Proceedings of the Second International Conference on Intelligent Computing and Control Systems (ICICCS 2018)
IEEE Xplore Compliant Part Number: CFP18K74-ART; ISBN:978-1-5386-2842-3

# A Novel Model based on Nonlinear Manifold Detection for Software Defect Prediction

*Soumi Ghosh**
Department of Computer Science & Engineering
Amity University Uttar Pradesh,
Noida, Uttar Pradesh, India
*soumighoshphd@gmail.com

*Ajay Rana*
Department of Computer Science & Engineering
Amity University Uttar Pradesh,
Noida, Uttar Pradesh, India
ajay_rana@amity.edu

*Vineet Kansal*
Department of Computer Science & Engineering,
Institute of Engineering and Technology,
Lucknow, India
vineetkansal@yahoo.com

***Abstract*—Development of software that may be encouraging for the developers and yield more customer satisfaction in lesser time and cost requires early prediction of defects lying already in the software system. Development of a defect-free and reliable software system involves conducting series of test cases which is actually a time consuming and cost oriented exercise. It requires framing a defect prediction model applying effective technique with suitable defect prediction performance measures that may be empirically validated for ensuring relevance to software organizations. Although series of defect prediction models have been developed using various classifiers and different techniques on defect datasets but those models were not at all fault- free and fully effective to achieve the goal. As such, it has become pertinent to set up an empirical framework and develop a newer Nonlinear Manifold Detection (NMD) Model along with various machine learning approaches for prediction of defects in software in most accurate manner. The new NMD Model ventured in identifying the attributes which are best and in that process all the unwanted, redundant and undesired attributes were eliminated. In this model, critical analysis and comparison with other Feature selection approaches have been made and the results have showed that NMD Model is more accurate and effective in predicting software defects. The prediction performance of various machine learning approaches have actually been compared by using measures like Accuracy, MAE, RMSE, AUC and they have also been tested statistically by use of Friedman and Nemenyi test. The experiment finally proved that NMD Model is more effective, significant and better result-oriented in terms of accuracy than other defect prediction approaches.**

***Keywords—Dimension Reduction; Feature selection; Friedman test; Nemenyi test; Nonlinear Manifold Detection techniques; Software Defect Prediction.***

## I. INTRODUCTION

Software system is associated with an absolute problem of defects which occurs mainly due to faulty code in case of programming. Such defects in the software system are also termed as faults, errors and failures. The defects are also responsible for unprecedented and unwanted level of output, poor quality software as well as hike in time and cost in software development process. It is necessary to predict the defect prone areas at the initial stage and adopt suitable measures to remove or curb the defects for obtaining quality software products and thus satisfaction of the consumers. For this task, it is really essential to find out and apply a very suitable model focusing mainly on two crucial aspects like higher accuracy and lesser complexity in the technique. We reviewed the literature from 1992 to 2016 [9-11], it is clear that considerable number of research work has been carried out in the field of defect prediction applying various techniques but none of such techniques have been found as full proof and truly effective. This induces the requirement and urge for going in for an absolutely effective method with desired level of accuracy to predict the defects in the system. In this work, we propose to set up an empirical framework that evolves constructing a newer model i.e. Nonlinear Manifold Detection (NMD) Model with new algorithm for prediction of defects in software system. To be specific, a new model has been developed based on different Nonlinear Manifold Detection Techniques (Nonlinear MDTs) and in association with various machine learning approaches on different software defect datasets, which is termed as NMD Model.

In order to get an accurate, unbiased and better result oriented model for software defect prediction, effort has been made to develop a new empirical framework based on proposing a new NMD Model associated with seven machine learning approaches and using different Nonlinear MDTs for predicting defects in software system. By way of application of different Nonlinear MDTs, this new model ventured in identifying the attributes which are best and in that process all the unwanted, redundant and undesired attributes were eliminated. For evaluating the effect of NMD Model as well as Feature selection approaches in software defect prediction, an exhaustive comparative and critical analysis has been made for the task of identification of an accurate and most effective technique for prediction of defects in the software system by

way of minimizing number of attributes as well as other research resources. The outcome of experiments made on seven machine learning approaches over four datasets has been thoroughly compared and validated by using statistical test and Nemenyi test for determining the fact as to any significant difference exist in between the prediction performance measuring values of a particular machine learning approach and all other approaches. The machine learning approaches that have been used in this experiment are Naive Bayes (NB), Bayesian Belief Network (BBN), IBk (IBk), C4.5 Decision Tree (C4.5 DT), Random Forest (RF), Random Tree (RT) and Ada Boosting (Ada Boosting).

