DOI: http://dx.doi.org/10.26483/ijarcs.v9i2.5660

ISSN No. 0976-5697

Volume 9, No. 2, March-April 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A FIREFLY APPROACH FOR PRIORITIZING FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS

Yesudoss. J
Assistant Professor, Department of Computer Science
Sri Ramakrishna Mission Vidyalaya College of Arts and Science
Coimbatore- 641 020, India

Ramani. A. V
Associate Professor and Head, Department of Computer ScienceSri Ramakrishna Mission Vidyalaya College of Arts and Science
Coimbatore- 641 020, India

Abstract: The prioritization of requirements is an important task in software development to implement requirements based on budget, time, customer expectations and practical constraints. An Interactive Genetic Algorithm (IGA) has been used to prioritize requirements by satisfying constraints. IGA sometimes provides a poor ranking due to population divergence. This is resolved in this paper by utilizing Firefly algorithm. The methods for prioritizing both functional and non functional requirements together are very less and have several drawbacks like of considering a small number of non functional requirements and applying less ranking weight. The firefly algorithm provides random prioritization initially, then improves the prioritization iteratively by minimizing disagreement between priorities and constraints while maximizing agreement between priority and customer satisfaction. The sum of disagreement is considered as light intensity of firefly. The population of firefly generates an ordered list of functional and nonfunctional requirements. The fitness value (intensity) is found for all fireflies then each firefly updates its position towards the best search of a firefly until all fireflies obtain same fitness value. The experimental result proves that the firefly based prioritization outperforms than Genetic Algorithm based prioritization.

Keywords: Requirements engineering; Firefly algorithm; Prioritization; Functional requirements; Nonfunctional requirements

I. INTRODUCTION

Software requirement prioritization is a process in which requirement engineers discover the most significant stakeholders' requirements so as to develop an efficient system, specifically an innovative system. The core requirements should be executed within the defined constraints of time, resources, cost and quality, to satisfy the customers [1]. Numerous methods were proposed to requirements prioritization [2]. Among the prioritization approaches utilized in these methods, Analytical Hierarchy Process (AHP) [3] exploits a pairwise comparison technique to extract the user's knowledge with respect to the ranking of the requirements. This paper is focused on firefly algorithm for functional and nonfunctional requirement prioritization as well as for the consistent requirement documents which needs to be prioritized for further software development processes [4].

An Interactive Genetic Algorithm (IGA) [5] was proposed to gather pairwise information helpful to prioritize the requirements for a software system. This algorithm was applied to a real case study, such as a nontrivial number of requirements that creates Analytical Hierarchy Process (AHP) hardly applicable. The proposed technique scaled to the size of the considered case study and created an outcome which outperforms IGA. Particularly, through eliciting among 50 or 100 pairwise comparisons from the user it is possible to achieve a substantially better ordering of the prioritized requirements. But in this approach, poor ranking was obtained because of population divergence.

In this paper, to overcome this problem, a new approach firefly algorithm is introduced for prioritizing both functional requirements (FR) and nonfunctional requirements (NFR). This algorithm is performed random prioritization and enhanced the prioritization iteratively based on disagreement between priorities and constraints while maximizing the agreement between priority and customer satisfaction. The sum of disagreement is considered as light intensity of firefly. The population of firefly generates an ordered list of FRs and NFRs. The light intensity (fitness) is discovered for the entire fireflies and every firefly update its position towards optimal firefly till the entire firefly obtain same fitness value.

II. RELATED WORK

Many algorithms have been proposed to prioritize requirements. A few of them are expressed here in this section.

Apriori algorithm [6] was utilized to discover the frequent requirements. This algorithm was followed through discovering the association among the frequent requirements. In databases, this algorithm was used to extract frequent item set. This algorithm was discovered the item sets by using minimum support value. This algorithm was executed two processes, i.e., prune and join actions.

An expert decision support system (i.e., PHandler) [7] was introduced to prioritize the large scale requirements. This approach was a combination of three methods such as the AHP, Value-based Intelligent Requirement Prioritization (VIRP) method and Neural Network. The AHP was used on prioritized groups of requirements for enhancing the scalability of the requirement prioritization procedure.

A prioritization method [8] was proposed with rework effort estimation, identification and examination of prioritized tasks to raise a final decision prior to software release. A text mining algorithm was used to assist the analysis of source code comments on open source projects and minimize the Self-admitted Technical Debt.

