

# An Insight to Soft Computing based Defect Prediction Techniques in Software

**Kritika Verma**

Computer Science and Engineering, ASET, AUUP, Noida, 201313, India  
Email: Kritikaverma358@gmail.com

**Pradeep Kumar Singh**

Computer Science and Engineering, ASET, AUUP, Noida, 201313, India  
Email: pksingh16@amity.edu

**Abstract**—Nowadays, the complexity and size of software systems is proliferating. These factors have various pros and cons. On one side where they lead to better performance and satisfactory results, on the other side they lead to high testing cost, wacky results, poor quality and non-reliability of the product. These problems have one root cause which is referred to as defects in the software systems. Predicting these defects can not only rule out the cons but can also boost up the pros. Various techniques are present for the same which are reviewed in depth in this paper. Moreover, a comparison of these techniques is also done to throw a lime light on those which provide the best results.

**Index Terms**—Software Defects, Software Defect Prediction, Software Defect Prediction Models, Soft Computing techniques, machine Learning Techniques.

## I. INTRODUCTION

Today, we are living in a world where software technology is used extensively. Software systems have become an integral part of our lives and they should be flawless. Furthermore, Software developers are concerned about defects in their respective software systems. The definition of bugs can be divided into three parts to make it simpler to understand i.e. the meaning, the causes and the consequences of defects. Some synonyms like flaws, errors, mistakes, Faults and so on can be used to explain the meaning of defects [13, 20]. All these words have one common interpretation i.e. “Something that is not right or correct”. Moreover, defects can be present or can originate in any system development phase. Thus, it can be summed up as “Defects are the root cause of every problem that affects the software system in some or the other way”. To understand the concept of defects the reasons for their presence or occurrence must be first understood.

Some of the important reasons have been listed below [26]:

- Misperception of the requirements.

- Changing the requirements after the development process starts.
- Impractical schedule of development.
- Lack of experience in many fields namely designing field, coding field and so on.
- Human interaction can also be a cause.
- Inadequate testing skills.

To move on let's talk about the damages that defects can cause. Some of them being, loss of money and time [1], deviation from actual proposed results [1,20], producing a system that does not meet customer's expectation level [8], behavior that is unexpected or incorrect [13,20]. To stop all this from happening it's necessary to get rid of these defects and to do so defects in software system's must be predicted. The scope of software defect prediction is very vast. To start with let's define Software Defect Prediction (SDP); Software Defect Prediction can be defined as a practice of finding those parts of any software system that are defective [7, 13, 20]. Software defect prediction model's (SDPM) effectively predict Software defects. These models make use of several Software metrics available to carry out the mechanism of SDP [2]. Efficient SDP is beneficial for any software system. Firstly, it improves quality and testing [1, 20]. Secondly, it increases customer satisfaction [1]. Thirdly, it helps in reducing the cost of correcting defects at a later stage [1]. Lastly, it helps in delivering reliable software [13].

A bountiful of Soft Computing approaches have been suggested in the past for SDP [11]. Soft Computing is a term used to combine computer science techniques, AI techniques, Machine Learning techniques and some other engineering disciplines [27].

Soft Computing includes [5, 27]:

- Fuzzy Logic
- Neural Network
- Probabilistic Reasoning
- Evolutionary Computation:
  - Evolutionary Algorithm : Genetic Algorithm
  - Swam Intelligence : Ant Colony, Particle Swam Intelligence
- Machine Learning techniques [13]:

- Artificial Neural Network
- Decision Tree
- Bayesian Belief Network
- Support Vector Machine
- K-Nearest Neighbour

Some statistical methods like PCA and Feature Subset Selection add to the prediction ability of various models available [13]. Next thing that gains importance is the software metrics. Software metrics help the models to predict defects. This paper gives intense knowledge about the defect prediction techniques and all relevant topics.

This paper is organized as follows: Section II talks about the various techniques of soft computing, Section III describes the research methodology, Section IV is all about the prior research done on this topic and Section V provides the analysis results.

