.6804%, J48-ReliefF-CFS 70.1031%, NB-ConFs 70.1031%, IBK-ConFs 71.1340% respectively). Similarly in case of Eclipse34_Debug dataset, classification methods with few

feature selection techniques achieved cent percent (100%) accuracy level. In Safe dataset, BN, J48, NB, IBK obtained accuracy percentage (BN-ReliefF, CFS 71.4286%, J48- ConFs 76.7857%, NB-CFS 76.7857%, IBK-ConFs 73.2143% respectively). Finally, in ZXing dataset, BN, J48, NB, IBK showed (BN-ConFs 65.9148%, J48- ConFs 71.1779%, NB-ReliefF 68.6717%, IBK- ConFs 69.4236%) percentage of accuracy.

4.4. Comparative Analysis of Results

Overall comparison of performance of all classification methods used with new model based on Nonlinear MDTs viz-a-viz performance results of classifiers used with existing Feature Selection techniques as shown in Fig 5, 6, 7, 8 indicates that in regard to Apache dataset, BN, J48, NB showed higher accuracy percentage with Nonlinear MDTs (BN-Diffusion Maps 73.7113%, J48-L-ISOMAP 74.2268%, NB-Laplacian Eigen Maps 74.2268% respectively) but IBK showed better accuracy rate with existing Feature Selection techniques rather than with new model based on Nonlinear MDTs (IBK-ConFs 71.1340%). In case of Eclipse34_Debug dataset, all classification methods performed as high as 100% accuracy percentage with existing Feature Selection techniques. But in Safe dataset, all classification methods presented higher accuracy rate when used with new model based on Nonlinear MDTs (BN-Diffusion Maps-SPE 80.3571%, J48-NPE 78.5714%, NB-SPE-CFS 76.7857%, IBK-LLE 78.5714% respectively). Similarly, in case of Zxing dataset, all classification methods obtained higher accuracy percentage with new model based on Nonlinear MDTs (SPE). The classifiers that are showing higher Accuracy percentage for new model based on Nonlinear MDTs on all four datasets have been depicted in Fig 5,6, 7, 8. The result of the analysis has shown that identification of high dimensional datasets and reduction of dimensions are very crucial for accurate prediction of software defects by using new model based on Nonlinear MDTs. Results obtained from the comparison made has shown that all the classification methods have performed better when used with new model based on Nonlinear MDTs as compared to original datasets with all attributes as well as application with existing Feature Selection techniques. It has also been observed from the outcome of the experiment that new model based on Nonlinear MDTs having an edge in accuracy percentage in case of software defect prediction. Also, IBK and J48 are taken as most effective classification methods in defect prediction when it is used with new model based on Nonlinear MDTs.

4.5. Statistical Test for Result Validation

Also, a paired two-tailed T-Test is performed for comparison of performance of all the classification methods with original datasets (all attributes) as well as with new model based on Nonlinear MDTs and existing Feature Selection techniques in pair-wise manner for validating the effectiveness of the technique and performance of classification methods at significance level 0.05. Those calculated P-values for each pair of comparison having less than α (0.05) are indicated in Bold in Table 1 and validated that all classification methods outperforms with new model based on Nonlinear MDTs. Hence, it proves that new model based on Nonlinear MDTs are having very significant difference statistically than existing Feature Selection techniques as well as original datasets with all attributes.

In order to examine as to whether performance of all classification methods with new model based on Nonlinear MDTs on all software defect datasets are having significant statistical difference or not, it is required to make a comparative analysis and also to validate the results by use of a statistical test based on Friedman Test. The basis of this test is statistical comparison of performances of classification methods with new model based on Nonlinear MDTs applied for software defect prediction. The critical value at significant level 0.05 is calculated with degree of freedom (three). The calculated value $X_2 = 7.815$ and tabulated value $X_2 = 17.867$ from Chi-Square tables at $\alpha = 0.05$. This calculated P-value = 0.0001for AUC is less than α (0.05) that proves all classification methods with new model based on Nonlinear MDTs is having very significant difference statistically. Thereafter, ranks are assigned to performances of each classification method and Friedman's rank is computed. It is an accepted fact that lower mean rank shows better performance level. Table 2 depicts that statistical results of Friedman test in terms of mean ranking carried out with all classification methods for all software datasets and showed that IBK method outperforms and J48 is second best classification method. The pair-wise comparison (multiple) as well as significant

difference of all classification methods with new model based on Nonlinear MDTs as per Post Hoc analysis in terms of AUC as given in Table 2 depicts that calculated critical difference is 0.8350. Out of a total of 12 pairs of classification methods, only 4 pairs (indicated in Bold) showed greater value than the computed value of critical difference. As such, based on results of Post Hoc analysis as shown in Table 2 validates that the performance level of prediction of IBK and J48 classification methods are taken more statistically different to a significant level than other methods with new model based on Nonlinear MDTs.