This paper has been systematically arranged in the following way – In section 2, related research background details have been revised, in section 3 research methodology includes proposed new NMD Model, research experimental setup, different Nonlinear MDTs, Feature selection approaches, Ten-Fold-Cross-Validation test and statistical tests have been explained. In section 4, all experimental results, its comparative analysis and its statistical validation has been discussed. Finally, section 5 covers conclusion and future work.

## II. RESEARCH BACKGROUND

A detailed review of literature showed that different techniques like Statistical methods, Regression, Case-Based-Reasoning [5], Genetic Programming [6], Neural Networks [8], Decision Trees [14], Naive Bayes [15], Fuzzy Logic [17], Machine Learning techniques those were used extensively by researchers for the purpose of prediction of software defects. S. H. Aljahdali *et al*. observed that in most cases neural networks had low or less error compared to regression models [12]. T. Menzies *et al*. proposed ROCKY classifier which showed better performance than other techniques [16]. K. El. Emam *et al*. made comparative analysis of different classification techniques and found no positive result in case of varying the combination of parameters of classifiers for obtaining better level of accuracy in defect prediction [5]. D. E. Neumann [1] proposed that performance of use of Neural Networks can be improved by using Principal Component Analysis (PCA). D. Zhang [2] stated and rated Bayesian Belief approach as the only effective and valid approach in case of prediction of software defects. In-fact, an exhaustive literature review in this particular field covering the period 1992 to 2016 has been made. The review showed that Support Vector Machine, Advance Machine Learning, Neural Networks have been widely applied by the researchers since these techniques were found to be more accurate than other techniques [10]. Similarly, some other research work in the same field has also shown that Machine Learning Techniques are also considerably effective in case of software defect prediction [15]. S. Ghosh *et al*. proved statistically that Bayesian Network (BN) performs much better and more effective in terms of misclassification error and level of accuracy, when applied on four different datasets like CM1, MW1and KC3 with or without MDTs [9]. But on the other hand, comparative analysis of functioning of various classifiers for defect prediction of software system showed and proved that Support Vector Machine (SVM) is the most effective classifier among all other techniques with or without MDTS [11]. Hence, it has become essential to find out a really accurate defect prediction technique which will be effective for all high-dimensional datasets.

## III. RESEARCH METHODOLOGY

### A. New Nonlinear Manifold Detection (NMD) Model

The empirical framework based on new NMD Model associated with seven machine learning approaches and using different Nonlinear MDTs are having following steps:

1. Defective software datasets have been collected from various open source repositories;
2. By eliminating missing values, the datasets have been trained.
3. Nonlinear MDTs were applied on all datasets for computing the embedding with low dimensions and for extracting the particular point of lower dimensional embedding of datasets, it was plotted in the form of an “an elbow curve”.
4. Use of new model based on Nonlinear MDTs for reducing the attributes of all datasets.
5. The reduced datasets were taken as inputs for the next following steps.
6. On new reduced datasets, all machine learning approaches have been used.
7. The measuring values of prediction performance are calculated and optimized by use of Ten-Fold-Cross-Validation.
8. The proposed NMD Model has undergone test and exclusively better performing measures have been considered as expected outcome.
9. The outcome of proposed NMD Model has also been statistically validated by use of Friedman and Nemenyi test.
10. Prediction of software defects in an accurate manner has been performed with the application of proposed NMD Model and the performance has also been compared viz-a–viz Feature selection approaches.

### B. Research Experimental Setup

To undertake the experiment, four different software datasets have been obtained from open source repositories in such a manner that two object-oriented datasets like BEREK and LUCENE are taken from Marian Jureczko datasets [4]. BEREK and LUCENE datasets consist of 43, 340 modules respectively, both having 21 attributes and written in JAVA language. Remaining datasets such as Ar1, Ar6 are collected from SOFTLAB repository [3] where Ar1 and Ar6 datasets are having 121, 101 modules respectively with 30 attributes each and written in C language. The machine learning approaches like Naive Bayes (NB), Bayesian Belief Network (BBN), IBk (IBk), C4.5 Decision Tree (C4.5 DT), Random Forest (RF), Random Tree (RT) and Ada Boosting (Ada Boosting) have been used in this experimental work**.** The

performance of these approaches has been evaluated using various measures like Accuracy, MAE, RMSE and AUC.

