International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 8, pp 273-278, August-2024

https://doi.org/10.47001/IRJIET/2024.808032

Software Requirement Specifications Using Intelligent Technical: Literature Review

[1]Sama Emad Sheet, [2]Ibrahim Ahmed Saleh

[1]Student, Department of Software, College of Computer & Math., University of Mosul, Iraq
[2]Professor, Department of Software, College of Computer & Math., University of Mosul, Iraq

***Abstract* - Software requirement is become more important in recent because the development which witness in projects, badly executed requirements engineering steps can result in bad quality software and more cost for expensive maintenance. Manual classification of requirements is difficult, time-consuming, and expensive, especially in large projects and is written as a Software Requirements Specification (SRS) document. For this reason, automating software requirements classification helps in obtaining higher accuracy and saving time and effort. Most of researcher applied Intelligence techniques algorithms to avoid erroneous requirements and human intervention, as well as analyze, classify, and priority of requirements. In this paper illustrated modern of artificial techniques algorithm to classify RT approaches. It is surveyed that existing techniques like machine learning algorithms such as K-Nearest Neighbor (K-NN), decision tree (DT),.. etc. Many other technical how ensemble learning and deep learning algorithm results in classification of RF. Researchers have proposed automated techniques to classify functional and non-functional requirements using several machine learning (ML) algorithms with a combination of different vector techniques. However, using the best method in classifying functional and non-functional requirements still needs clarification, and through many studies and research by researchers.**

Keywords: Requirements engineering, functional requirements, non-functional requirements, machine learning, classification, intelligent requirements engineering.

I. INTRODUCTION

Classification of software requirements is critical phase in the software development life cycle. It organizes project's requirements into different categories to facilitate the management process and enable designers to prioritize and track them easier. The main types of software requirements are functional, non-functional, and domain requirements [1]. FRs, which relate to the behavior of functions that the system implements. NFR, which describe features (such as Quality, security, ease of use, privacy, etc.),In addition to constraints used in the application, while domain requirements are specific to domain or industry in which software operates [2].

FR represents the basic features or characteristics that a software system must possess in order to achieve its goal. In simpler terms, these requirements specify what the system must do. They describe the interactions between the software and its users, as well as the behavior of the software under various conditions. Functional requirements typically have the following characteristics:

- Specificity: They are detailed and specific, leaving little room for ambiguity. They describe the precise functions, inputs and outputs of the system.
- Verifiable: Functional requirements are testable and can be validated to ensure that the software works as intended.
- User-centric: They are closely aligned with the user's needs and expectations, ensuring that the software fulfills its purpose.
- Changeable: Functional requirements can change over the course of a project as user feedback and business needs evolve [3].

NFR defines the quality attributes of the system, including performance, security, availability, look and feel, fault tolerance, legal and operational, essential for meeting user needs and imposing additional constraints on software quality. Prioritizing NFR from user requirements is challenging, requiring specialized skills and domain knowledge [4]. NFRs go hand-in-hand with the Functional Requirements (FRs) and are highly essential to ensure the development of an efficient and a reliable software system that meets the customers' needs and fulfills their expectations. Set of NFRs need to be correctly identified in the initial phases of Software Development Lifecycle (SDLC) process as they play a crucial role in the architecture and design of the system which in turn affects the quality of the system [5].

Identifying both functional requirements and NFRs is individually crucial for ensuring an effective software development process. But, both types of requirements are mixed together within the same documents, which is challenging to separate manually [6, 7]. However, neglecting

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 8, pp 273-278, August-2024

https://doi.org/10.47001/IRJIET/2024.808032

NFRs extraction can lead to project failure or increase production costs [8].

The automatic supervised ML algorithm was applied to classify requirements as functional and non-functional using syntactic and keyword features in [9]. Machine learning (ML) techniques take a lot of human effort to carefully design features from raw data. ML methods are capable to deal with raw data through many levels of representation where each level transform the previous data, the most important thing about ML is that each representation of the data is not made and modeled by software engineers, actually are built automatically by the models [10-11].