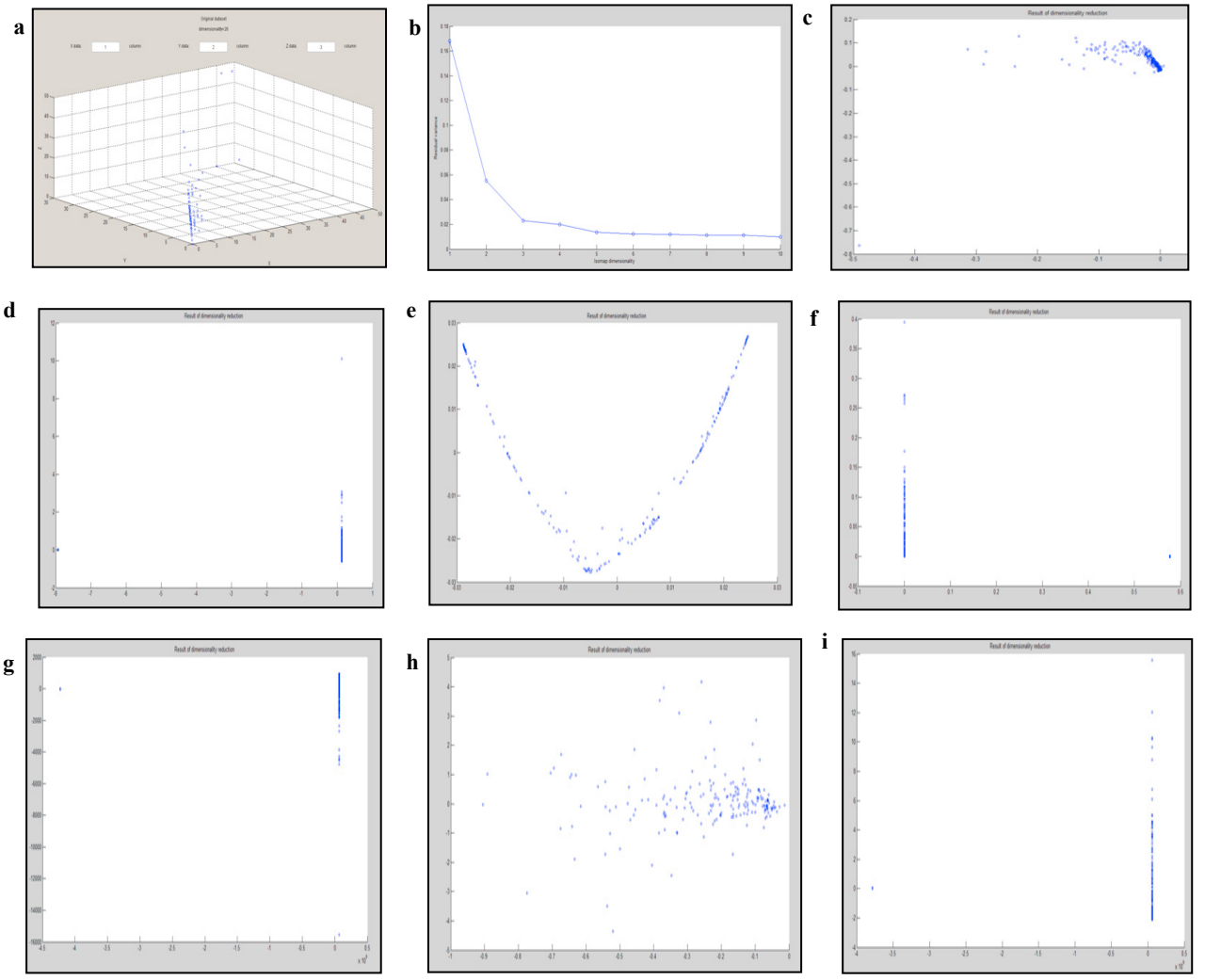

Fig. 1. (a) Original Dataset of Apache ; (b) ISOMAP on Apache; (c) LLE on Apache; (d) Diffusion Maps on Apache; (e) Laplacian Eigen Maps on Apache; (f) NPE on Apache; (g) SPE on Apache; (h) LPP on Apache; (i) L-ISOMAP on Apache.