A requirements prioritization scheme named Case-Based Ranking [9] was introduced for requirements prioritization. This was figured out by the software planner's first choices with the requirements arrangements approximations by using one of the data mining techniques (machine learning). While preserving the precision of the final ranking was estimated, the human effort was decreased. Domain knowledge encoded as partial order relations were described over the requirement attributes was exploited, so supported an adaptive elicitation procedure.

A novel technique [10] was developed to prioritize optional requirements. In this technique, a prioritization evaluation attributes tree was constructed for creating the ranking criteria selection. Rankboost is used for computing the subjective requirements prioritization along with stakeholder preferences. An approach using the weighted PageRank was used to examine the dependencies among requirements. An integrated requirements prioritization approach was proposed to modify the stakeholder's subjective preferences with the objective requirements dependencies. However, this approach only supports contribution dependencies and business dependencies.

In agile approaches, a conceptual framework [11] was developed for requirement prioritization procedure. The objective of this framework was to discover the factors (environment, product and process) which influence the selection of requirements by prioritization procedure in agile development.

The requirements prioritization [12] was investigated to be potentially utilized based on thousands of users. A method for requirements prioritization was applied by using web questionnaires to the probable users for investigating the gain of the requirement condition. An additional part of the instability of the requirements and promotion contest were discussed. At last, in the Macbeth technique, ratio scale of computation was obtained from the Respondents questionnaire and a score of requirements.

In the end, a respondent sent in the questionnaire and a score of requirements in ratio scale of measurement was obtained, and the criteria have been consolidated through the Macbeth approach.

A new method [13] was addressed for estimating the degree of customization an ERP (Enterprise Resource Planning) requires to suit the executing organization. In this method, a framework was used for requirement prioritization and the algorithm was applied for customization requirements. This method was utilized the idea of requirements traceability to discover the gaps among the prioritized requirements of the customer and those embedded in the ERP software.

.

III. PROPOSED METHODOLOGY

In this section, a firefly algorithm [14] is used for minimizing disagreement between priorities $(Pr_1, Pr_2, \dots, Pr_n)$ and its requirements. This algorithm improves prioritizing requirement documents for both FRs and NFR constraints such as disagreement, distance and time consumption. The NFRs are given more importance by the user for prioritizing the required documents to develop softwares in a prioritizing order.

A. Minimize the disagreement of priority order with constraints using Firefly Algorithm

This algorithm is utilized for determining the minimum disagreement by using the objective function (OF). The OF for a given optimization problem is depending upon the light intensity. This function is used to move the fireflies towards brighter and more attractive locations to find the best solutions. Through the light intensity and the related to OF, the whole fireflies are illustrated. Each firefly has changed its position iteratively to attain the best solution.

The population of fireflies is initialized to begin the process [14]. There are two essential tips in firefly algorithm: formulation of the attractiveness and the variation in light intensity. The attractiveness is determined for finding the minimum disagreement based on the lower brightness of the fireflies. Also, the OF is used to determine the brightness (I) of the firefly in a specific position.

The OF is to minimize the disagreement (D) between priorities and constraints that are expressed iteratively by the user during the prioritization process. It is computed as follows,

$$dis\ (ord_1, ord_2) = \{(p,q) \in ord_1^* | (q,p) \in ord_2^*\} \quad (1)$$

In equation (1), the disagreement among two orders $order_1, order_2$ (fractional or whole), are defined from the same set of R elements. This is the set of pairs in the transitive closure nature of the first order, $order_1^*$ which appear reversed in the second order closure $order_2^*$. The measurement of disagreement is given using the size of the set $dis\ (ord_1, ord_2)$ [5].

The intensity of the fireflies is given as follows:

$$I(x) = Min(D) \quad (2)$$

The attractiveness function of the firefly is:

$$\beta(r) = \beta_0 . ew^{-\gamma . r^2} \quad (3)$$

From equation (3), γ indicates the media light absorption coefficient. β_0 denotes the attractiveness value of the firefly at r=0. The estimation of the distance among the two fireflies i and j at the positions x_i and x_j is defined.

$$r_{ij} = \left|\left|X_i - X_j\right|\right| = \sqrt{\sum_{k=1}^{d} (X_{i,k} - X_{j,k})^2} \quad (4)$$

From equation (4), X_i in the i^{th} firefly, $X_{i,k}$ denotes the k^{th}element of the spatial coordinates and d indicates the number of dimension. The movement of firefly (i) is characterized as follows:

$$x_i = x_i + \beta_0 * \exp(-\gamma r_{ij}^2) * (x_j - x_i) + \alpha * (rand - \frac{1}{2}) \quad (5)$$

In equation (5), the movement of the firefly is determined if the intensity value (fitness) of the firefly is high. From this equation, the first term represents the current position of the firefly i, the second term denotes the attractiveness of the firefly and third term indicates the movement of the firefly if there is no any brighter firefly, $\alpha \in (0,1)$ and $\beta_0 = 1$ in some of the cases. The light absorption coefficient varies from 0.1 to 10.

