Functional Requirement Classification by Clustering and Artifical Neural Network with Genetic Algorithm

Neha Dhiman, Er Mohan Singh
Deptt. Of Computer Science & Engg. HPTU Hamirpur
E-mail- nehadhimanneha333555@gmail.com
Assistant Professor HPTU Hamirpur
E-mail- mcamohanchoudhary@gmail.com

Abstract— The suggested method for extracting Functional requirement from the SRS by using concept of clustering with neural genetic approach. Extracting Functional requirement is a different task for judgment a pattern in the Functional requirement that is communicated as regular expression. In the contribution of SRS. We proposed and implement pre-processing of SRS by text mining and classified by cluster based supervised leaning. We reduce the complexity and increase the accuracy Functional requirement extraction by using text from SRS and then clustered then classified by Cluster based supervised learning. It targets at providing an automatic supervised Functional requirement content mining based on mode based structure that dividers the SRS documents when a shared pattern is found. We could detect and remove local noises that rendered the Functional requirement by non-functional requirement into well-formed SRS documents that enhanced the extraction process. We analysed the proposed approach by precision, recall, accuracy and F1 measure.

Keywords- Accuracy, *Clustering, Recall, Precision, SRS*

I. INTRODUCTION

"Intelligence is the ability to adapt to change"- Stephen Hawking. This quote rightly highlights the importance of adapting to changing environment. Every software goes through certain activities under software development life cycle, and Agile methods have emerged as one of the most popular means of software development in recent times due to their ability to adapt to changing requirements and changing environment; thus, making these methods apt for present day business scenario. According to 10th annual agile survey [6]. However, uniform customer involvement throughout the project, appointing appropriate team to adapt to agile methodology, ranking of changes to be accommodated in software, maintaining simplicity, difficulty in scaling agile procedures to larger projects and deciding upon the contract terms account for the major disadvantages involved in the agile development[7]. Large systems involve large separate teams working on separate interacting systems may be at different locations. Large systems also have external development restrictions and rules which limit their extent of flexibility. Also, managing large systems and involving various stakeholders of large projects is different. Adapting agile development for larger systems involves scaling up process which needs attention towards design documentation, arranging communication and meetings among different teams and careful maintenance of continuous integration of builds. Scaling out is required to introduce agile methodology in large organizations (following traditional approaches) and making agile methods compatible with them [3]. As compiled by agile approach provides benefits to both development team as well as the client by addressing project drawbacks like schedule predictability, scope creep and budget control etc [1]. Machine learning is basically a subfield of artificial intelligence which lends computers the learning ability without explicitly programming them. This involves development of those computer programs which have the ability of teaching themselves for growing and changing when they are exposed to fresh data [2]. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages [3].

II. LITERATURE REVIEW

Harleen K. Flora *et.*al [1] this paper explains the values and principles of ten agile practices that are becoming more and more dominant in the software development industry. Agile processes are not always beneficial, they have some limitations as well, and this paper also discusses the advantages and disadvantages of Agile processes. Agile development methodology is a conceptual framework for undertaking any software engineering project. In general agile approach can provide a shorter development cycle, higher customer satisfaction, and quicker adaptation to rapidly changing business requirements. It also attempts to minimize risk and maximize productivity by delivering working software in short iterations that increases the internal and external practices of the software development. A selection and implementation of appropriate process is crucial as it ensures the organization to maximize their chance to deliver their software successfully with long term implications.

Gaurav Kumar *et.*al[2] in this paper identify the impacts that agile methodology has on software development processes with respect to quality within the organizational, methodical, and cultural framework. Agile software development stresses in -evolving requirements accomplished by direct user involvement in the development process, rapid iterations, small and frequent releases. The improvements in software development process include more stable requirements, earlier fault detection, less lead times for testing, increased communication, and increased adaptive capacity. Different methodologies require different changes to the management and software development cultures. There are number of factors that can directly and indirectly influence the development projects in agile framework. Adopting agile development methodologies has a positive impact on both the productivity and the quality. Hence, development team and customer both are satisfied with its implementation in software development processes.

Sheetal Sharma *et.*al[3] This paper deals with the comparative study of agile processes. The paper will serve as guide to other software development process models. Agile processes have important applications in the areas of software project management, software schedule management, etc. In particular the aim of agile processes is to satisfy the customer, faster development times with lower defects rate. This paper compares the agile processes with other software development life cycle models. Agile processes are not always advantageous, they have some drawbacks as well; the advantages and disadvantages of agile processes are also discussed in this paper. In the comparative study of agile software development with other software development models it conclude that agile project is much better than other software development process in terms of productivity, performance, faster time cycles, risk analysis. Agile processes are implemented in important applications such as web based, testing tools, etc.