## II. BASIC SOFT COMPUTING TECHNIQUES USED FOR SOFTWARE DEFECT PREDICITON

The term Defect Prediction has become an immensely growing research topic nowadays. To predict defects efficiently the developers adopt plenty of techniques in some or the way so that they achieve desired results. A large number of techniques are available that ease the process. But which technique to choose and adopt, this has become the concern factor. We have adopted the similar approach considered by Singh *et al.* [29] for conducting the survey in our paper for analysis.

In this paper, an attempt has been made to provide a brief knowledge about all the relevant techniques that can be employed in the prediction process.

The basic aim of writing this research paper is to:

- 1) Provide the readers in depth knowledge about varous terms like Defect , Software defect prediction and Softeare defect prediction models.
- 2) Delve about the various techniques that can be used to predict software defects.
- 3) Conduct a comparison between the techniques and find out the ones that provide the best outcome.

The basic aim of writing this research paper is to:

- 4) Provide the readers in depth knowledge about varous terms like Defect , Software defect prediction and Softeare defect prediction models.
- 5) Delve about the various techniques that can be used to predict software defects.
- 6) Conduct a comparison between the techniques and find out the ones that provide the best outcome.

In order to accomplish these objectives intense descriptive survey was done. Numerous research papers from 1999 to 2014 were downloaded and studied based on the usage of various Soft Computing Techniques. A total of 25 research papers related to the usage of

techniques were taken into consideration and about 13 papers were considered for relevant literature review on the broad topics like the Soft Computing techniques, the Machine Learning techniques and so on.

Comparitive analysis was done on these techniques to find out the ones that give outstanding performance.

Plenty of models have been proposed in the past for SDP which use various ML techniques to learn and predict defected modules in a software system [13]. Before discussing the models we need to have in depth knowledge about the various ML techniques. To start with let's describe what Machine Learning is in brief.

Learning is characterized by two things namely gaining and understanding. Therefore, learning can be defined as a sequence of steps starting from observing data to constructing a model and finally using the grasped knowledge for prediction [13]. Whereas Machine Learning is characterized by one thing i.e. automated learning (learning of computers).

Machine learning can be of two types:

- 1) Supervised Learning
- 2) Unsupervised Learning.

Moreover, it's important to know that where can these techniques be used and why are they being used.

Why and where Machine learning is used:

- Where human expertise does not exist.  
E.g. Navigation on mars.
- When humans cannot explain their expertise well.  
E.g. Speech recognition
- When solutions are a constraint of time.  
E.g. Data Mining, Routing etc.
- When evolution needs to be adopted.  
E.g. Biometric applications

### A. Neural Network

Neural network/Artificial neural network can be defined as a system that deals with information [12, 13, 14]. Human brains working is what inspired the working of the NN/ANN [7, 13, 28].

Working of a NN can be defined as:

- 1) Design phase: Here the architecture of the network is designed keeping the key points in consideration like the no. of layers and the no. of neurons in each layer.
- 2) Training phase: Here calculations are done on the neurons and their resp. weights. Manipulations are done so that the NN gets trained.
- 3) Prediction phase: Now the new model is ready to be used and can perform the prediction.

Neural network classification is shown in figure1

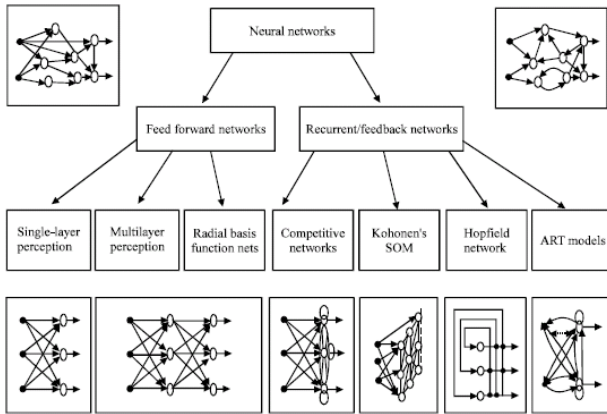


Fig.1. Classification of Neural Network

**Feed Forward Networks**

**Single Layer Perceptron Network**

These type of feed forward networks contain only one layer i.e. the input layer. Moreover it's called feed forward network because of the direction of flow of information i.e. input to output.