Algorithm of FAPFNR (Firefly Algorithm for Prioritizing Functional and Nonfunctional requirements)

Input:

FR: A set of functional requirements
NFR: A set of nonfunctional requirements
$ord_1, \dots, ord_k$: partial orders describing priorities and constraints upon R $(ord_i \subseteq R \times R \; defines \; a \; DAG)$, MAXItr

Output:
$< FR_1, \dots, FR_n; NFR_1, \dots, NFR_n >$: ordered list of functional and nonfunctional requirements

Steps:

1. Begin
2. thresholdDisaggreement = TH, Iteration = 1
3. Initialize firefly population with a set of ordered list of functional and nonfunctional requirements as initial solutions $\{Fr_i, \dots\} \; and \; \{NFr_i, \dots\}$
4. Evaluate light intensity I_i (sum of disagreement) for each firefly at $\{FPr_i, \dots\} \; and \; \{NFPr_i, \dots\}$ by using $f(PFr)$ and $f(PNFr)$
5. While(Iteration ≤ MAXItr)
6. While $(\min(I_i) \geq thresholdDisagreement)$ do
7. While ($I_i \geq I_j$)
8. Change the attractiveness of fireflies using equation (3)
9. Evaluate new solutions ie. light intensity (sum of disagreement)
10. End while
11. End while
12. Sort fireflies based on intensity; get best solutions ie. best prioritization of FR and NFR
13. End while
14. End

In the above algorithm, the pseudocode of the firefly algorithm is used to prioritize a set of FRs and NFRs and reduce the disagreement. The input of the algorithm is a set of one or more partial orders $(ord_1, \dots, ord_k)$ of functional and nonfunctional requirement documents. In step 1-12, parameters of the algorithm disagreement threshold and iteration are initialized. The population of fireflies is initialized with a set of totally ordered FRs and NFRs. Evaluate the light intensity (sum of disagreement) of each firefly for ordered FRs and NFRs. The attractiveness of firefly is changed using equation (3) then move towards best firefly. After movement, evaluate the light intensity (sum of disagreement) for each firefly. Finally the best solution is found from the updated solutions. The ordered list of FR and NFR for best firefly is selected as best prioritized requirements.

Iv. EXPERIMENTAL RESULTS

In this section, the overall performance of the FAPFNR (Firefly Algorithm for Prioritizing Functional and Nonfunctional requirements) is compared with IGA in terms of Disagreement, Average distance and Time consumption. The functional requirement documents for student information software like Transcript Requests, Centralized Location Records, Student Interface and Viewing Account Balance are prioritized. The nonfunctional requirements taken for the experiments are Reports Information, Usability Requirement, Web Accessibility, Intuitive Interface and Crash Handling.

A. Disagreement

A total order of prioritized requirements is obtained by using FAPFNR and IGA methods. A total order of prioritized requirements and initial order requirements are compared. Based on this comparison, incorrectly positioned requirements are calculated that is called disagreement. Average disagreement is calculated as follows,

$$Average \; Disagreement = \frac{Total \; sum \; of \; all \; the \; disagreement}{Total \; number \; of \; disagreement}$$

Table I. Disagreement

Prioritizations (Pr)	***Order of FAPFNR prioritized requirements***	***Order of IGA prioritized requirements***	***Sum of Disagreements FAPFNR***	***Sum of Disagreements IGA***
Pr_1	R_3, R_2, R_4, R_5, R_1	R_3, R_1, R_4, R_5, R_2	3	4
Pr_2	R_2, R_1, R_5, R_4, R_3	R_2, R_1, R_5, R_3, R_4	4	5
Pr_3	R_4, R_1, R_3, R_2, R_5	R_5, R_1, R_3, R_2, R_4	4	5
Pr_4	R_3, R_2, R_4, R_1, R_5	R_5, R_2, R_1, R_3, R_4	2	3
Pr_5	R_3, R_2, R_1, R_5, R_4	R_3, R_1, R_4, R_5, R_2	2	4

