AN IMPROVED SELF ORGANIZING MAP USING JACCARD NEW MEASURE FOR TEXTUAL BUGS DATA CLUSTERING

ATTIKA AHMED

A thesis submitted in
fulfillment of the requirement for the award of the
Degree of Master of Information Technology

Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia

JANAUARY, 2018

This research work I dedicated to my late mother, who passed away on April 2016 and to my brother Syed Nabeel Ahmed.

ACKNOWLEDGEMENTS

In the name of Allah, Most Gracious, Most Merciful. All Praise is to Allah, the Cherisher and Sustainer of the worlds. Oh Allah, send Grace and Honor on Prophet Muhammad (S.A.W) and on the family and true followers of Prophet Muhammad (PBUH).

I would like to express my appreciation and gratitude to my down to earth supervisor Associate Professor Dr. Rozaida Binti Ghazali for her motivation and support to pursue my Masters. I would also like to thank her for the providing me the grant for publishing my paper in IEEE. I am indeed very grateful. I would also like to thank my family for their support and motivation through the difficult times in during my Masters research work journey. And to the faculty members especially FSKTM postgraduate student's society, I say thank you. Thank you for the good company and friendship you have provided me during my Masters journey.

Finally, I would like thank to my very good friends Abdulkadir Hassan Disina and Dr. Zaharuddin Pindar for their unconditional support specially for helping me in writing research papers and to University Tun Hussein Onn Malaysia for providing the supportive environment to conduct my Masters research.

ABSTRACT

In software projects there is a data repository which contains the bug reports. These bugs are required to carefully analyze to resolve the problem. Handling these bugs humanly is extremely time consuming process, and it can result the delaying in addressing some important bugs resolutions. To overcome this problem researchers have been introduced many techniques. One of the techniques is the bug clustering. For the purpose of clustering, a variety of clustering algorithms available. One of the commonly used algorithm for bug clustering is K-means, which is considered a simplest unsupervised learning algorithm for clustering, yet it tends to produce smaller number of cluster. Considering the unsupervised learning algorithms, Self-Organizing Map (SOM) considers the equally compatible algorithm for clustering, as both algorithms are closely related but different in way they were used in data mining. This research attempts a comparative analysis of both the clustering algorithms and for attaining the results, a series of experiment has been conducted using Mozilla bugs data set. To address the data sparseness issue, the experiment has been performed on textual bugs' data by using two different distance measure which are Euclidean distance and Jaccard New Measure. The research results suggested that SOM has a limitation of poor performance on sparse data set. Thus, the research introduced the improved SOM algorithm by using a Jaccard NM (SOM-JNM). The SOM-JNM produced significantly better results therefore; it can be consider a challenging approach to address the sparse data problems.

ABSTRAK

Dalam projek perisian terdapat satu simpanan data yang mengandungi laporan pepijat. Pepijat ini diperlukan untuk menganalisis dengan teliti bagi menyelesaikan masalah ini. Mengawal pepijat ini secara semulajadi memakan masa yang sangat lama, dan ia boleh menyebabkan kelewatan dalam menyelesaikan beberapa isu penting pepijat. Untuk menyelesaikan masalah ini, penyelidik telah diperkenalkan dengan pelbagai teknik. Salah satu dari teknik ini adalah pengelasan pepijat. Untuk tujuan pengelasan, pelbagai algoritma tersedia. Salah satu algoritma yang biasa digunakan untuk pengelasan pepijat adalah K-means, yang mana ia dianggap sebagai algoritma pembelajaran tanpa pengawasan yang paling mudah untuk teknik pengelasan, namun ia cenderung untuk menghasilkan bilangan kelas yang lebih sedikit. Memandangkan algoritma ini tidak diawasi, Peta Penyusun Sendiri (SOM) boleh dianggap algoritma yang sesuai untuk pengelasan, kerana kedua-dua algoritma ini berkait rapat tetapi berbeza dengan cara mereka digunakan dalam perlombongan data. Kajian ini cuba untuk membandingkan kedua-dua kelas algoritma dan untuk mencapai keputusan yang lebih tepat, satu siri eksperimen telah dijalankan menggunakan set data pepijat Mozilla untuk menangani isu kejarangan data. Eksperimen telah dilakukan pada data pepijat berbentuk teks dengan menggunakan dua jarak yang berbeza iaitu jarak Euclidean dan Jaccard New Measure. Hasil kajian mencadangkan bahawa SOM mempunyai batasan prestasi yang lemah pada set data jarang. Oleh itu, kajian memperkenalkan algoritma SOM yang lebih baik dengan menggunakan Jaccard NM (SOM-JNM). SOM-JNM telah menghasilkan keputusan yang jauh lebih baik. Oleh itu, SOM-JNM boleh dipertimbangkan sebagai pendekatan yang baik untuk menangani data jarang.