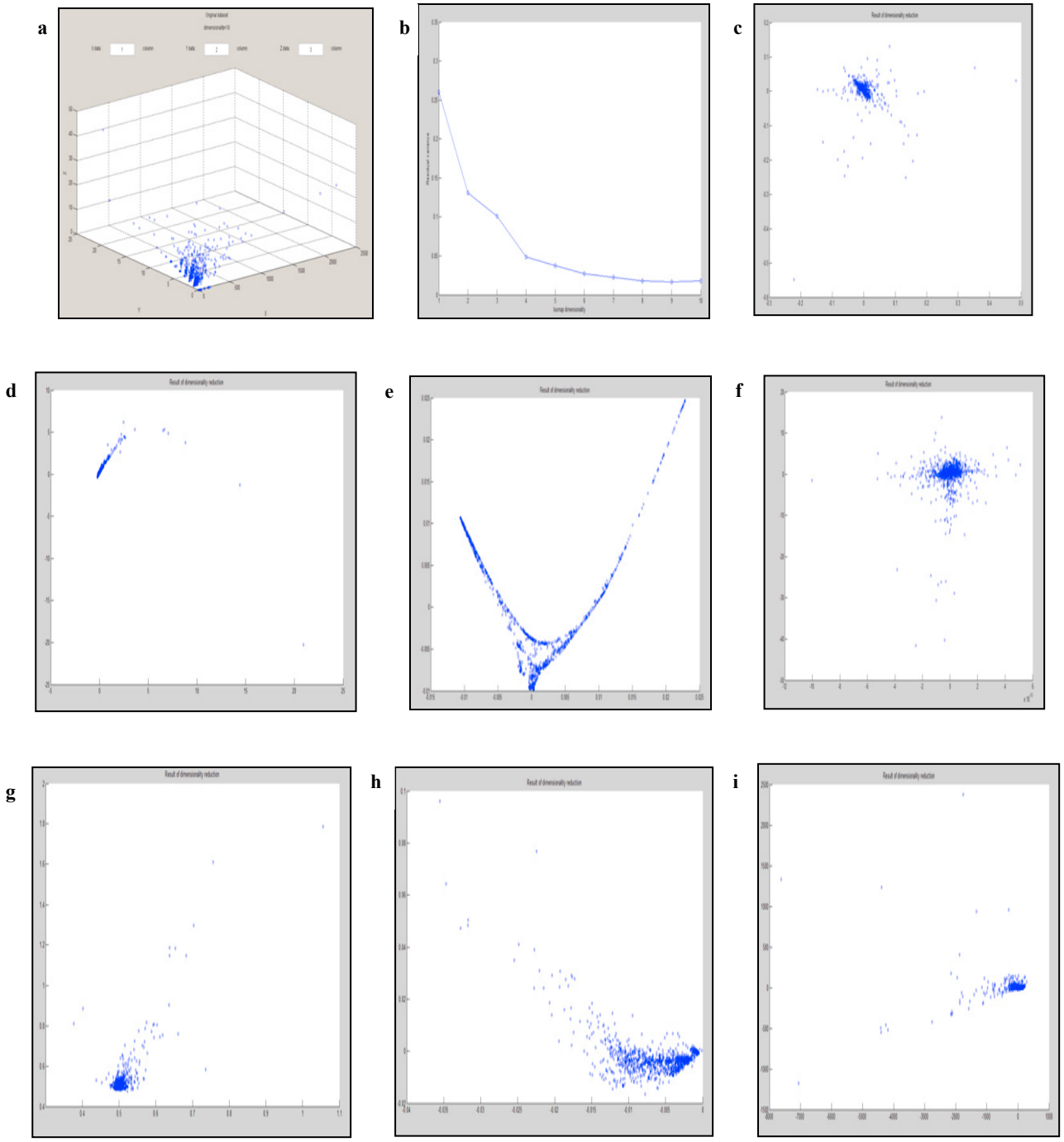

Fig. 2. (a) Original Dataset of Eclipse34_debug; (b) ISOMAP on Eclipse34_debug; (c) LLE on Eclipse34_debug; (d) Diffusion Maps on Eclipse34_debug; (e) Laplacian Eigen Maps on Eclipse34_debug; (f) NPE on Eclipse34_debug; (g) SPE on Eclipse34_debug; (h) LPP on Eclipse34_debug; (i) L-ISOMAP on Eclipse34_debug.