From table I, requirements order is given based on the FAPFNR method to calculate the disagreement value. Pr denotes the prioritizations list and R_1, R_2, R_3, R_4, R_5 are the requirements list of documents. Here, the total number of disagreements are 5. For example, an initial prioritization order requirements R_3, R_2, R_1, R_4, R_5 are denoted to check the disagreement value of FAPFNR and IGA methods for the table I. Disagreement values are measured by comparing the initial prioritization order requirements with order of FAPFNR prioritized requirements as listed in the table I. Similarly, disagreement values are measured by comparing the initial prioritization order requirements with order of IGA prioritized requirements as shown in the table I.

Table II. Average Disagreement

Methods	***IGA***	***FAPFNR***
Average Disagreement	62	46

From table II, in IGA method, 10 requirement documents (R1 to R10) are considered. Totally 10 numbers of disagreement prioritization lists and their disagreement values are obtained by using IGA method. The total disagreement value of IGA method is 620. Hence, the average disagreement is,

$$Average \; Disagreement \; for \; IGA = \frac{620}{10} = 62$$

Similarly, in FAPFNR method, the same 10 requirement documents are considered. Here also totally 10 numbers of disagreement prioritization lists and their disagreement values are obtained by using FAPFNR method. The total disagreement value of FAPFNR method is 460. Hence, the average disagreement is,

$$Average \; Disagreement \; for \; FAPFNR = \frac{460}{10} = 46$$

Figure 1 shows that the comparison of FAPFNR and IGA techniques in terms of Average disagreement. In this

graph, methods are represented in X-axis and the average distance is denoted in Y-axis. From this graph, proposed FAPFNR approach decreases 26% less than the existing IGA approach in terms of average disagreement.

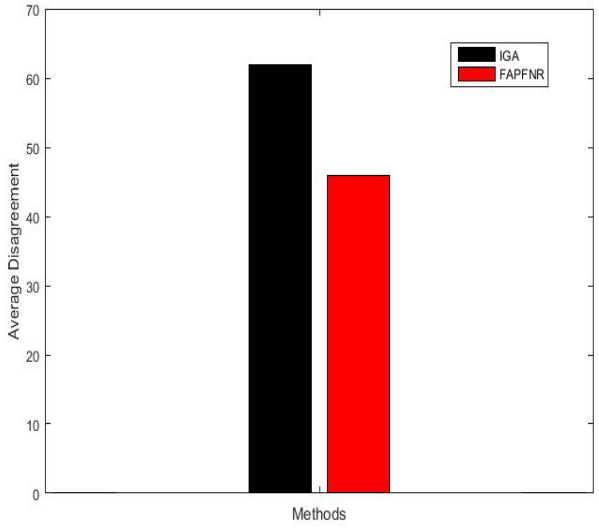

Figure 1. Comparison of Average disagreement

While comparing the above disagreement values of the two methods, naturally position of each firefly is decreased ie. each requirements document positions are identified in a shorter distance order. Hence, Figure 2 denotes an another view of the disagreement of IGA and FAPFNR for the prioritized list produced by these two techniques according to its position. Each requirement is placed (positioned) in some location by using IGA and FAPFNR approaches.

For example, in an order $R_4R_1R_3R_2R_5$, R_4 is denoted as position one and R_1 is represented as position two and so on. From the analysis, the position of requirement is represented in X-axis and the values of the disagreement are denoted in Y-axis. In FAPFNR approach, the figure 2 shows that the positional order of the requirements is also decreased related to their disagreement values than to the existing IGA approach.

Table III. Position Vs Disagreement

Position	***IGA disagreement (out of 10 requirements)***	***FAPFNR disagreement (out of 10 requirements)***
1	3	2
2	5	3
3	7	4
4	2	1
5	8	5
6	6	4
7	4	2
8	9	7
9	5	3
10	6	5