.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIX

LIST OF PUBLICATIONS

i. Ahmed, A., & Ghazali, R. (2016, October). An improved self-organizing map for bugs data clustering. In *Automatic Control and Intelligent Systems (I2CACIS), IEEE International Conference on* (pp. 135-140). IEEE.

CHAPTER 1

INTRODUCTION

1.1 Research Background

The word ‘bug’ actually is short for Bugbear. Sometimes found as Bugaboo. Its meaning is much closer to ‘Gremlin’, “where the people who worked on engineering prototypes often grew to suspect that the problems were due to malicious spooks” (Guo *et al*., 2011). In 1947 at the computational laboratory of Harvard faculty a moth trapped in a relay which caused an error in the Mark II computer, this bug carefully removed and taped to the log book which still exists (Guo *et al*., 2011). Since then the term “bugs” established for software error.

During the software development or maintenance process the bugs introduce due to many reasons such as misunderstanding of software architectural design, poor strategy of design, implementation, error handling methodology missing in coding, continuous changes in code after development, etc. In software projects, there is a database which use as bug repository since it is used to collect and manage the bugs reports from users and developers as well as with any other technical support or team member. These bugs are required to analysis very carefully that whether the bugs are valid or invalid (occurred due to operating system, but not by product), duplicate or unique (reported earlier or not), important or unimportant (are they really bugs or something else) and who will resolve it. This process is called bug triaging and the person who performs this job is called triage. The trigger manage the bugs repository in a way that only real bugs should contain and so that the important bugs can be addressed quickly. Every day large number of bugs reports, according to Jadhav *et al*.,(2015) almost 20~30 reports received in software system on an average per day (Jadhav *et al*., 2015). However, in larger projects such as Mozilla (Mozilla, 2010) more than 300 bugs reports on an average per day (Jadhav *et al*.,2015). Handling

these bugs humanly is extremely time consuming process and it can result the delaying in addressing some important bugs resolutions.

To resolve this problem, bugs reports contain a field where the reporter asked to assign bug priority.The reporter needs to report based that how important it is, but sometimes the field left empty or the reporter may not assign the correct priority. As its possible that the reporter opinion about the bug priority is different with the triager. Since the reporter can be a naïve user of the software and the level of software knowledge is insufficient while the triager has more information of the software as a whole. To assign the correct priority to the bug reports, traiger analyized the content of report using his own knowledge as well which he gained during triagging and than send it to the developer where the developer assigned the priority to the bug depends on its seviourity.

As the bugs testing is one of the important aspects of software maintenance and testing, research has been growing significantly, but still bugs are one of the major concerns for developers. This study focuses on providing the alternate to the developers in order to minimize the bugs' problem. This can be significantly useful as it provides the better accuracy and purity to coup with the duplicate bugs' issues and weak bugs' prioritization problem.