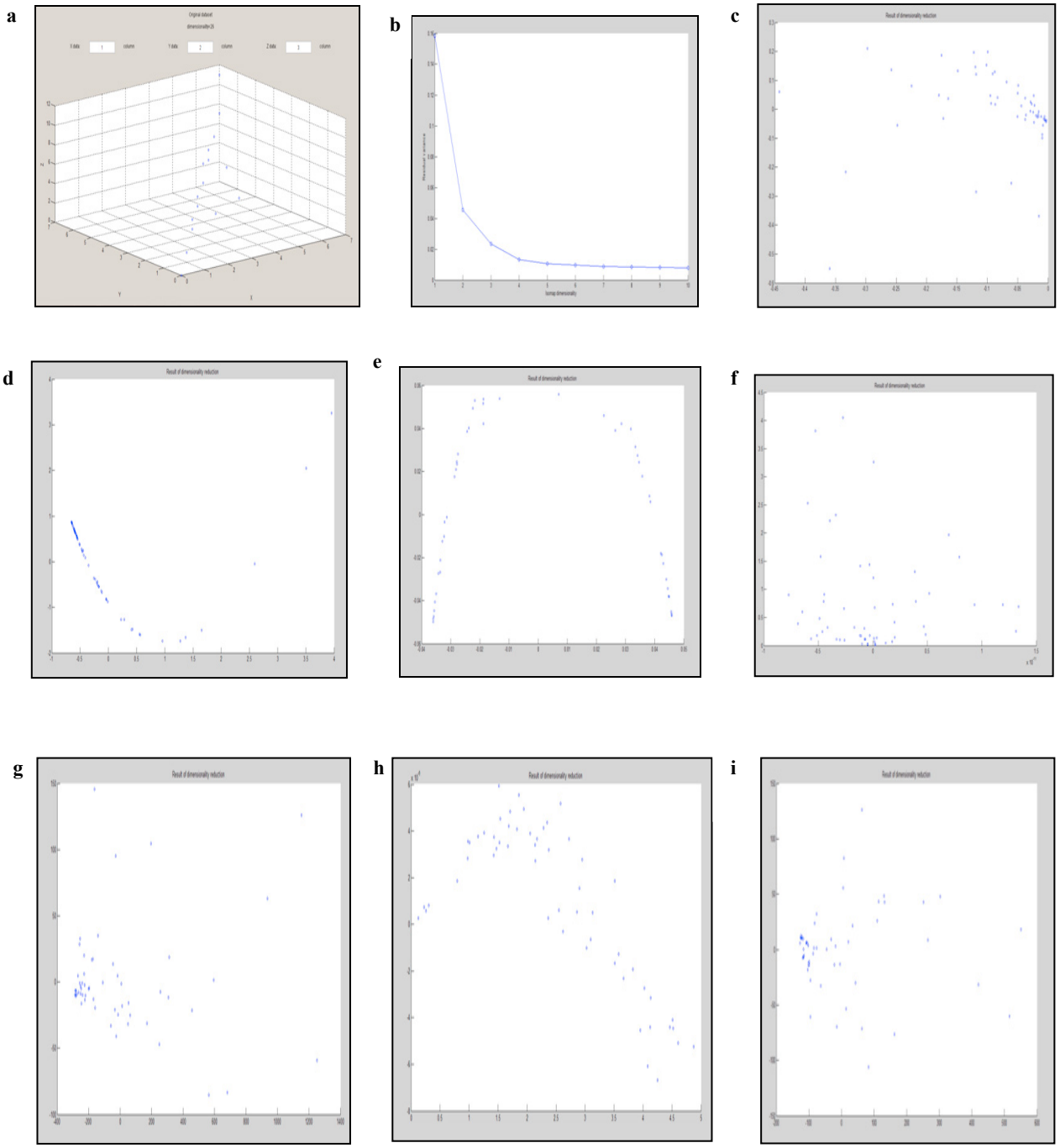

Fig. 3. (a) Original Dataset of Safe; (b) ISOMAP on Safe; (c) LLE on Safe; (d) Diffusion Maps on Safe; (e) Laplacian Eigen Maps on Safe; (f) NPE on Safe; (g) SPE on Safe; (h) LPP on Safe; (i) L-ISOMAP on Safe.