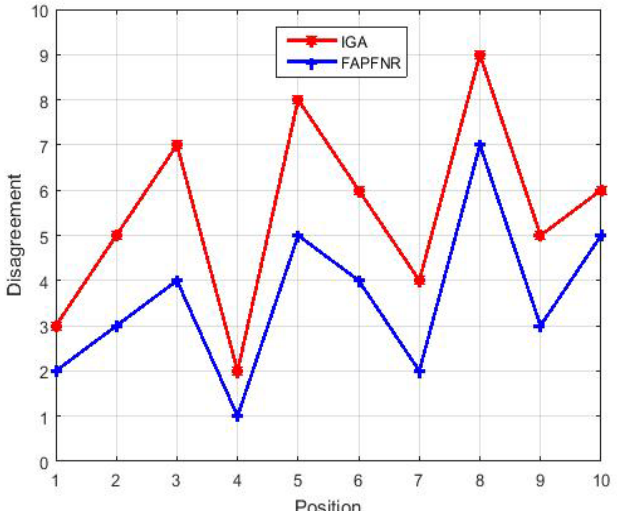

Figure 2. Position Vs Disagreement

B. Average Distance

It is defined as the distance between the position of each requirement in the prioritization order produced by using IGA and FAPFNR and the position of the same requirement in the initial order. The Average Distance is calculated as follows,

$$Average\ Distance = \frac{Total\ sum\ of\ all\ the\ distance}{Total\ number\ of\ distance}$$

For instance, a prioritization order of $Pr_3(R_4,R_1,R_3,R_2,R_5)$ from the table I with the initial prioritization order R_3,R_2,R_1,R_4,R_5 and the distance is measured by using position of each requirement. For instance, first position of Pr_3 is R_4 and the initial prioritization order is R_3. Therefore, the distance is 1. The sum of all distances in $Pr_3(1,1,2,2,0)$ is 6.

From table IV, in IGA method, 10 requirement documents (R1 to R10) are considered. Totally 4 numbers of distance lists are taken for IGA method and identified 32 total sum of all the distances. Hence, the average distance is,

$$Average\ Distance\ for\ IGA = \frac{32}{4} = 8$$

Similarly, in FAPFNR method, 10 requirement documents (R1 to R10) are considered. Totally 4 numbers of distance lists are taken for FAPFNR method and found 24 total sum of all the distances. Hence, the average distance is,

$$Average\ Distance\ for\ FAPFNR\ = \frac{24}{4} = 6$$

Table IV. Average Distance

Methods	***IGA***	***FAPFNR***
Average Distance	8	6

In Figure 3, comparison results of the proposed FAPFNR approach with existing IGA method in terms of average distance. From the bar graph, methods (IGA and FAPFNR) are represented in X-axis and average distance is indicated in Y-axis. In this analysis, it is reduced for proposed FAPFNR approach compared to the existing IGA method.

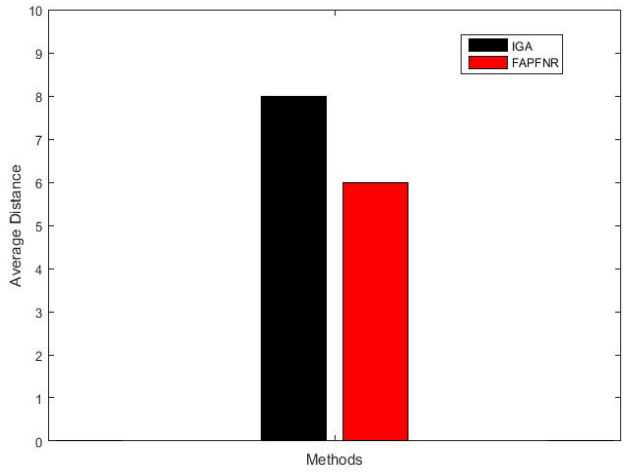

Figure 3. Comparison of Average distance

C. Time Consumption

Time consumption is measured as calculating the difference between the start and end time of implementing prioritization task based on IGA and FAPFNR approaches. The sequential search process of IGA method taken more time to execute. The proposed FAPFNR approach has reduced the time consumption due to the process of parallel searching. The comparison of existing IGA and proposed FAPFNR schemes for metric time consumption is illustrated in Figure 4. From this analysis, the horizontal axis represents the methods and the vertical axis denotes the time consumption values (ms). From the analysis, it is demonstrated that the proposed FAPFNR approach achieves very less than existing IGA method. From this analysis, proposed approach reduces 46% more than the existing approach in terms of time consumption.