1.2 Problem Statement

Software bugs have created much concern among the researchers due to catastrophic consequences which led to many life claiming incidents and financial losses. Significantly the research has been grown in software bugs resolving issue but still bugs are a major concern for both researchers and developers. Considering the conventional approach to overcome the bugs' problems, one of the strategies follows is called reassignment. Developers often used the bug tracking system in order to find the expertise in search of queries, which is the best to resolve the issue in the particular software or sub module so that the task can be reassign to him/her. It is simply because lack of time to investigate deeply the bug and genuine attempt to find a person with better expertise. It happens mostly because of five reasons; finding the root cause, determining ownership, poor bug report quality, hard to determine proper fix, and workload balancing. As the number of reassignment increase, the time

required to fix the bug increase too. On contrary to popular belief, reassignments aren't necessarily 'bad', since it does take a few reassignments to find the true cause of a bug and who to properly fix it. On the other hand, if there were few reassignments but the optimal bug fixer was not identified, then that could lead to a low quality or faulty fixings (Guo *et al*., 2011). These reassignments can be happens not only for the search of expertise in that particular software or sub module, but also because of disagreements between developer teams.

As the reassignment strategy if not failed, but challenge due to time consuming issue, researchers stress for an alternative approach. Over the last decades there has been some encouraging research for addressing this issue, but still bug resolving issue a big concern for the software industry.

One of the non-conventional approaches to resolve the bugs' issues used by researchers are Neural Networks (NN), often refers as Artificial Neural Networks (ANN). NN is the mathematical alternative approach inspired by the biological nervous system. The NN has adaptability and can handle complex system same like human brain such as images, pattern, speech recognition, etc. NN is the non liner statistical data modeling tool (Mishra, 2015) used by researchers for bugs' data clustering. The advantage of NN is that it is adaptive and does not need any understanding of input data. They are self-adaptive techniques. The drawback is that NNs are not easy to represent and fewer statistical techniques can be applied (Zeng & Rine, 2004).

One of the statistical approaches is K-means. Russ *et al*.,(2009) chose the bug cluster to experiment on the fact that they describe the same defect "We regard a bug and its duplicates as a cluster" (Russ*et al.*, 2009). This is one of the issues among the bug clustering problems. To avoid the bug duplication researchers emphasize on bug prioritization. Bug prioritization strategy is very helpful to address the bugs' issues. Many clustering algorithms including K-means, used for bug prioritization but the problem arise when the algorithms space complexity increases due to data sparseness (NRC, 2012).

Therefore, this research proposed the Self-Organizing Map using Jaccard New Measure (SOM-JNM) which is a type of NN. SOM is considered good for large data clustering without any external training. Beside the advantage of SOM over other algorithms to cluster the large data set, the main disadvantage of SOM is that it is a slow learner, as some researcher describe it "Time consuming algorithm, this is

because as the number of neurons activation at same time slower the performance of the algorithm. And as the number increases the computation increases which results in increasing computational time (Patole *et al*., 2010) . The main reason of being time consumption is the limitation of Euclidean distance of working on multi-dimensional or sparse data, high sensitivity to noise and outliers (especially for sparse data),which results the too much time and resources consumption. Due to Euclidean distance limitation of computing the distance on sparse data set, this research suggested to use another distance measure Jaccard New Measure (Jaccard NM). Since the Jaccard NM provides a room to address the data sparseness issue and to compute the distance with null values. It motivates this research to substitute the Euclidean distance with Jaccard NM. By combining the properties of SOM with Jaccard NM, the clustering ability of the algorithm enhanced and overcome the weak performance over the sparse data set.

1.3 Aim and Objectives

The aim of this research is to cluster bug data using the proposed SOM-JNM. Therefore in order to achieve the research aim, few objects have been set:

(i) To propose an improved Self Organizing Map (SOM) by adopting a new distance measure, the Jaccard NM.

(ii) To simulate (i) on bug data.

(iii) To evaluate (ii) based on its cluster purity and accuracy, and benchmark it against the ordinary SOM with other clustering technique K-means.