### C. Nonlinear Manifold Detection Techniques

Different Nonlinear Manifold Detection Techniques (Nonlinear MDTs), existing Feature selection approaches and also the effectiveness of Nonlinear MDTs for software defect prediction have been analyzed. In earlier research work, both the Linear and Nonlinear aspects of MDTs have been analyzed with a comparative analysis of their impacts and effective outcome in case of prediction of defects in software system [9, 11]. Emphasis has been given on Nonlinear MDTs such as ISOMAP, LLE, FASTMVU, DIFFUSION MAPS, SPE and NPE have been analyzed for reduction of dimensions in a way that original properties of datasets remain intact even in the datasets of lower dimension.

**ISOMAP (Isometric Feature Mapping)** - is a nonlinear method for reduction of dimension of datasets and it has wide use in case of computation of low dimensional datasets from high dimensional one.

**LLE (Locally Linear Embedding)** – It is regarded as a local technique which is nonlinear in nature and used for dimensionality reduction by means of computing datasets of low dimensions and side by side preserving the embedding of high dimensional datasets.

**FASTMVU** – It is having much similarity with ISOMAP in the sense that it is also presenting a neighborhood graph keeping the pair wise distance intact in the resulting graph.

**D Maps (Diffusion Maps) –** It is a framework with some unique features and it is applied as a nonlinear algorithm for the purpose of reduction of high dimensional datasets.

**NPE (Neighborhood Preserving Embedding)** – It is a technique nonlinear in nature and basically used for dimensionality reduction of datasets presented in a neighborhood graph.

**SPE (Stochastic Proximity Embedding)** – It is used as a nonlinear technique for reduction of dimensions of datasets by way of minimization of cost function of the technique.

### D. Feature Selection Approaches

This approach aims at reducing the number of actual or original features by means of directly selecting a subset of features that helps in providing required information related to classification. Feature selection approaches are of two types: based on Wrapper and based on Filter. Approaches based on Wrapper are complex in nature and so complicated in case of computation. But Filter based approaches are simpler in nature and based on the features of the data and so for large projects these approaches are preferred. Performance of classification is estimated by these approaches through indirect assessment like distance measures which show how the classes are separate from each other without having any feedback from the classifiers.

In this work, Filter based approaches such as **Correlation based Feature selection subset evaluator (CFs)** along with **Chi-Squared attribute evaluator (Chi-Squared)** have been used along with seven number of machine learning approaches for software defect prediction.

### E. Ten-Fold-Cross-Validation Test

For the basic purpose of training and validating of the proposed NMD Model for defect prediction, this particular test is applied. In this test, the datasets are in-fact divided into as many as ten parts. Training of each machine learning approach covers the nine parts of each dataset and validation of each approach is done in the tenth part. Repetition of this very process is done for ten times and the final result is obtained from a combination of the outcome of the entire process. It ensures accurate and unbiased nature of the training of such a model from the selected datasets.

### F. Friedman and Nemenyi Test

Friedman test belongs to the category of nonparametric statistical test [7] which is applied basically for finding out significant difference, if any among performances of various machine learning approaches associated with Nonlinear MDTs and assigning rank to them as per order.

Nemenyi test is considered as a powerful statistical test meant for post-hoc analysis in a condition when the sample size is same or equal but the data is not fully normalized [13]. As a matter of fact, this test is applied basically for comparing the performance of different machine learning approaches with that of Nonlinear MDTs and then finding out whether any significant statistical difference is there among the performance of different machine learning approaches in case of software defect prediction.

## IV. Experimental Results and Comparative Analysis

### A. New Nonlinear Manifold Detection (NMD)Model

In this research work, initially different Nonlinear MDTs like ISOMAP, LLE, FASTMVU, Diffusion Maps, SPE, NPE have been applied for reducing the dimensions of BEREK, LUCENE, Ar1, AR6 datasets into lowered three dimensional datasets by virtue of an 'Elbow Curve', simply by elimination of unwanted, irrelevant and redundant features. These lowered dimensional datasets having relevant and important features are used as input for applying all machine learning approaches **(NB, BBN, IBk, C4.5 DT, RF, RT and Ada Boosting).** The results obtained shows the values and performance level of different machine learning approaches in terms of Accuracy, MAE, RMSE, AUC used for software defect prediction and then these measures are compared and optimized by using Ten-Fold Cross-Validation test. The values of performance measures of different machine learning approaches using NMD Model on BEREK, LUCENE, Ar1, Ar6 datasets have been compared and represented graphically in Fig 1, 2, 3, 4.