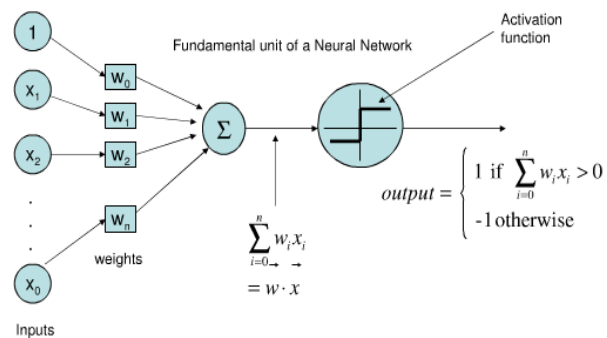


Fig.2. Single Layer Perceptron Network

**Multilayer Perceptron Network**

These type of feed forward networks consist of three different layers [13].

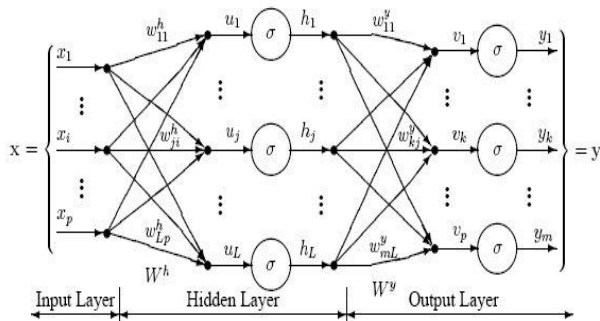


Fig.3. Multilayer Perceptron Network

**Radial Basis Function (RBF) Network**

These types of feed forward networks work on Radial basis function. These functions are used in multidimensional space for the purpose of interpolation. Here the hidden weights formulate the output with the help of some linear function.

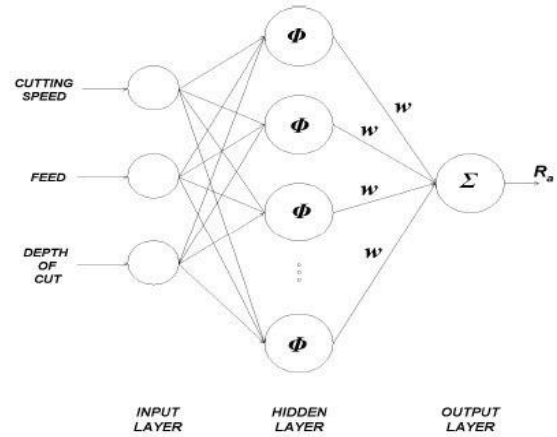


Fig.4. Radial Basis Function Network

**Feedback Networks**

**Hop Field Network**

Hopfield networks are single layer feedback networks. It's an auto associative fully interconnected network, moreover it's also symmetrically weighted network [28] It's of two types:

- Continuous.
- Discrete.

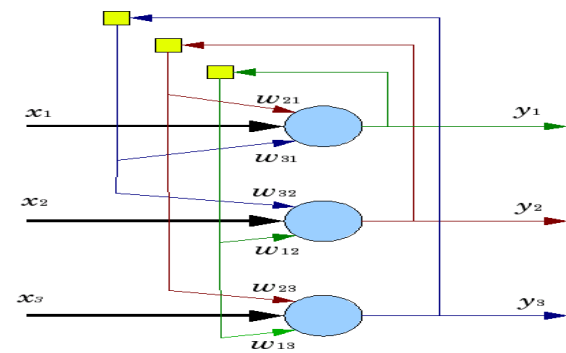


Fig.5. Hop Field Network

**B. Support Vector Machine**

These are also referred to as non-probabilistic binary linear classifiers. The basic property of SVM is the construction of such a hyperplane that separates the available dataset into two broad groups [20, 21].

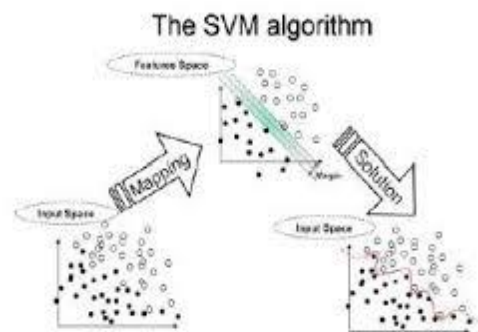


Fig.6. Support Vector Machine

C. Decision Tree

These type of networks are referred to as probabilistic models. The major feature that distinguishes these type of models from others is that it forms a tree like structure for any particular given problem. The questions are represented by the root and all the nodes. The possible answers to that particular question are represented by the arcs and the prediction of the solution is represented by the leaf node [10, 1, 7].