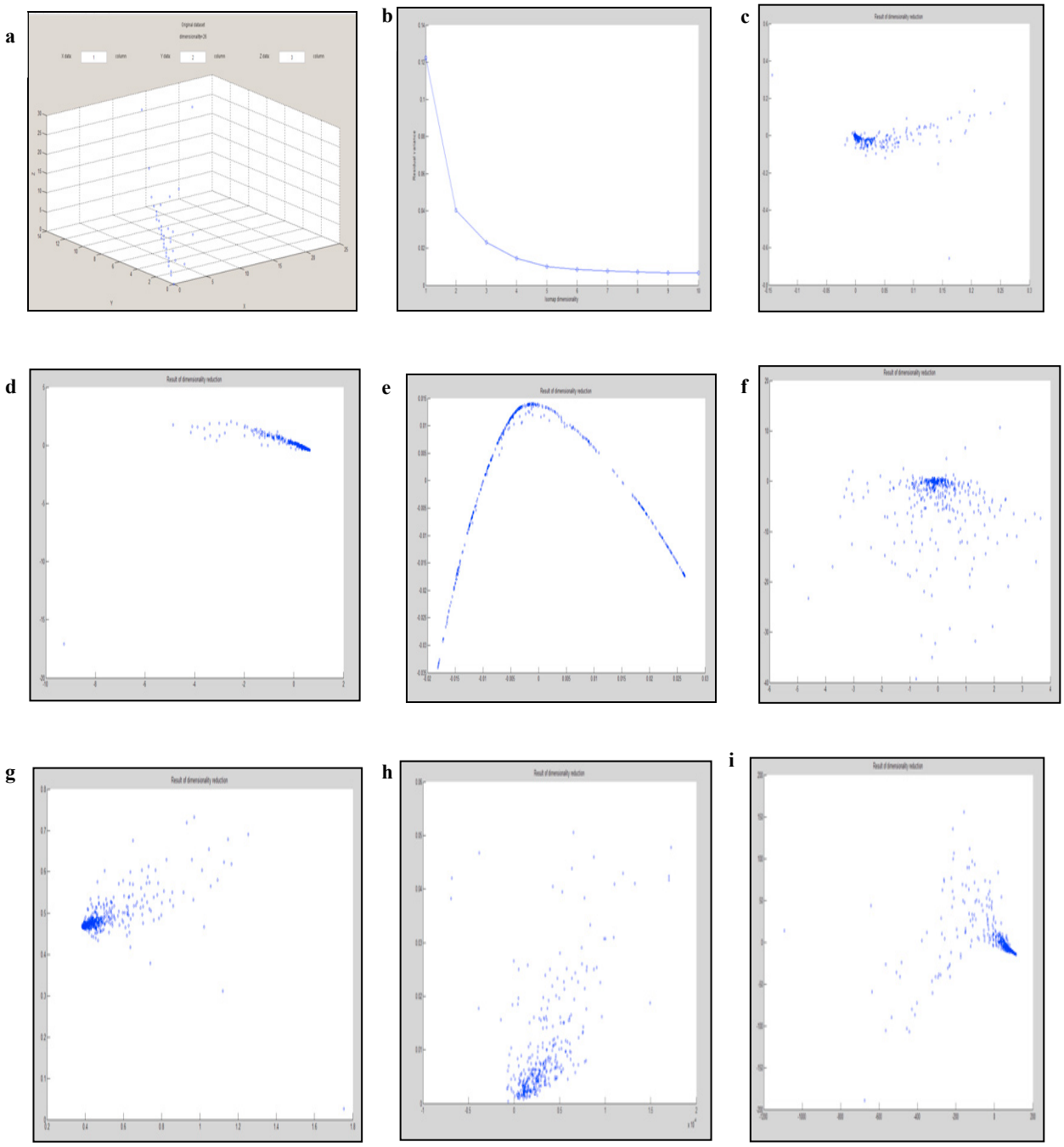

Fig. 4. (a) Original Dataset of Zxing; (b) ISOMAP on Zxing; (c) LLE on Zxing; (d) Diffusion Maps on Zxing; (e) Laplacian Eigen Maps on Zxing; (f) NPE on Zxing; (g) SPE on Zxing; (h) LPP on Zxing; (i) L-ISOMAP on Zxing.

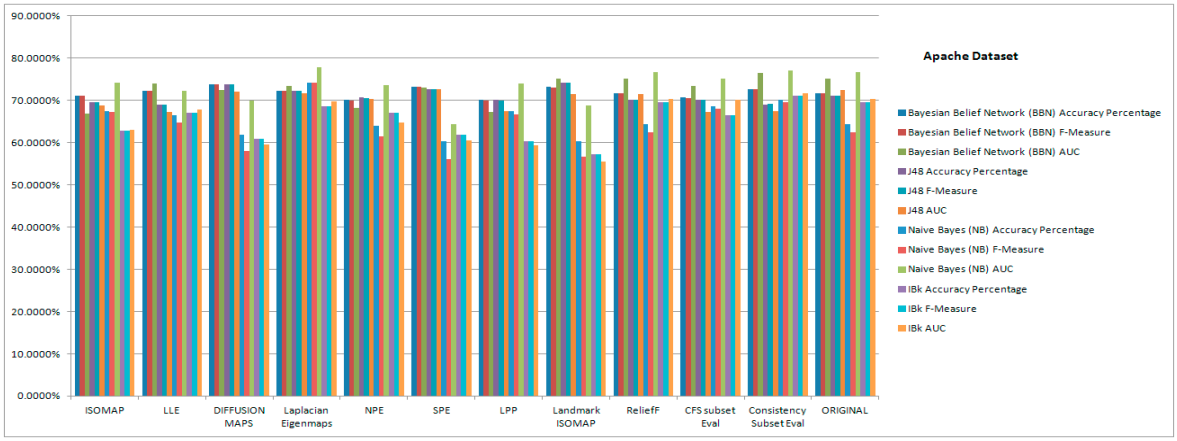

Fig. 5. Comparison of performances of all the classification methods in terms of measuring values on Apache dataset

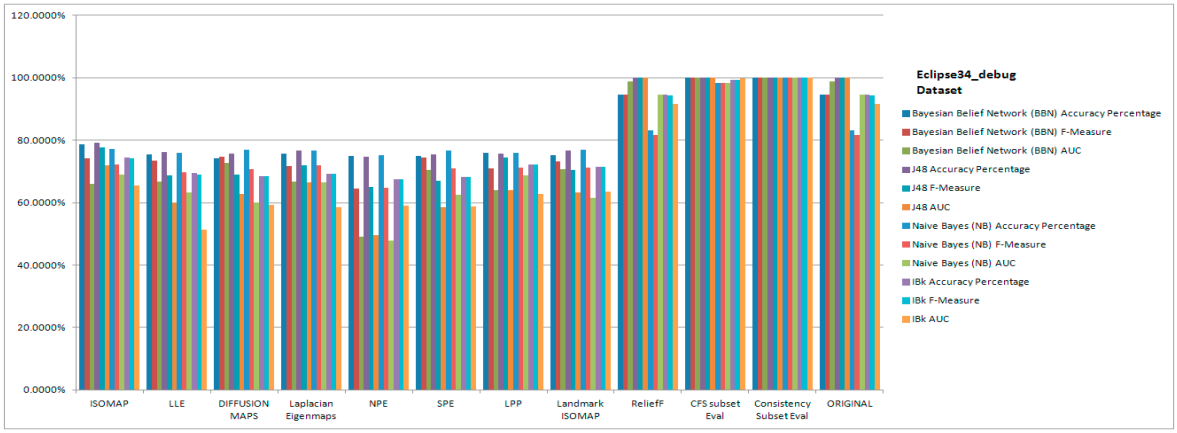

Fig. 6. Comparison of performances of all the classification methods in terms of measuring values on Eclipse34_debug dataset