With regard to BEREK dataset, higher percentage of Accuracy level of all machine learning approaches using NMD Model are (NB-LLE, Diffusion Maps 93.023%, BBN–FASTMVU 93.023%, IBk–ISOMAP 90.698%, C4.5 DT–Diffusion Maps 90.698%, RF and RT–ISOMAP 90.698%, Ada Boosting- ISOMAP, SPE 90.698% respectively). In respect of LUCENE dataset, percentage of Accuracy level obtained are NB-ISOMAP 65%, BBN-ISOMAP, FASTMVU 60.882%,

IBk-ISOMAP 62.647%, C4.5 DT–ISOMAP 63.824%, RF–ISOMAP 65.882%, RT-ISOMAP 60.882%, Ada Boosting–SPE 65% respectively. In case of Ar1 dataset, the higher level of Accuracy are NB- FASTMVU 92.5620%, BBN- ISOMAP, LLE, FASTMVU, Diffusion Maps, SPE 92.562%, IBk-NPE 91.736%, C4.5 DT–LLE, FASTMVU, NPE 92.562%, RF-NPE 91.736%, RT-NPE 91.736%, Ada Boosting–FASTMVU, NPE 92.562% respectively. In regard to Ar6 dataset, Accuracy level are NB-NPE 87.129%, BBN-ISOMAP, LLE, FASTMVU, Diffusion Maps, SPE, NPE 85.149%, IBk-LLE,FASTMVU 81.188%, C4.5 DT-ISOMAP, FASTMVU, SPE, NPE 85.149% , RF-ISOMAP 87.129%, RT-ISOMAP 83.168%, Ada Boosting-ISOMAP 86.139% respectively.

### *B. Feature Selection Approaches*

Similarly, in case of Feature selection approaches, the high dimensions of datasets are reduced to lower dimensions by using Correlation based Feature selection subset evaluator (CFs) and Chi-Squared attribute evaluator (Chi-Squared) approaches. In case of application of CFs approach on all high dimensional datasets BEREK, LUCENE, Ar1 and Ar6, the reduced number of dimensions selected inclusive of defect class are 7, 14, 4 and 4 respectively. In the same way, use of Chi-Squared approach on datasets BEREK, LUCENE, Ar1, Ar6 with high dimensions are reduced to 21, 21, 30, 30 number of dimensions respectively along with defect class. Further, all machine learning approaches have been applied on these dimensionally reduced datasets obtained from Feature selection approaches. The results obtained shows the values and performance level of different machine learning approaches in terms of Accuracy, MAE, RMSE, AUC used for software defect prediction and then these measures are compared and optimized by using Ten-Fold Cross-Validation test. The values of performance measures of different machine learning approaches using Feature selection approaches on BEREK, LUCENE, Ar1, Ar6 datasets have been compared and represented graphically in Fig 1, 2, 3, 4. In case of BEREK datasets, Accuracy rate shown are NB-Chi-Squared 90.698%, BBN-CFs, Chi-Squared 83.721%, IBk-CFs 95.349%, C4.5 DT- CFs, Chi-Squared 83.721%, RF- CFs, Chi-Squared 86.047%, RT-CFs 86.047%, Ada Boosting-Chi-Squared 90.698% respectively. Regarding LUCENE dataset, the higher percentage of Accuracy level found are NB-Chi-Squared 60%, BBN-CFs 63.235%, IBk-Chi-Squared 67.647%, C4.5 DT-Chi-Squared 67.647%, RF-Chi-Squared 72.647%, RT-Chi-Squared 68.235%, Ada Boosting-Chi-Squared 62.647% respectively. With respect to Ar1 dataset, the percentage of Accuracy rate shown are NB-CFs 87.6033%, BBN-CFs, Chi-Squared 90.909%, IBk-Chi-Squared 90.083%, C4.5 DT-CFs 90.909%, RF-Chi-Squared 90.083%, RT-Chi-Squared 89.256%, Ada Boosting-CFs 90.909% respectively. For the case of Ar6 dataset, higher percentage of Accuracy level is NB-CFS 84.158%, BBN-CFs, Chi-Squared 82.178%, IBk-Chi-Squared 83.168%, C4.5 DT-Chi-Squared 83.168%, RF-Chi-Squared 85.149%, RT-CFs 81.188%, Ada Boosting-CFs 83.168% respectively.