1.4 Scope of Research

The research focuses on improvising SOM by adopting a new distance measure Jaccard NM and to implement SOM-JNM for clustering the Bugzilla Mozilla data (Mozilla , 2010). Mozilla data has been used and the preprocessing has been applied to the data in order to bring the data in a form where clustering can be applied. To simulate the proposed SOM-JNM, bug data from Bugzilla Mozilla data has been normalized and used to evaluate the simulation based on the cluster purity and accuracy. The simulated results compared against the ordinary SOM with other

benchmark algorithms K-means. To evaluate the performance, the research focuses on internal and external measures which are Silhouette, Homogeneity, Separation, Adjusted Rand, Jaccard and Fowlkes Mellow respectively.

1.5 Significance of Research

The research provides an improved approach to solve the bug issues facing by the software industry. SOM is considered one of the best approach for categorize and textual data both as it has the quality of self-organizing against the large data and cluster it. Thus, this research can be beneficial for addressing the bugs' issues into large data repository and the bug prioritization for large data repository will become less time consuming and less costly. Moreover, the study provides the steady and reliable approach for addressing the database bug repository issues to deal in a proper manner so that it will be beneficial for developer and user both as well as the technical support team. On the larger scale it has been beneficial for the software industry to decrease the financial cost needed to spend on bugs resolving issues at the same time for the economy in general.

1.6 Thesis Organization

This thesis has been organized in 5 chapters. The first chapter discusses the existing research gaps in using SOM along with aim and objective of this research work. The second chapter is dedicated to the literature review related to the previously done work over SOM and K-means clustering. The research methodology of this research work has been discussed, to improvise the ordinary SOM by adopting a Jaccard NM and introduce SOM-JNM in third chapter. Fourth chapter is about the experimental results of this research work, where all the experiment and its results discussed in detail. The final chapter of this thesis is chapter five, which is the discussion over the research novelty and the objectives achieved in this research work. It also discussed this research work limitations and the future recommendation for the relevant research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Over the last fifteen years the interest in Artificial Intelligence (AI) in the computer science field has been immensely increased and some very amazing applications of bioscience and engineering field have been designed based on AI. AI basically deals with intellectual mechanism, which includes Neural Network (NN), offers a great advantage over conventional modeling, including the ability of large amount of noisy data from dynamic and nonlinear processes where nonlinearities and variables plays a vital role. NN's are at the forefront of computational systems designed to produce, or at least mimic, intelligent behavior. Unlike classical AI systems that are designed to directly emulate rational, logical reasoning, neural networks aim at reproducing the underlying processing mechanisms that give rise to intelligence as an emergent property of complex, adaptive systems (Heger, 2005). The NN has adaptability and can handle complex system same like human brain; such as image, pattern, speech recognition, etc. In AI most tasks that require intelligence, require an ability to induce new knowledge from experiences. Thus, a large area within AI is machine learning. Machine learning involves the study of algorithms that can extract information automatically. Similarly to AI, machine learning is very broad and can include almost everything, as long as there is some inductive component to it. Data mining is an area that has taken much of its inspiration and techniques from machine learning. Data mining is carried out in specific situation, on a particular data set, with a goal in mind. Moreover, data mining procedures could be either unsupervised

(when the answer is unknown-discovery) or supervised (the answer is known-prediction). Since the goal and data set for data mining can be different so the techniques also varies to extract the results. Common data mining techniques are including clustering, classification, regression trees, and NN (Han & Kamber, 2000). Classification is the task where the answer is known and called as supervised learning, while clustering is the task which is unsupervised learning because the answer is unknown. Clustering procedure used different algorithms to perform the clustering task. One of the commonly used clustering algorithms is K-means (Bação, Lobo, & Painho, 2005). Other than K-means, SOM is another algorithm used for large data clustering. Beside its limitation of slow learning, SOM considered as good algorithm for large data clustering (Patole *et al*., 2010).