The fit criteria for NFR identification and ML training lack clarity [12]. Mostly, syntactic feature part-of-speech (POS) tagging is used [13, 14–15]. Abad found that pre-processing improved the performance of an existing classification method [12].

The second most common technique, Bag of Word (BOW) [14, 16, 17, 18], fails to maintain the sentence order and cannot deal with polysemy [19]. In comparison, n-grams feature can consider the word order in a close contest to its neighboring words [17, 20], but it also suffers from data scarcity [21].

The reset of paper in section 2 software requirements, section 3 related works, finally section 3 discuss conclusion.

II. RELATED WORK

There are many researchers interesting for their articles about classify identified requirements into functional and non-functional requirements. Several studies have been performed to investigate how well the AI and ML approaches apply in this context.

In 2013 Ramadhani *et al.* [22] Proposed an automated system to identify non-functional requirements from sentence-based classification algorithms required for FSKNN with addition of semantic factors such as the development term by hyper name and measuring the semantic association between the term and each quality aspect class. Based on ISO/IEC 9126. The result of their research is Semantic-FSKNN method can reduce the multiplication loss or error rate by 21.9%, and also increase the accuracy value by 43.7%, and the accuracy value also 73.9% compared to the FSKNN method without adding semantic factors to it. The researchers proved if adding semantic factors to the FSKNN method will improve performance of hamming loss in proportion mentioned above.

In 2017 Kurtanović *et al.* [23] used over- and over-sampling strategies to deal with imbalanced classes in a dataset and validated classifiers using precision, recall, and F1 metrics in a series of experiments based on the Support Vector Machine classifier algorithm. It achieves an accuracy and recall of up to approximately 92% for automatically identifying FRs and NFRs. To identify specific NFRs, it achieved the highest levels of precision and recall for security and performance NFRs with up to 92% precision and approximately 90% recall.

Abad, S. *et al.* 2017 [24] presented how to improve the automatic classification of requirements into FR and NFR using Latent Dirichlet allocation (LDA), Biodegradable Temporising Matrix (BTM), Hierarchical, K-means, Hybrid, and Binarized Na¨ıve Bayes (BNB) machine learning algorithms. BNB performed the highest in subclassifying NFRs. Although BTM performs better than LDA in general, it does not perform well for subclassifying NFRs. This method standardizes and normalizes requirements before applying classification algorithms. The study was conducted on 625 requirements from the OpenScience tera-PROMISE repository. It was found that preprocessing improves the performance of both FR/NFR classification and NFR subclassification.

In 2020 S Tiun *et al.* [25] introduce software classification used Word2vec and fast Text methods are perform text analytics to gain intuition or knowledge from the crowd's feedback. They concluded that "fast Text" was best model for FR and NFR classification. The superior result given by fast Text compared to Word2vec with deep learning classifier concludes using deep learning classifier does not necessarily outperform linear classifier in text classification problem. For binary text classification with very short document length and minimal vocabulary, fast Text can do a better job. They suggested that if one prefers to use traditional features and classifiers in classification similar to FR and NFR, one should consider using TFIDF with NB as their model as the model had the highest F1 score of 92.8%.

In the same year Canedo *et al.* [26] showed that combining two text routing techniques with four machine learning algorithms to manually classify user requirements into two types "functional requirements and nonfunctional requirements". The NFR has eleven types (subcategories of non-functional requirements), and twelve types of FR plus subcategories of non-functional requirements. The researcher found that combination of TF-IDF and LR had the best performance metrics for binary classification, NFRs classification, and requirements classifications overall, with an F-measure of 91% on binary classification, 74% on 11-detail classification, and 78% on 12-detail classification.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 8, pp 273-278, August-2024

https://doi.org/10.47001/IRJIET/2024.808032

In 2021 Abdel Majeed *et al.* [27] present in their study a method for classifying automatic software requirements. Natural language is full of ambiguity, is not well defined, and has no regular structure, it is considered to be somewhat variable. The system development requirements are classified into functional and non-functional requirements using two machine learning methods .They applied two machine learning models: linear regression (LR) and multinomial model (MNB).