### *C. New Nonlinear Manifold Detection (NMD)Model Vs Feature Selection Approaches*

By virtue of comparative analysis of performances of all machine learning approaches using proposed new NMD Model side by side with feature selection approaches as indicated in Fig 1, 2, 3,4 shows that in case of BEREK dataset, all machine learning approaches performed better with higher level of Accuracy using NMD Model (NB-LLE, DM 93.023%, BBN–FASTMVU 93.023%, C4.5 DT–DM 90.698%, RF and RT–ISOMAP 90.698% , Ada Boosting-ISOMAP, SPE 90.698% respectively) except in case of IBk that performed well with CFs approach. For LUCENE dataset, most of the machine learning approaches had better performance with Feature selection approaches except NB and Ada Boosting approaches which performed well with proposed new NMD Model. Further, in respect of Ar1 dataset, all the machine learning approaches functioned well with new NMD Model compared to Feature selection approaches. Similarly, it is found that in case of Ar6 dataset, except IBk all other machine learning approaches performed in a better way with the proposed new NMD Model. Those machine learning approaches showing higher Accuracy when used with new NMD Model have been represented in Fig 1, 2, 3, 4. The outcome of comparative analysis of overall performance of various machine learning approaches in case of software defect prediction by use of Accuracy, MAE, RMSE and AUC measures, has finally been proved that proposed new NMD Model are having better results and more effective compared to Feature selection approaches. Particularly, BBN machine learning approach is having much better and more effective performance with new NMD Model as compared to all other approaches.

### *D. Experimental Results Validation using Statistical Tests*

A detailed comparative research work and statistical test has been performed in order to verify the fact that whether the performance of all machine learning approaches along with proposed new NMD Model is having a significant difference or not than other approaches. In order to compare performance of software defect prediction of seven machine learning approaches on four defect datasets, a popular test namely Friedman test has been applied. On the basis of calculation for RMSE and AUC, the critical value at significance level ($\alpha$) 0.05 and the degree of freedom (df) is 6. For RMSE, the calculated value of $X^2$ is 12.592 and tabulated value of $X^2$ is 82.463 obtained from Chi-Square table with df =6 and $\alpha$ =0.05 respectively. Similarly, for AUC, the calculated value of $X^2$ is 12.592 and tabulated value of $X^2$ is 36.346 obtained from Chi-Square table with df =6 and $\alpha$ =0.05 respectively. As in case of both RMSE and AUC, the P-value comes to 0.0001 after computation, that is much lower compared to the significance level $\alpha$, this ultimately proves that performance of all machine learning approaches used with proposed new NMD Model are having significant difference as compared to other approaches in prediction of software defects. For the sake of ranking, the prediction performance of each and every machine learning approaches Friedman's rank has been calculated for both RMSE and AUC. The outcome of statistical validation on the basis of mean ranking of all machine learning approaches for RMSE measure used on various software datasets proved that Ada Boosting approach is best performing one and BBN approach ranks second in performance as it has been shown in Table I. Similarly, mean ranking of all machine learning approaches for AUC measure showed that Bayesian Belief Network (BBN) approach performs best and C4.5 DT approach comes next as shown in Table II.

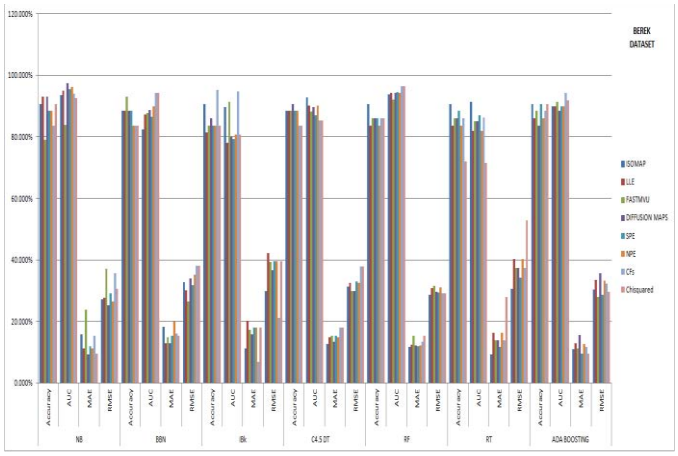


Fig. 1. Comparing the values of performance measures of different machine learning approaches using NMD Model and Feature Selection on BEREK dataset