DAVID COHEN *et.*al [4]The aim of this paper is to introduce the reader to agile methods allowing him/her to judge whether or not agile methods could be useful in modern software development. The chapter discusses the history behind agile methods as well as the agile manifesto, a statement from the leaders of the agile movement. It looks at what it means to be agile, discusses the role of management, describes and compares some of the more popular agile methods, provides a guide for deciding where an agile approach is applicable, and lists common criticisms. It summarizes empirical studies, anecdotal reports, and lessons learned from applying agile methods and conclude with an analysis of various agile methods. The target audiences for this chapter include practitioners, who will be interested in the discussion of the different methods and their applications, researchers who may want to focus on the empirical studies and lessons learned, and educators looking to teach and learn more about agile methods.

Merisalo-Rantanen Hilkka *et.*al[5]This paper explores Extreme Programming (XP) as an information systems development approach and argues that it is mainly old wine in new bottles. It takes an interpretive and critical view of the phenomenon. It made an empirical study of two companies that apply an XP style development approach throughout the information systems development lifecycle. The results of research suggest that XP is a combination of best practices of traditional information systems development methods. It is hindered by its reliance on talented individuals, which makes its large scale deployment as a general purpose method difficult. It claims that XP can be useful for small teams of domain experts, who are physically close to and able to communicate well with the endusers and who are good designers and implementers. However, these skilled and motivated individuals with high working moral can exhibit high productivity regardless of the methods used, if they are not overly constrained by bureaucracy.

III. METHODOLOGY

Step 1: Documents containing various requirements specifications (functional as well as non-functional requirements) are collected from various sources.

Step 2: These documents are preprocessed to generate the required text for computer analysis. This preprocessing through text mining prepares the text for computer understandable representation and extracts the necessary, useful matter from the text. This is carried out in three steps:

Tokenization: It converts a stream of characters into tokens (generally word tokens). Delimiters like space, punctuations, etc. are used for separating one-word token from another.

Stop-word Removal: Words that carry negligible or little meaning for the sentences like

‘is’, ‘of’, ‘and’, ‘the’ are removed.

Stemming: Words in the word stem are reduced to their root form like ‘writes’, ‘writing’, ‘wrote’, ‘written’ these all correspond to a single root “write”.

Step 3: Once the text is preprocessed, the document term weight is represented as Term Frequency- Inverse Document Frequency which evaluates the importance of a document word in a collection of documents. Terms that capture the document essence are more frequent and have high Term Frequency whereas good terms that discriminate a document from others occur fewer in the documents and have high Inverse Document Frequency. Dot product is taken of both to calculate a final Term Frequency- Inverse Document Frequency.

Step 4: Clustering is used to partition the data into sets of similar items. Each of the sets is called a cluster. This process aims at increasing cohesion in a single cluster and minimizing coupling between two or more clusters that is, trying to reduce the intra-cluster distance and increase inter-cluster distance. The clustering technique used in the methodology is a hybridization of a flat and hard clustering algorithm K-means with an agglomerative bottom-up hierarchical clustering method UPGMA. Simplicity and efficiency of K-means is used to produce a large number of clusters and then UPGMA is applied to refine these clusters so that their quality is enhanced.

Step 5: Supervised learning is applied to implement two classifier models:

SVM with RBF kernel: SVM can only perform linear classification. RBF kernel is combined with SVM to provide the capability of non-linear classification. This allows the algorithm to fit the maximum margin hyperplane in a transformed feature space.

Neural network with genetic algorithm: Use of the genetic algorithm helps to reduce the error of neural network.

Step 6: Results from the above two classifier models are then analyzed and compared based on Precision, Recall, and Accuracy which are calculated by the

Formulae mentioned below:

$$\text{Precision} = \frac{\sum\text{True Positive}}{\sum\text{Predicted Positive}} = \frac{\sum\text{True Positive}}{\sum\text{True Positive}+\sum\text{False Positive}}$$

$$\text{Recall} = \frac{\sum\text{True Positive}}{\sum\text{Actual Positive}} = \frac{\sum\text{True Positive}}{\sum\text{True Positive}+\sum\text{False Negative}}$$

$$\text{Accuracy} = \frac{\text{Correctly predicted observations}}{\text{Total number of events}} = \frac{\sum\text{True Positive} + \sum\text{True Negative}}{\sum\text{True Positive} + \sum\text{False Positive} + \sum\text{True Negative} + \sum\text{False Negative}}$$

where, True Positive is correctly predicted positive, False Positive is falsely predicted positive, True negative is correctly predicted negative and False Negative is falsely predicted negative.