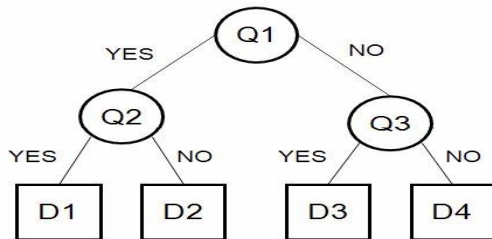


Fig.7. Decision Tree

D. Bayesian Belief Networks

These are another type of networks that forms a directed acyclic graph. It is used to represent probabilistic relation between the various variables. The uncertain variables are represented by the nodes whereas the relationship between these variables is represented by the arcs [15, 7].

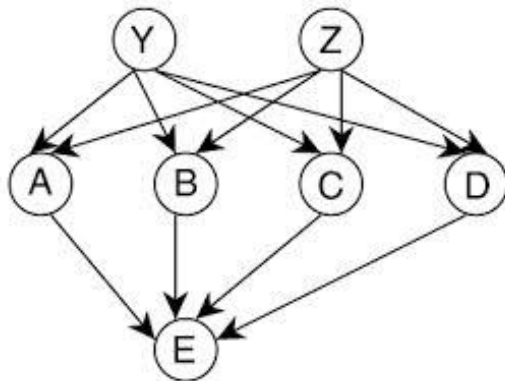


Fig.8. Bayesian Belief Network

E. Naïve Bayes Classifier

These types of networks make use of Bayesian theory or Bayes rule of conditional probability [16, 27]. All the attributes that are contained in the dataset are analyzed individually giving all of them equal importance.

F. K-Nearest Neighbour

The K-nearest neighbour algorithm's working is based on the distance factor that is calculated between the various set of scenarios. The working of KNN is described by a sequence of steps. Firstly we are given a reference scenario and some related scenarios. The next most important task is to calculate the difference between the given set and each of the rest available sets. Based

upon the value of the distance the scenarios are arranged in ascending order. From this list K members are chosen from the start.

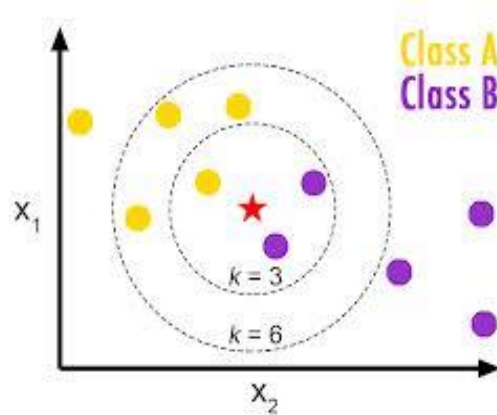


Fig.9. K-Nearest Neighbour

III. RELATED WORK

In this section we have talked about the remarkable work that has been done with respect to the Software Defect Prediction.

Malhotra *et al* [12] proposed a new technique using neural network to predict software defects. In this paper, the writer has used text mining, feature selection and Radial Basis Function to predict defects.

Gayathri *et al* [13] proposed a Bell function based Multilayer perceptron network whose performance was compared to various other ML techniques and this proposed system gave an accuracy of 98.2%.

Thwin *et al* [14] used ward network to predict defects and found out that this particular neural network is more accurate in predicting defects than other regression models.

Selvaraj *et al* [20] discussed the concept of Support Vector Machine's and compared its capability of predicting software defects with respect to Naïve Bayes model and Decision stumps. The results that they obtained showed that SVM performed better than the rest two approaches.

Malhotra *et al* [21] used the concept of SVM to predict the fault proneness of the software modules but found out that this technique is helpful only for a certain population of system.

Sehgal *et al* [10] explained the use of Decision tree models in defect prediction process. J48 algorithm of the Decision tree model was used for the same. Its performance was analyzed against the IDE algorithm and the earlier available Natural Growing Gas technique. When calculated J48 gave 93.32% accuracy whereas NGG technique gave only 80% accuracy.