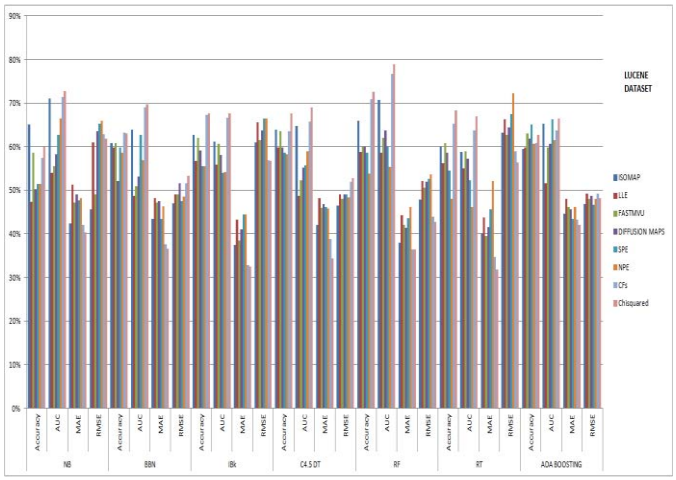


Fig. 2. Comparing the values of performance measures of different machine learning approaches using NMD Model and Feature Selection on LUCENE dataset

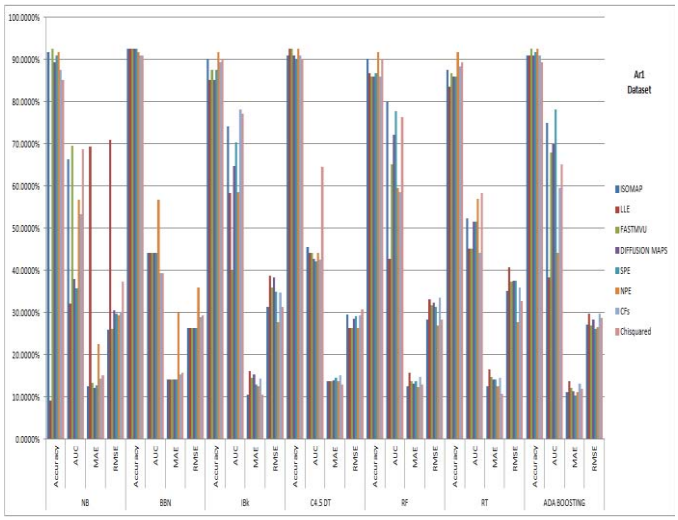


Fig. 3. Comparing the values of performance measures of different machine learning approaches using NMD Model and Feature Selection on Ar1 dataset

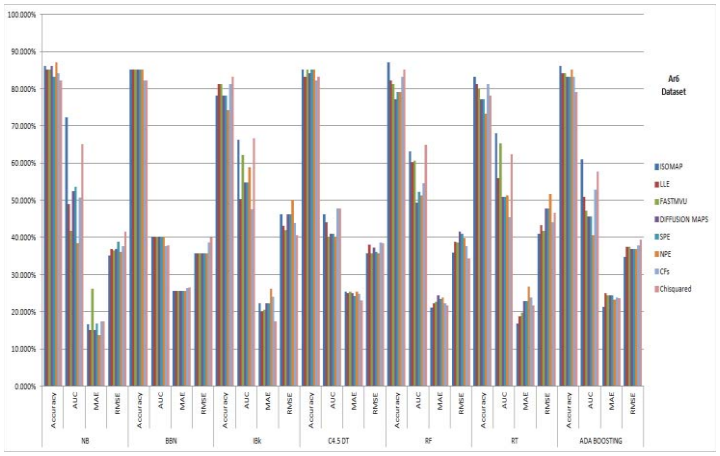


Fig. 4. Comparing the values of performance measures of different machine learning approaches using NMD Model and Feature Selection on Ar6 dataset

TABLE I. FRIEDMAN'S MEAN RANKING OF ALL MACHINE LEARNING APPROACHES FOR RMSE MEASURE

| Machine Learning Approaches | NB | BBN | IBk | C4.5 DT | RF | RT | Ada Boosting |
|---|---|---|---|---|---|---|---|
| Mean of ranks | 3.17 | **2.688** | 6.000 | 2.854 | 4.083 | 6.542 | **2.667** |
| | (4) | **(2)** | (6) | (3) | (5) | (7) | **(1)** |