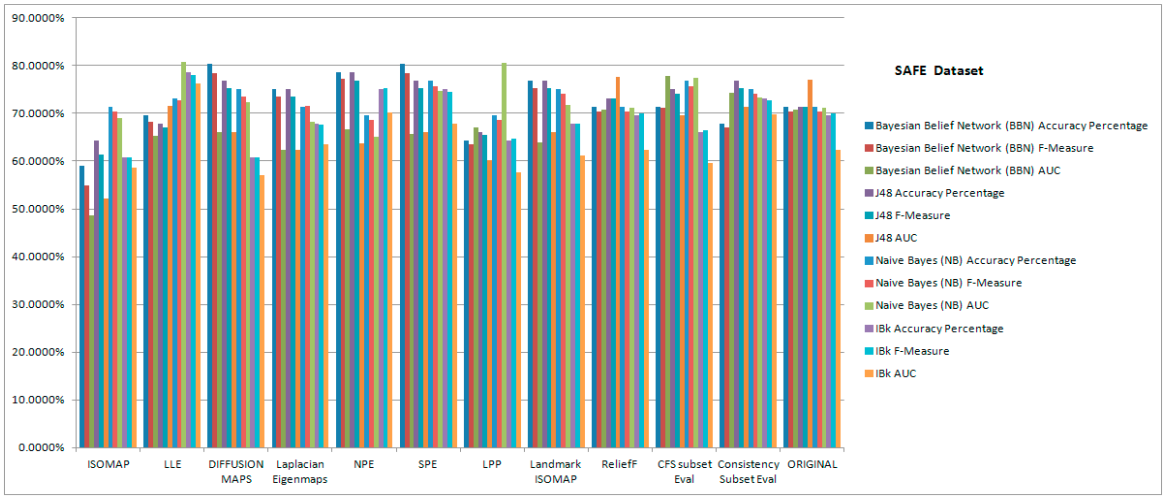

Fig. 7. Comparison of performances of all the classification methods in terms of measuring values on Safe dataset

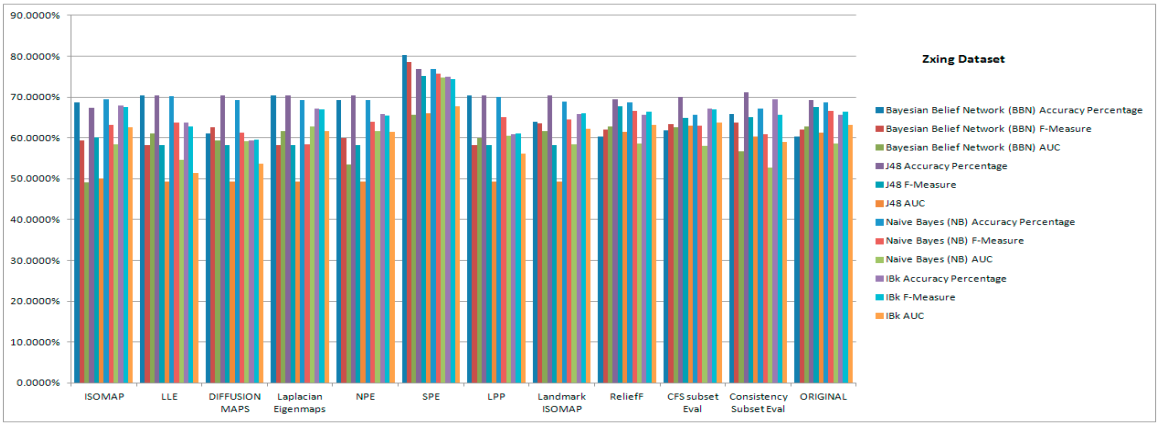

Fig. 8. Comparison of performances of all the classification methods in terms of measuring values on Zxing dataset

Table 1. Results obtained from comparison of performance of all classification methods with original datasets as well as new model based on Nonlinear MDTs and existing Feature Selection Techniques using Paired Two-Tailed T-test

Classifiers	ORIGINAL vs Feature Selection Techniques			ORIGINAL vs Nonlinear MDTS							
	ReliefF	CFS	ConFs	ISOMAP	LLE	DIFFUSION MAPS	LAPLACIAN EIGEN MAPS	NPE	SPE	LPP	L-ISOMAP
Bayesian Belief Network (BBN)	0.0000	0.2345	0.4956	0.0183	0.1304	0.1004	0.1167	0.1008	0.1618	0.1003	0.1311
	No	No	No	**Yes**	No	No	No	No	No	No	No
J48 (J48)	0.4063	0.1446	0.0692	0.0303	0.0769	0.0734	0.0553	0.0822	0.1680	0.0390	0.0692
	No	No	No	**Yes**	No	No	No	No	No	**Yes**	No
Naive Bayes (NB)	0.0000	0.1715	0.4167	0.1490	0.2219	0.1627	0.2239	0.1640	0.2966	0.3039	0.1434
	No	No	No	No	No	No	No	No	No	No	No
IBK (IBK)	0.0000	0.2885	0.1747	0.0987	0.2174	0.0479	0.1877	0.2106	0.2146	0.0498	0.0902
	No	No	No	No	No	**Yes**	No	No	No	**Yes**	No

Table 2. The performance ranking, Pair-wise comparison (Multiple) and Statistical significance difference of all classification methods with new model as per Friedman test and Post Hoc analysis in terms of AUC measure