Gayatri *et al* [11] suggested a new method i.e. the Decision tree induction method. This new way of feature selection proved to be a better way for feature selection when compared to existing techniques i.e. the RELIEF and the SVM. When this technique was used with Naïve

Bayes method or Random forest method it outperformed all the other techniques.

Tao *et al* [16] discussed the concept of Naive Bayes approach and compared the working of different methods of the Naive Bayes. The results showed that Multivariants Gauss Naive Bayes proved to be better than other types of Naive Bayes methods and J48 algorithm.

Liu *et al* [17] compared Bayesian network to Zimmerman approach and Naive Bayes approach. The Bayesian approach gave better results and provided 80% accuracy.

Boetticher *et al* [18] analyzed 2-NN (K-nearest neighbour) algorithm for the process of prediction and found out that 2-NN performance was unsatisfactory. It

was found that 2-NN algorithm gets diluted with large data sets.

Malhotra *et al* [25] talked about Machine learning and Logistic Regression (LR) approaches for defect prediction. All the research done by them pointed out that ML techniques are far better than LR techniques.

Al-Jamimi *et al* [4] compared two defect prediction approaches namely SVM and Perceptron NN. According to their work the PNN performed better than SVM. Sankar *et al* [5] talked about the various existing techniques available for defect prediction and they compared the working of two of them. SVM and Naive Bayes approaches were analyzed, the result of the analysis was in favor of Naive Bayes.

Table 1. An Analysis on Soft Computing Based Defect Prediction Techniques

S.N.	Reference	Techniques	Authors	Major Findings
01	[1]	SVM , FIS , GA	Punitha <i>et al.</i>	A new model that consisted of all the three techniques was proposed.
02	[3]	Software metrics	Wang <i>et al.</i>	This paper talked about the significance of software metrics in defect prediction and also proved that not more than 3 metrics are required for the same.
03	[4]	SVM , PNN	AL-Jamimi <i>et al.</i>	Comparison between SVM And PNN was shown and clearly PNN proved to be more useful.
04	[5]	SVM , NB	Sankar <i>et al.</i>	Comparison between SVM and NB was shown and clearly NB proved to be more useful.
05	[6]	LR , NB , NN , SVM , K* , DT , LDA , KNN	Wahono <i>et al.</i>	All these techniques were compared on the basis of performance and LR showed best results.
06	[9]	BAGGING CLASSIFICATION , NB , REGRESSION	Haghighi <i>et al.</i>	Bagging Classification which is an ensemble learning method had better results than the other two.
07	[10]	Decision Tree	Sehgal <i>et al.</i>	Compared J48 algorithm with ID3 algorithm and Natural growing gas technology, J48 performed better than the rest two.
08	[11]	Decision Tree	Gayathri <i>et al.</i>	A new feature selection method using the DT Induction method was proposed which when combined with NB and Random forest technique outperformed all the other techniques.
09	[12]	Neural Network	Malhotra <i>et al.</i>	RBF technique was used for prediction, and this technique was successful in predicting high severity bugs.
10	[13]	Neural Network	Gaythri <i>et al.</i>	A new bell function based Multilayer PNN was proposed
11	[14]	NN , Regression Model	Thwin <i>et al.</i>	Comparison between NN and Regression model was shown and clearly NN proved to be more useful.
12	[16]	Naive Bayes	Tao <i>et al.</i>	Compared various NB techniques and found out that Multivariants gauss NB technique was the best.
13	[17]	Bayesian Belief Network , Naive Bayes	Liu <i>et al.</i>	Compared NB with Bayesian belief network which showed that BBN gave 80% accuracy.
14	[20]	SVM , NB , Decision stumps	Selvaraj <i>et al.</i>	Comparison between these three techniques was done on the basis of their performance capacity and SVM showed outstanding results.
15	[25]	LR , ANN , DT , SVM	Malhotra <i>et al.</i>	Proved that ML techniques are better than LR techniques.