TABLE II. FRIEDMAN'S MEAN RANKING OF ALL MACHINE LEARNING APPROACHES FOR AUC MEASURE

| Machine Learning Approaches | NB | BBN | IBk | C4.5 DT | RF | RT | Ada Boosting |
|---|---|---|---|---|---|---|---|
| Mean of ranks | 4.563 | **2.417** | 4.104 | **3.000** | 5.708 | 3.750 | 4.458 |
| | (6) | **(1)** | (4) | **(2)** | (7) | (3) | (5) |

TABLE III. STATISTICAL SIGNIFICANCE DIFFERENCE (P-VALUES) OF ALL MACHINE LEARNING APPROACHES WITH NMD MODEL USING NEMENYI TEST IN TERMS OF RMSE MEASURE

| **Machine Learning** | **NB** | **BBN** | **IBk** | **C4.5 DT** | **RF** | **RT** | **Ada Boosting** |
|---|---|---|---|---|---|---|---|
| **NB** | 1 | 0.988 | **0.000** | 0.999 | 0.762 | **< 0.0001** | 0.985 |
| | NO | NO | **YES** | NO | NO | **YES** | NO |
| **BBN** | 0.988 | 1 | **< 0.0001** | 1.000 | 0.279 | **< 0.0001** | 1.000 |
| | NO | NO | **YES** | NO | NO | **YES** | NO |
| **IBk** | **0.000** | **< 0.0001** | 1 | **< 0.0001** | **0.038** | 0.977 | **< 0.0001** |
| | **YES** | **YES** | NO | **YES** | **YES** | NO | **YES** |
| **C4.5 DT** | 0.999 | 1.000 | **< 0.0001** | 1 | 0.436 | **< 0.0001** | 1.000 |
| | NO | NO | **YES** | NO | NO | **YES** | NO |
| **RF** | 0.762 | 0.279 | **0.038** | 0.436 | 1 | **0.002** | 0.262 |
| | NO | NO | **YES** | NO | NO | **YES** | NO |
| **RT** | **< 0.0001** | **< 0.0001** | 0.977 | **< 0.0001** | **0.002** | 1 | **< 0.0001** |
| | **YES** | **YES** | NO | **YES** | **YES** | NO | **YES** |
| **Ada Boosting** | 0.985 | 1.000 | **< 0.0001** | 1.000 | 0.262 | **< 0.0001** | 1 |
| | NO | NO | **YES** | NO | NO | **YES** | NO |

***Critical difference: 1.8543***

From both the cases of RMSE and AUC, it has been proved that Bayesian Belief Network (BBN) approach performs much better and more accurately with the proposed new model.

Then, by applying Nemenyi test in terms of RMSE (as given in Table III) reveals that the critical difference calculated is 1.8543. Out of 21 pairs of all machine learning approaches, 20 pairs given in Bold showed higher value than the value of critical difference, as computed. Hence, on the basis of outcome of Nemenyi test given in Table III, validated that the performance of Bayesian Belief Network (BBN) approach in case of defect prediction is having significant statistical difference than other approaches when used with proposed new NMD Model.

## V. CONCLUSION

In this research work, the basic idea was to develop a new Nonlinear Manifold Detection (NMD) Model for prediction of defects in the software system and thus help qualitative improvement of software. Another objective was to evaluate the performance of various machine learning approaches used with new NMD Model proposed and they have further been compared with Feature Selection approaches in terms of measures like Accuracy, MAE, RMSE, AUC for the purpose of software defect prediction. By way of application of different Nonlinear MDTs, this new model ventured in identifying the attributes which are best and in that process all the unwanted, redundant and undesired attributes were eliminated. This new NMD Model is based on an empirical framework having seven machine learning approaches and using different Nonlinear MDTs for software defect prediction with utmost accuracy. In order to evaluate the impact of NMD Model and Feature selection approaches, a comparative and critical analysis have been made for the task of identification of an accurate and most effective technique for prediction of defects in the software system by way of minimizing number of attributes as well as other research resources. The outcome of this comparative analysis has been validated and statistically tested applying Friedman and Nemenyi test. The result of this experiment showed that performance of all machine learning approaches used with proposed new NMD Model are having better and more accurate results as compared to Feature selection approaches. In-fact, the performance of various machine learning approaches were evaluated and tested statistically by applying Friedman test in terms of RMSE and AUC. From both the cases of RMSE and AUC, it has been proved that Bayesian Belief Network (BBN) approach performs much better and more accurately with the proposed new NMD model. Moreover, the result of Nemenyi test validates that the performance of Bayesian Belief Network (BBN) approach in case of defect prediction is having significant statistical difference than other approaches when used with proposed new NMD Model. Based on this experimental results obtained it may concluded that this new NMD Model can be applied very well with all machine learning approaches and replacing existing Feature selection techniques. It can also be used as a future work in order to test varied and high dimensional datasets having additional attributes as well as in software systems based on industry.