Thus, this research is focused the bugs data clustering which is usually a large data set and the poor clustering results would have negative effect when the bugs prioritization applied to that data set. This chapter also discussed the related works that in line with the problem under study, the bug data clustering.

2.2 Clustering

Clustering is the division of data into groups of similar objects. In clustering, some details are marginalized in exchange for data simplification. Clustering can be regarded as data modeling technique that offers concise summaries of the data. Clustering is therefore related to many disciplines and plays an essential role in a broad range of applications. The set of different balls has been used to elaborate the clustering process. The data set before clustering as shown in Figure 2.1(a), while Figure 2.1(b) is the visualization of the after clustering where, three types of balls has been clustered.

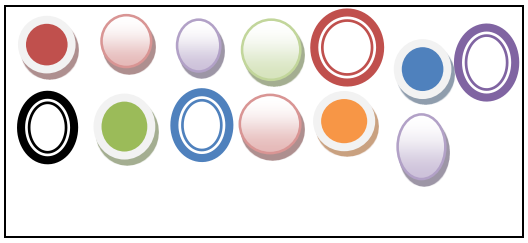

Figure 2.1(a): Before clustering, the set of round shape balls.

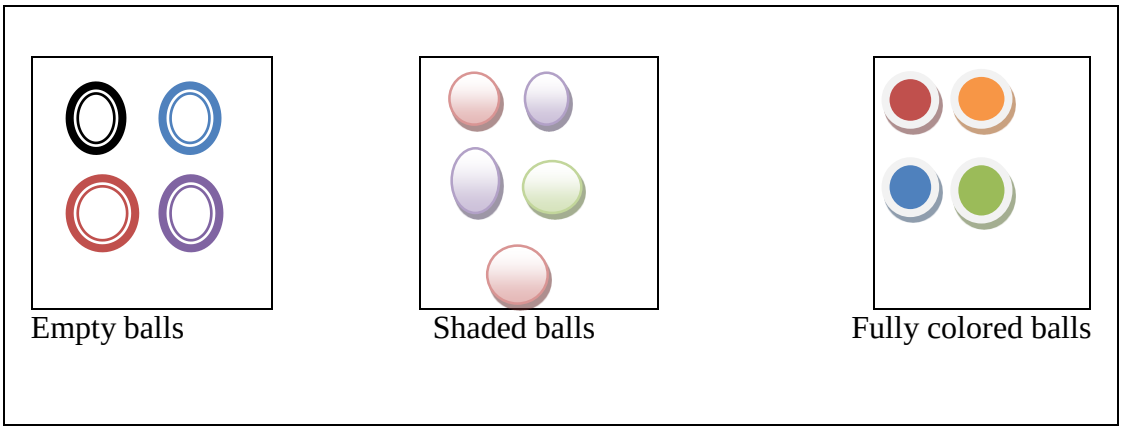

Figure 2.1(b): After clustering the empty, shaded and fully colored balls grouped.

Clustering algorithms are required to classify some conditions in order to be considered efficient algorithms. These conditions are as follows:

1) Data must be scalable otherwise it may get the wrong result.
2) Clustering algorithm must be able to deal with different types of attributes.
3) Clustering algorithm must be able to find clustered data with the arbitrary shape.
4) Clustering algorithm must be insensitive to noise and outliers.
5) Clustering algorithm must be able to deal with data set of high dimensionality.

Clustering algorithms broadly classified into two categories: unsupervised linear clustering algorithms and unsupervised non-linear clustering algorithms.

2.3 Unsupervised Linear and Non-linear Clustering Algorithm

In clustering, unsupervised linear algorithms are usually based on sequential search where the data is linearly separable. For instance, the points belonging to cluster 1 can be separated from the points belonging to cluster 2 by a hyper plane. While in nonlinear algorithms, due to increased data dimension the computation required between two data points. Some of the examples of theses algorithms are listed in Table 2.1.