#### IV. RESULT ANALYSIS AND FUTURE SCOPE

A descriptive survey and analysis was performed to provide efficient results. The various techniques used for defect prediction in software systems can be arranged in order of their performance capacity. Neural Network being the best followed by Decision Tree and Bayesian Network, then the SVM and lastly comes the KNN method. It was also analysed that some areas of defect prediction techniques have not been explored namely the Apriori algorithm, the Fuzzy logic, and the Genetic algorithm. It is suggested that Ensemble Machine learning and One class SVM are two areas that can be used extensively in future. But according to our research SVM technique does not perform well so it's better to focus on Ensemble Learning in future to predict defects.

#### ACKNOWLEDGMENT

The authors wish to acknowledge the Management of Amity University Uttar Pradesh, Noida, India. We would like to express our sincere thanks to our Director (Dr. Abhay Bansal) for encouragement and guidance. The organization has provided research environment, and their faculties and scholars from the department of CSE, Amity School of Engineering and Technology, AUUP, Noida provided valuable suggestion during the analysis and research work.

#### REFERENCES

- [1] K. Punitha, Dr. S. Chitra, "Software Defect Prediction Using Software Metrics-A Survey," *IEEE International Conference on Information Communication and Embedded Systems*, pp. 555-558, 2013.
- [2] Y.Xia, G.Yan, Q.Si, "A Study on the Significance Of Software Metrics In Defect Prediction," *IEEE Sixth International Symposium On Computational Intelligence And Design*, pp. 343-346, 2013.
- [3] H.Wang, T.M.Khoshgoftaar, N.Seliya, "How Many Software Metrics Should Be Selected For Defect Prediction," *In Proceedings Of Twenty-Fourth International Florida Artificial Intelligence Research Society Conference*, pp. 69-74, 2011.
- [4] H.A.Al-Jamimi, L.Ghouthi, "Efficient Prediction Of Software Fault Proneness Modules Using Support Vector Machines And Probabilistic Neural Network," *IEEE Fifth Malaysian Conference In Software Engineering*, pp. 251-256, 2011.
- [5] K.Sankar, S.Kannan, P.Jennifer, "Prediction Of Code Fault Using Naïve Bayes And Svm Classifiers," *Middle-East Journal Of Scientific Research*, Vol. 20, No. 1, pp. 108-113, 2014.
- [6] R.S.Wahono, N.S.Herman, S.Ahmad, "A Comparison Framework Of Classification Models For Software Defect Prediction," *American Scientific Publishers*, Vol. 20, No. 10-11, pp. 1945-1950, 2014.
- [7] Er.R.Mahajan, Dr.S.K.gupta, R.K.Bedi, "Comparison Of Various Approaches Of Software Fault Prediction: A Review," *International Journal Of Advanced Technology & Engineering Research*, Vol. 4, No. 4, pp. 13-16, 2014.
- [8] P.Paramshetti, D.A.Phalke, "Survey On Software Defect Prediction Using Machine Learning Techniques," *International Journal Of Science And Research*, Vol. 3, No. 12, pp. 1394-1397, 2014.
- [9] A.A.S.Haghighi, M.A.Dezfuli, S.M.Fakhra, "Applying Mining Schemes To Software Fault Prediction: A Proposed Approach Aimed At Test Cost Reduction," *Proceedings Of The World Congress On Engineering*, Vol. 1, p. 415, 2014.
- [10] L.Seagal, N.Mohan, Dr.P.S.Sandhu, "Quality Prediction Of Function Based Software Using Decision Tree Approach," *International Conference On Computer Engineering And Multimedia Technologies*, pp. 43-47, 2012.
- [11] N.Gayathri, S.Nikolas, A.V.Reddy, "Feature Selection Using Decision Tree Induction in Class Level Metrics Dataset for Software Defect Predictions," *In Proceedings of the World Congress on Engineering and Computer Science*, Vol. 1, 2010.
- [12] R.Jindal, R.Malhotra, A.Jain, "Software Defect Prediction Using Neural Networks," *IEEE Third International Conference On Reliability, Infocom Technologies And Optimization*, pp. 1-6, 2014.
- [13] M.Gayathri, A.Sudha, "Software Defect Prediction System Using Multilayer Perceptron Neural Network With Data Mining," *International Journal Of Recent Technology And Engineering*, Vol. 3, No. 2, pp 54-59, 2014.
- [14] M.M.T.Thwin, T.S.Quah, "Application Of Neural Network For Predicting Software Development Faults Using Object Oriented Design Metrics," *In Proceedings Of The Ninth International Conference On Neural Information Processing*, p. 2312-2316, 2002.
- [15] N.E.Fenton, M.Neil, "A Critique of Software Defect Prediction Models," *IEEE Transactions on Software Engineering*, Vol. 25, No. 5, pp. 675-689, 1999.
- [16] W.Tao, L.Weihua, "Naïve Bayes Software Defect Prediction Model," *International Conference on Computational Intelligence and Software Engineering*, pp. 1-4, 2010.
- [17] Y.Liu, W.P.Cheah, B.K.Kim, H.Park, "Predict Software Failure-Prone By Learning Bayesian Network," *International Journal of Advanced Science and Technology*, Vol. 33, No. 10, pp 35-42, 2008.
- [18] G.D.Boetticher, "Nearest Neighbour Sampling for Better Defect Prediction," *ACM Journal*, Vol. 30, No. 4, pp.1-6, 2005.
- [19] T.Zimmermann, R.Premraj, A.Zeller, "Predicting Defects For Eclipse," *International Workshop On Predictor Models In Software Engineering*, 2007.
- [20] P.A.Selvaraj, Dr.P.Thangaraj, "Support Vector Machine for Software Defect Prediction," *International Journal of Engineering & Technology Research*, Vol. 1, Issue 2, pp. 68-76, 2013.
- [21] Y.Singh, A.Kaur, R.Malhotra, "Software Fault Proneness Prediction Using Support Vector Machines," *In Proceedings of the World Congress on Engineering*, Vol. 1, 2009.
- [22] Y.Weimen, L.Longshu, "A Rough Set Model for Software Defect Prediction," *International Conference on Intelligent Computation Technology & Automation*, pp. 747-751, 2008.
- [23] M.Liu, L.Miao, D.Zhang, "Two Stage Cost Sensitive Learning For Software Defect Prediction," *IEEE Transactions on Reliability*, Vol. 63, No. 2, pp. 676-686, 2014.
- [24] M.D.Ambros, M.Lanza, R.Robbles, "An Extensive Comparison of Bug Prediction Approaches," *IEEE*