Pseudo Code of Proposed Algorithm

Following is the pseudo code of the proposed algorithm:

```
Step 1: Input the data with features and labels.
Step 2: Apply the neural network:
Ii    =∑    ( ) wijxj

Step 3: Training neural network:
 ( ) = 1/2 ∑( ( ) −    ( ( )))

Next weight:

=  −                  ( )

Step 4: Evaluation of Mean Squared Error:
        2
       1
=     ---∑( ( ) −  ( )−1)
       2

Step 5: Start genetic algorithm t=0

Initialize the random population
                                   ( );
Step 6: Increase counter t+1;
′ =                                ( )
Mutate (  ′)
```

Classifier	Precision	Recall	Accuracy
SVM-RBF	86.86	86.97	85.25
NN-GA	87	87.23	90.23

Step 7: Evaluate ′( )
Step 8: = ;
If weight is optimized, stop genetic algorithm else
Go to step6.

IV. RESULTS

Table 4.1Comparative Analysis for EU eProcurement data set:

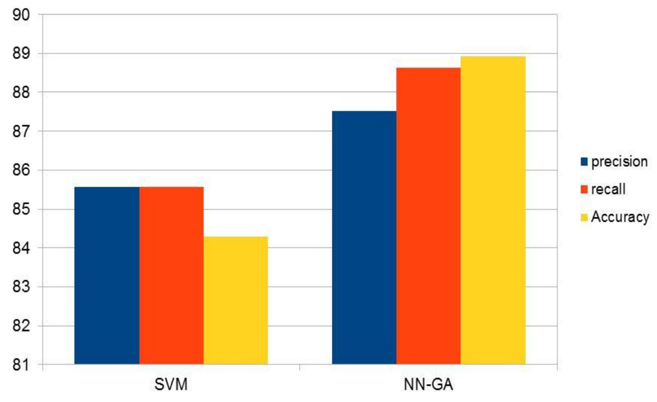
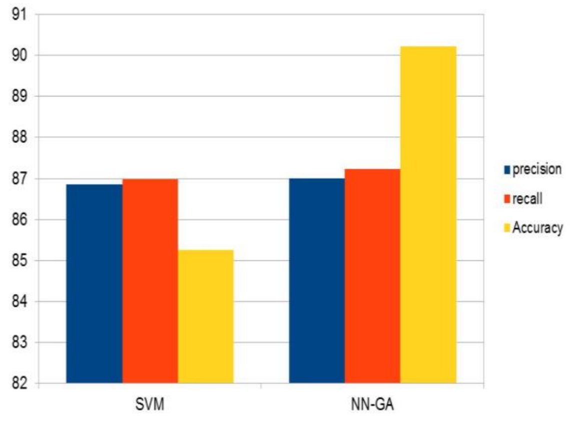


Table 4.2 Comparative Analysis for PROMISE data set



V. CONCLUSION

Agile development has emerged out as the most popular means of software development in recent times. Agile offers a variety of different methodologies each of which has its own set of pros and cons and may be selected by software companies based on their environment, project types, available resources and other constraints. Requirement engineering is a crucial part of this development technique. This paper proposes an automatic approach based on supervised learning for classification of functional and non-functional requirements in agile development. The proposed method does so by using the concept of clustering with the neuro-genetic approach. The requirements in requirement engineering step of the process are communicated through regular expressions through Software Requirement Specification (SRS).

Classifier	Precision	Recall	Accuracy
SVM-RBF	85.56	85.56	84.29
NN-GA	87.52	88.62	88.92

VI. REFERNCES

[1] H. K. Flora and S. V. Chande, "A Systematic Study on Agile Software Development Methodologies and Practices," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3626-3637, 2014.

[2] G. Kumar and P. K. Bhatia, "Impact of Agile Methodology on Software Development Process," *International Journal of Computer Technology and Electronics Engineering*, vol. 2, no. 4, pp. 46-50, 2012.

[3] S. Sharma, D. Sarkar and D. Gupta, "Agile processes and methodologies: A conceptual study," *International journal on computer science and Engineering*, vol. 4, no. 5, pp. 892-898, 2012.

[4] D. Cohen, M. Lindvall and P. Costa, "An introduction to agile methods," *Advances in computers*, vol. 62, pp. 1-66, 2004.

[5] M.-R. Hilkka, T. Tuure and R. Matti, "Is extreme programming just old wine in new bottles: A comparison of two cases," *Journal of Database Management*, vol. 4, no. 41-61, p. 16, 2005.

[6] "10th Annual State of Agile Survey," VersionOne Inc., 2016. [Online]. Available: https://www.versionone.com/about/press-releases/versionone-releases-10th-annual-state-of-agile-report/.

[7] B. Boehm, "Get Ready for Agile Methods, with Care," *Computer*, vol. 35, no. 1, pp. 64-69, 2002.