REFERENCES

Bação, F., Lobo, V., & Painho, M. (2005). Self-organizing Maps as Substitutes for K-Means. *Springer* , 476 – 483.

Bakos, Y. J. (2010). *Mining Similarity Using Euclidean Distance, Pearson Correlation, and Filtering*. Retrieved dec 7, 2015, from humanoriented.com: http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/mvoget/similarity/similarity.html

Begum, M., & Akthar, M. N. (2013). KSOMKM: An Efficient Approach for High. *International Journal of Electrical Energy*.

Berglund, E., & Sitte, J. (2013). The Parameter-Less Self-Organizing Map. *IEEE* , 305-316.

Bloehdorn, S., &Blohm, S. (2006). A self organizing map for relation extraction from wikipedia using structured data representations. Paper presented at the Proc. Int. Workshop on Intelligent Information Access (IIIA-2006).

Bnanankhah, A., & Nejadkoorki, F. (2012). Artificial Neural Networks: A Non-Linear Tool. *International Conference on Applied Life Sciences (ICALS2012)* (pp. 81-85). Turkey: INTECH.

Boroš, M. (2011, may 24). *Cluster Analysis*. Retrieved june 16, 2016, from let.rug.nl: http://www.let.rug.nl/nerbonne/teach/rema-stats-meth-seminar/presentations/Boros-Clustering-2011

Cavazos, T. (2000). Using Self-Organizing Maps to Investigate Extreme Climate Events: An Application to Wintertime Precipitation in the Balkans. *AMS* .

CBS, Department of Systems Biology. (2015). Retrieved from http://www.cbs.dtu.dk/courses/27618.chemo/27618_lecture_clustering_part2.pdf

Christian, S. P. (2013). *Machine Learning :: Cosine Similarity for Vector Space Models.* Retrieved dec 7, 2015, from pyevolve:

http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/

Edhlund, B., & McDougall, A. (2015). *NVivo 10 Essentials.* griffith: Amazon.com.

Fry , Z. P., & Weimer, W. (2013). Clustering Static Analysis Defect Reports to Reduce Maintenance Costs. *IEEE* , 282-291.

Ghaemmaghami, F., & Sarhadi, R. M. (2013). SOMSN: An Effective Self Organizing Map for Clustering. *International Journal of Computer Applications*, (0975 – 8887).

Ghahrai, A. (2008, November 9). *Defect Clustering in Software Testing.* Retrieved may 7, 2015, from testingexcellence.com: http://www.testingexcellence.com/defect-clustering-in-software-testing/

Guo, P. J., Zimmermann, T., Nagappan, N., & Murphy, B. (2011). Not my bug! and other reasons for software bug report reassignments. Paper presented at the Proceedings of the ACM 2011 conference on Computer supported cooperative work.

Han, J., & Kamber, M. (2000). *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers.

Heger, D. A. (2005). An Introduction to Artificial Neural Networks(ANN)-Methods, Abstraction, and Usage. Retrieved from http://www.dhtusa.com/media/NeuralNetworkIntro.pdf

Herbert, J., & Yao, J. T. (2007). GTSOM: GAME THEORETIC SELFORGANIZING. *Springer* , 199-224.

Hynar, M., Burda, M., &Sarmanova, J. (2005). Unsupervised clustering with growing self-organizing neural network--a comparison with non-neural approach. Paper presented at the DATESO.

Ishii, M., Shimodate, T., Kageyama, Y., Takahashi, T., & Nishida, M. (2012). Quantification of Emotions for Facial Expression: Generation of Emotional Feature Space Using Self-Mapping. In *Applications of Self-Organizing Maps* (p. 298). InTech, Chapters.

Jadhav, A., Jadhav, K., Bhalerao, A., & Kharade, A. (2015). A Survey on Software Data Reduction Techniques. *IJCSIT*, 4611-4612.

Kanwal, J., & Maqbool, O. (2012). Bug Prioritization to Facilitate Bug Report Triage. *JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY* .