Classification Methods	Mean of Ranks	Bayesian Belief Network (BBN)	J48	Naive Bayes (NB)	IBK
Bayesian Belief Network (BBN)	2.734 (3)	0	-0.656	(-0.438)	-0.719
		No	No	No	No
J48	**2.078 (2)**	(-0.656)	0	**(-1.094)**	-0.063
		No	No	**Yes**	No
Naive Bayes (NB)	3.172 (4)	(0.438)	**-1.094**	0	**-1.156**
		No	**Yes**	No	**Yes**
IBK	**2.016 (1)**	(-0.719)	(-0.063)	**(-1.156)**	0
		No	No	**Yes**	No

Critical difference: 0.8350

5. Conclusion

A novel and dynamic model based on Nonlinear Manifold Detection Techniques (Nonlinear MDTs) has been developed to deal with the problem of datasets having noisy attributes and high dimensions. It has been proposed exclusively for identifying the best result-oriented attributes by eliminating the irrelevant and undesired attributes of high dimensional datasets without any change in the basic properties of these datasets. Attention has been paid on comparing the effectiveness of new model based on Nonlinear MDTs with existing Feature Selection techniques mainly for the purpose of identifying the most suited and accurate technique that may predict software defects with minimum time, space complexity as well as lesser computational cost. Moreover, a comparison of performance of all the classification methods with original datasets (all attributes) as well as new model based on Nonlinear MDTs and existing Feature Selection techniques have been performed in terms of Accuracy Percentage, F-measure and AUC measure. Results obtained from the comparison and statistical validation through Paired Two-Tailed T-test proved that all the classification methods with new model based on Nonlinear MDTs are having very significant difference statistically than existing Feature Selection techniques as well as original datasets with all attributes. Also, the outcome of Friedman test and Post Hoc analysis validates that the performance level of prediction of IBK and J48 classification methods are taken more statistically different to a significant level than other methods with new model based on Nonlinear MDTs. Hence, it may be concluded that the proposed new model based on Nonlinear MDTs is having a significant and impressive outcome with more avenues in case of predicting defects in software system.

Research work in future will be associated with benchmarking this proposed new model with other Nonlinear MDTs for evaluating the performances as well as impact of other Machine Learning techniques. Finally, it is anticipated that result of this paper has contributed considerably and helped in acquiring more knowledge in the domain of predicting defects in software system, which will enable the software developers in future to make adequate effort in an improved manner for the task of defect prediction with ease, more accuracy and thus pave the way to develop qualitatively improved software product.

References

[1] Armah, Gabriel Kofi, Guangchun Luo, and Ke Qin. (2013) "Multi_Level Data Pre_Processing for Software Defect Prediction", *6th International Conference on Information Management, Innovation Management and Industrial Engineering*, 170–174. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6703111.

[2] Bell, Robert M., Thomas J. Ostrand, and Elaine J. Weyuker. (2013) "The limited impact of individual developer data on software defect prediction". *Empirical Software Engineering* **18(3):** 478–505.

[3] Coelho, Rodrigo A., Fabricio dos R.N. Guimaraes, and Ahmed A. A. Esmin. (2014) "Applying Swarm Ensemble Clustering Technique for Fault Prediction Using Software Metrics", in 2014 *13th International Conference on Machine Learning and Applications*, 356–361. http://www.scopus.com/inward/record.url?eid=2-s2.0 84924940251&partnerID=tZOtx3y1.

[4] Corder, Gregory W., and Dale I. Foreman. (2014) "Nonparametric Statistics: A Step-by Step Approach", John Wiley & Sons,(2nd eds).

[5] Dash, Manoranjan, and Huan. (2003) "Consistency-Based Search In Feature Selection." *Artificial Intelligence* **151(1):** 155-176.

[6] Dash, Manoranjan, Huan Liu, and Hiroshi Motoda. (2000) "Consistency Based Feature Selection", in Proceedings of the *4th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 98-109.

[7] Dhiman, Puneet, Manish,and Rakesh Chawla. (2012) "A Clustered Approach to Analyze the Software Quality Using Software Defects", *Second International Conference on Advanced Computing & Communication Technologies*, 36–40. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6168329.

[8] Fenton, N. E. and N. Ohlsson. (2000) "Quantitative Analysis of Faults and Failures in a Complex Software System." *IEEE Transactions on Software Engineering*, **26(8):** 797-814.

[9] Gao, Kehan, Taghi Khoshgoftaar, and Randall Wald, R. (2010) "Combining Feature Selection and Ensemble Learning for Software Quality Estimation", in Proceedings of the *Twenty-Seventh International Florida Artificial Intelligence Research Society Conference*, 47–52.

[10] Gao, Kehan, Taghi M. Khoshgoftaar, Huanjing Wang, and Naeem Seliya. (2011) "Choosing Software Metrics for Defect Prediction: An Investigation on Feature Selection Techniques." *Software: Practice and Experience*, **41(5):** 579- 606.

[11] Ghosh, Soumi, Ajay Rana, and Vineet Kansal. (2017) "A Statistical Comparison for Evaluating the Effectiveness of Linear and Nonlinear Manifold Detection Techniques for Software Defect Prediction." *International Journal of Advanced Intelligence Paradigms*, Accepted.