## REFERENCES

[1] D. E. Neumann, "An Enhanced Neural Network Technique For Software Risk Analysis", IEEE Transactions on Software Engineering, vol. 28, no.9, pp. 904 -912, 2002.

[2] D. Zhang, "Applying Machine Learning Algorithms in Software Development, Modeling Software System Structures in a Fastly moving Scenario", Workshop on Monterey, Santa Margerita Ligure, Italy, 2000.

[3] G. Boetticher, T. Menzies and T. Ostrand, "PROMISE Repository of empirical software engineering data", http://promisedata.org/ repository, West Virginia University, Department of Computer Science, 2007

[4] https://code.google.com/p/promisedata/wiki/MarianJureczko, Accessed on 2017.

[5] K. El. Emam, S. Benlarbi, N. Goel, and S. Rai, "Comparing Case-Based Reasoning Classifiers for Predicting High Risk Software Components", Journal of Systems and Software, vol.55, no. 3, pp. 301–310, 2001.

[6] M. Evett, T. Khoshgoftaar, P. Chien, and E. Allen, "GP-based Software Quality Prediction", In: Proceedings of the Third Annual Genetic Programming Conference, San Francisco, CA, pp.60–65, 1998.

[7] M. Friedman, "A Comparison of Alternative Tests of Significance for the Problem of m rankings", The Annals of Mathematical Statistics, vol.11, no. 1, pp. 86-92, 1940.

[8] M. M. Thwin, and T. Quah, "Application of neural networks for software quality prediction using object-oriented metrics", In Proceedings of the 19th International Conference on Software Maintenance, Amsterdam, The Netherlands, pp. 113–122, 2003.

[9] S. Ghosh, A. Rana, and V. Kansal, "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, 2017.

[10] S. Ghosh, A. Rana, and V. Kansal, "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, vol. 516, pp. 55-67, 2017.

[11] S. Ghosh, A. Rana, and V. Kansal, "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, 2017.

[12] S. H. Aljahdali, A. Sheta, and D. Rine, "Prediction of Software Reliability: A Comparison between Regression and Neural Network Non- Parametric Models", Computer Systems and Applications. ACS/IEEE International Conference on Computer systems and Applications, Lebanon, pp. 470 -473, 2001.

[13] S. Lessmann, B. Baesens, C. Mues , and S. Pietsch, "Benchmarking Classification Models for Software Fault Prediction: A Proposed Framework and Novel Findings", IEEE Transactions on Software Engineering, vol. 34, no. 4, pp. 485-496, 2008.

[14] T. M. Khoshgoftaar, and N. Seliya, "Software Quality Classification Modeling Using The SPRINT Decision Tree Algorithm", In: Proceedings of the Fourth IEEE International Conference on Tools with Artificial Intelligence, Washington, DC, pp. 365–374, 2002.

[15] T. Menzies, J. Greenwald, and A. Frank, "Data Mining Static Code Attributes to Learn Defect Predictors", IEEE Transactions on Software Engineering, IEEE Computer Society, vol. 32, no. 11, pp. 2–13, 2007.

[16] T. Menzies, K. Ammar, A. Nikora, and S. Stefano, "How Simple is Software Defect Prediction?", Journal of Empirical Software Engineering, http://menzies.us/pdf/03simpled.pdf, 2003

[17] X. Yuan, T. M. Khoshgoftaar, E. B. Allen and K. Ganesan, "An Application of Fuzzy Clustering to Software Quality Prediction", In Proceedings of the Third IEEE Symposium on Application-Specific Systems and Software Engineering Technology, IEEE Computer Society, Washington, DC, pp. 85, 2000.