Kaur, M., & Garg, S. K. (2014). Survey on Clustering Techniques in Data Mining for Software Engineering. International Journal of Advanced and Innovative Research, 3, 238-243.

Kohonen, T. (2012). Self-organization and associative memory (Vol. 8): Springer.

Kriesel, D. (2007). *A Brief Introduction to Neural Networks.*Retrieved August, 15, 2011.Bonn, Germany.

Liu, B., Xia, Y., & Yu, P. S. (2005). Clustering Via Decision Tree Construction Foundations and advances in data mining (pp. 97-124): Springer.

Lobo, V. J. (2009). Application of Self-Organizing Maps. In *Information Fusion and Geographic Information Systems* (pp. 19-36). Springer.

McCulloch, W. S., & Pitts, W. (1990). A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biology, 52(1-2), 99-115.

Mehta, P. (2016). *Karl Pearson's Formula for Finding the Degree of Correlation.* Retrieved 2016, from economicsdiscussion.net: http://www.economicsdiscussion.net/correlation/karl-pearsons-formula-for-finding-the-degree-of-correlation/2579

Mozilla . (2010). Retrieved from Mozilla: Mozilla. http://www.mozilla.org, 2010.

Naseem, R., Maqbool, O., & Muhammad, S. (2010). An improved similarity measure for binary features in software clustering. Paper presented at the Computational Intelligence, Modelling and Simulation (CIMSiM), 2010 Second International Conference on.

Naseem, R., Maqbool, O., & Muhammad, S. (2011). Improved similarity measures for software clustering. Paper presented at the Software Maintenance and Reengineering (CSMR), 2011 15th European Conference on

NRC, N. R. (2012). Model Validation and Prediction. In *Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification* (pp. 52-83). The National Academic Press.

Patel, K., Sawant, P., Tajane, M., & Shankarmani, D. R. (2015). BUG TRACKING AND PREDICTION. *nternational Journal of Innovative and Emerging Research in Engineering* , 174-179.

Patole, M. V., Pachghare, M. V., & Kulkarni, D. P. (2010). Self Organizing Maps to Build Intrusion Detection System. *International Journal of Computer Applications* .

Prajapat, R. (2015, February 24). *DnI Institute*. Retrieved May 14, 2015, from http://dni-institute.in: http://dni-institute.in/blogs/neural-network-tutorial/

Rosenberg, A., & Hirschberg, J. (2007). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. *EMNLP-CoNLL Vol. 7*, (pp. 410-420). paruge.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Elsevier* , 53-65.

Rus, & Shiva. (2009). Clustering of Defect Reports Using Graph Partitioning Algorithms. 442-445.

Russo, T., Scardi, M., & Cataudella, S. (2014). Applications of Self-Organizing Maps for Ecomorphological Investigations through Early Ontogeny of Fish. *PLOS* .

Saimadhu. (2015, april 4). *Five most popular similarity measures implementation in python*. Retrieved march 2016, from dataaspirant: http://dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/

Sathya, R., & Abraham, A. (2013). Comparison of Supervised a. *(IJARAI) International Journal of Advanced Research in Artificial Intelligence*, , 34-38.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *elesvier* , 85–117.

Stefnoic, P., & Kurasova, O. (2011). Influence of learning rate and nebghiouring functions on Self Organizing Map. In *Advances in Self Organizing Map* (pp. 141-150). Springer Berlin Heidelberg.

Weber, M., Teeling, H., Huang, S., Waldmann, J., Kassabgy, M., M Fuchs, B., et al. (2011). Practical application of self-organizing maps to interrelate biodiversity and functional data in NGS-based metagenomics. *The ISEM Journal* , 918-928.

Xu, R., &Wunsch, D. (2005). Survey of clustering algorithms. Neural Networks, IEEE Transactions on, 16(3), 645-678.

Zeng, H., &Rine, D. (2004). Estimation of software defects fix effort using neural networks. Paper presented at the Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International.