[12] Ghosh, Soumi, Ajay Rana, and Vineet Kansal. (2017) "Predicting Defect of Software System", in Proceedings of *the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*. Advances in Intelligent Systems and Computing, Springer, Singapore, **516:** 55- 67.

[13] Ghosh, Soumi, Ajay Rana, and Vineet Kansal. (2017) "Software Defect Prediction System Based on Linear and Nonlinear Manifold Detection", in Proceedings of the *4th International Conference on "Computing for Sustainable Global Development" INDIACom-2017*, Accepted.

[14] Hall, Mark A. (2000) "Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning", in Proceedings of the *17th International Conference on Machine Learning (ICML)*, Stanford University, 359-366.

[15] He, Xiaofei, and Partha Niyogi. (2004) "Locality Preserving Projections", *in Advances in Neural Information Processing Systems*, Cambridge, MA, USA, The MIT Press, **16:** 37.

[16] Khosgoftaar, T. M., and J.C. Munson. (1990) "Predicting Software Development Errors Using Software Complexity Metrics." *IEEE Journal On Selected Areas In Communications*, **8(2):** 253-261.

[17] Kononenko. Igor (1994) "Estimating attributes: analysis and extensions of RELIEF", *European Conference on Machine Learning*, Lectures Notes in Computer Science, **784:** 171-182.

[18] Liu, Lu, Kewen Li, Mingwen Shao, and Wenying Liu. (2015) "Fuzzy Integral Based on Mutual Information for Software Defect Prediction", in 2015 *International Conference on Cloud Computing and Big Data. IEEE Computer Society*, 93–96.

[19] Liu, Shulong, Xiang Chen, Wangshu Liu, Jiaqiang Chen, Qing Gu, and Daoxu Chen. (2014) "FECAR: A Feature Selection Framework for Software Defect Prediction", *IEEE 38th Annual Computer Software and Applications Conference*.

[20] Lu, Huihua, Bojan Cukic, and Mark Culp. (2012) "Software Defect Prediction using Semi-Supervised Learning with Dimension Reduction", in Proceedings of the *27th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 314-317.

[21] Menzies, Tim, Jeremy Greenwald, and Art Frank. (2007) "Data Mining Static Code Attributes to Learn Defect Predictors." *IEEE Transactions on Software Engineering*, **33(1):** 2-13.

[22] Najadat, Hassan, and Izzat Alsmadi. (2012) "Enhance Rule Based Detection for Software Fault Prone Modules." *International Journal of Software Engineering and Its Applications*, **6(1):** 75-86.

[23] Pushphavathi, T. P., V. Suma, and V Ramaswamy. (2014) "A Novel Method for Software Defect Prediction: Hybrid of FCM and Random Forest", *2014 International Conference on Electronics and Communication Systems (ICECS)*, 1–5.

[24] Seliya, Naeem, Taghi M. Khoshgoftaar, and Jason Van Hulse. (2010) "Predicting Faults in High Assurance Software", in *Proceedings of IEEE International Symposium on High Assurance Systems Engineering*. 26–34.

[25] Shepperd, M., and G. Kadoda (2001) "Comparing Software Prediction Techniques using Simulation." *IEEE Transactions on Software Engineering,* **27(11):** 1014-1022.

[26] Shivaji, Shivkumar, E. James Whitehead, Jr., Ram Akella, and Sunghun Kim. (2013) "Reducing Features to Improve Code Change-Based Bug Prediction". *IEEE Transactions on Software Engineering*, **39(4):** 552-569.

[27] Singh, V. B., K. K. Chaturvedi, Sunil Kumar Khatri, and Vijay Kumar. (2015) "Bug Prediction Modeling using Complexity of Code Changes." *International Journal of Systems Assurance Engineering and Management*, **6(1):** 44–60.

[28] Valles-Barajas, Fernando (2015) "A Comparative Analysis Between Two Techniques For the Prediction of Software Defects: Fuzzy and Statistical Linear Regression." *Innovations in Systems and Software Engineering*, **11(4):** 277–287.

[29] Wang, Huanjing, Taghi M. Khoshgoftaar, Jason Van Hulse, and Kehan Gao. (2011) "Metric Selection for Software Defect Prediction." *International Journal of Software Engineering and Knowledge Engineering*, **21(2):** 237-257.

[30] Wu, Rongxin, Hongyu Zhang, Sunghun Kim, and S. C. Cheung. (2011) "Relink: Recovering Links Between Bugs and Changes", in Proceedings of *19th ACM SIGSOFT symposium and the 13th European Conference on Foundations of Software Engineering ESEC/FSE 2011*, 15-25.

[31] Xu, Zhou, Jifeng Xuan, Jin Liu, and Xiaohui Cui. (2016) "MICHAC: Defect Prediction via Feature Selection based on Maximal Information Coefficient with Hierarchical Agglomerative Clustering", in Proceeding of 2016 *IEEE 23rd International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 370–381.

[32] Yuan, X., T. M. Khoshgoftaar, E. B. Allen, and K. Ganesan. (2000) "An Application of Fuzzy Clustering to Software Quality Prediction", in Proceedings of the 2000 *3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology*.