*Seventh International Conference on Mining Software Repositories*, pp. 31-41, 2010.

- [25] R.Malhotra, "Comparative Analysis Of Statistical And Machine Learning Methods For Predicting Faulty Modules," *Elsevier Journal Of Applied Soft Computing*, Vol. 21, pp. 286-297, 2014.
- [26] [www.softwaretestingclass.com/top-10-0reasons-why-there-are-defects-in-software/](http://www.softwaretestingclass.com/top-10-0reasons-why-there-are-defects-in-software/) last accessed on 7/03/2015.
- [27] Gunjan Goswami, *Data Mining and Data Warehouse*, Katsons Publishers, New Delhi, Chapter 5, Chapter 6, Chapter 7, 2012.
- [28] S.N.Sivanandam and S.N.Deepa (2013). *Principles Of Soft Computing*, Wiley Publishers, Coimbatore, Chapter 3, Chapter 5, Chapter 6, Chapter 7, 2013.
- [29] Pradeep Kumar Singh, O.P. Sangwan and A. Sharma (2013). A Systematic Review on Fault Based Mutation Testing Techniques and Tools for Aspect-J Programs, published in 3<sup>rd</sup> IEEE International Advance Computing Conference, IACC-2013 at AKGEC Ghaziabad, India, 22-23, February 2013, IEEE Xplore, pp. 1455-1461.

### Authors' Profiles



**Kritika Verma** is a student pursuing Masters in Computer Science and Engineering at the Amity School of Engineering and Technology, Amity University, Uttar Pradesh, Noda, India. She has published 2 papers in International Conferences and Journals of repute.



**Dr. Pradeep Kumar Singh** is an Assistant Professor in Computer Science & Engineering department at the Amity School of Engineering and Technology, Amity University, Uttar Pradesh, Noida India. He is member of ACM, CSI and many professional bodies. He has published more than 25 papers in International Conferences and Journals of repute with Scopus, ISI Indexed repositories.