Table V. Time Consumption

Methods	***IGA***	***FAPFNR***
Time Consumption (ms)	1300	700

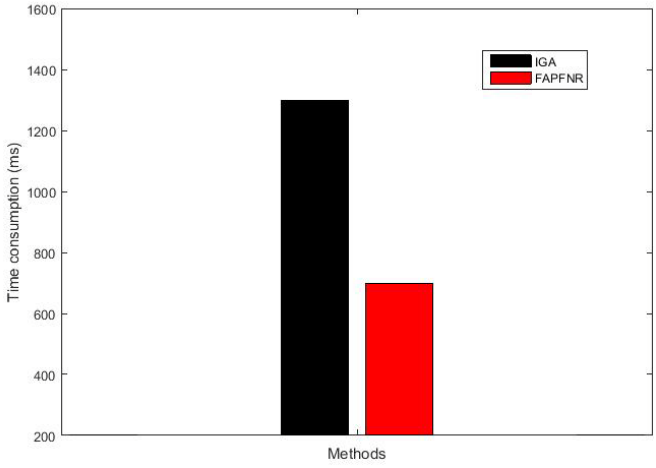

Figure 4. Comparison of Time Consumption

V. CONCLUSION

In this paper, a firefly algorithm is proposed for Functional and Nonfunctional Requirements prioritization. This algorithm is obtained the minimum disagreement with FRs and NFRs prioritization. Additionally, the proposed approach is used for reducing the time consumption in prioritizing the requirement documents. The experimental results show that the proposed approach provides better results in terms of Disagreement, Average distance and Time consumption. This approach may be additionally enhanced by comparing other FR and NFR constraints.

VI. REFERENCES

[1] Abran, A., Moore, J. W., Bourque, P., Dupuis, R., & Tripp, L. L. (2004). *Guide to the software engineering body of knowledge: 2004 version SWEBOK*. IEEE Computer Society.

[2] Moisiadis, F. (2002, October). The fundamentals of prioritising requirements. In *Proceedings of the systems engineering, test and evaluation conference (SETE'2002)*.

[3] Saaty, T. L., & Vargas, L. G. (2012). *Models, methods, concepts & applications of the analytic hierarchy process* (Vol. 175). Springer Science & Business Media.

[4] Yesudoss, J., Ramani, A.V., (2017). Inconsistency Checking in Requirements by Improved Semantic Reasoning (ICRISR). International Journal of Recent Scientific Research, Vol. 8, Issue, 8, pp. 19461-19466, August, 2017.

[5] Tonella, P., Susi, A., & Palma, F. (2010, September). Using interactive GA for requirements prioritization. In *Search Based Software Engineering (SSBSE), 2010 Second International Symposium on* (pp. 57-66). IEEE.

[6] Anand, R. V., & Dinakaran, M. (2017). Handling stakeholder conflict by agile requirement prioritization using Apriori technique. *Computers & Electrical Engineering*, *61*, 126-136.

[7] Babar, M. I., Ghazali, M., Jawawi, D. N., Shamsuddin, S. M., & Ibrahim, N. (2015). PHandler: an expert system for a scalable software requirements prioritization process.*Knowledge-Based Systems*, *84*, 179-202.

[8] Mensah, S., Keung, J., Svajlenko, J., Bennin, K. E., & Mi, Q. (2018). On the value of a prioritization scheme for resolving Self-admitted technical debt. *Journal of Systems and Software*, *135*, 37-54.

[9] Perini, A., Susi, A., & Avesani, P. (2013). A machine learning approach to software requirements prioritization. *IEEE Transactions on Software Engineering*, *39*(4), 445-461.

[10] Shao, F., Peng, R., Lai, H., & Wang, B. (2017). DRank: A semi-automated requirements prioritization method based on preferences and dependencies. *Journal of Systems and Software*, *126*, 141-156.

[11] AL-Ta'ani, R. H., & Razali, R. (2013). Prioritizing requirements in agile development: a conceptual framework. *Procedia Technology*, *11*, 733-739.

[12] Santos, R., Albuquerque, A., & Pinheiro, P. R. (2016). Towards the Applied Hybrid Model in Requirements Prioritization. *Procedia Computer Science*, *91*, 909-918.

[13] Parthasarathy, S., & Daneva, M. (2016). An approach to estimation of degree of customization for ERP projects using prioritized requirements. *Journal of systems and software*,*117*, 471-487.

[14] Kumar, R., & Sakthivel, S. (2016). Optimized Multi-Metric Method and Packet Hiding Method (OM3-PHM) for Detection and Prevention of Jamming Attacks. *Asian Journal of Research in Social Sciences and Humanities*, *6*(7), 2140-2151.