The paper showed that MNB outperforms LR in the requirements classification task taking into account accuracy, sensitivity, and measurement accuracy. It also proved that the MNB model achieved the highest accuracy of 95.55%, sensitivity of 93.09%, and accuracy of 96.48%. However, when using LR, the proposed model has a classification accuracy of 91.23%, a sensitivity of 91.54%, and an accuracy of 94.32%. In Figure (1) the outline of the work shows that optimizing the preprocessing task in research has a significant impact on the result of automatic classification. For requirements.

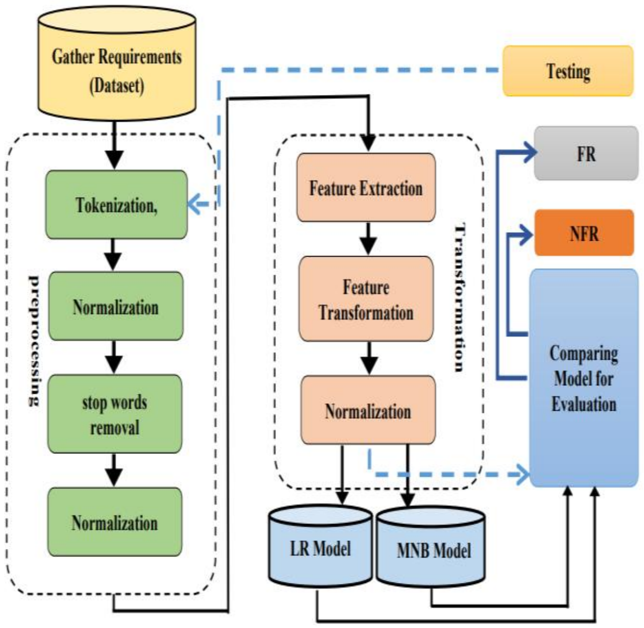

Figure (1) explains the block diagram including the suggested work's approach [27]

In the same year Althunibat *et al.,* [28] proposed a technique to automatically classify software requirements using machine learning. Text vectorization (BoW) technique was used with two ML algorithms (SVM and KNN to classify requirements into two categories (NFRs and FRs).The result of paper's performance metrics (recall, precision, and F1 score (F-measure) of the classification output). It was found that using BoW with SVM is better than using KNN algorithms with an average F-measure for all cases of 0.74.

The F1 score (F-measure) is 90% on binary classification (FR or NFR), 66% in (11) classes. Subcategories on NFR type (availability, performance, security, etc.), rating and 72% on rating of 12 subcategories.

In 2022 Khurshid *et al.* [29] proposed a new machine learning algorithm based on KNN rules to automatically classify NFR with better accuracy. They adopted many ML algorithms for classification are LR, SVM, MNB, KNN, Ensemble, Random Forest (RF), and hybrid rule-based KNN algorithms. The researchers depended BoW and TF IDF for feature extraction while ML algorithms for classification, an average accuracy of 85.7% can be achieved, which they believe was excellent performance

Traditional machine learning techniques for NFR extraction often rely on a large number of pre-classified requirements, which can be limiting. To overcome this limitation, In 2023 Amin Khan *et al.* [30] proposed a transfer learning strategy to identify and classify NFRs in software development using a Word2Vec model trained on the NFR dataset. Our approach took advantage of the semantic similarity between functional requirements (FRs) and NFRs, resulting in the identification of an effective NFR. While evaluating several transfer learning models, including BERT, Distil Bert, Distil Roberta, Electra-base, and Electra-small, it is found that the proposed XLNet model outperforms other models. It achieved exceptional precision, precision, recall, and F1-score, including an impressive Matthews Correlation Coefficient (MCC) of 0.906670. These results clearly demonstrate the effectiveness of XLNet in solving the challenges of NFR identification and classification, leading to accurate and consistent results.

In the same year Abdur Rahman *et al.,* [31] combined four text routing methods with fifteen machine learning classifiers to classify requirements into eleven types of non-functional requirements. Among the conversion methods, Chi Squared performed the worst, followed by Hashing, BoW, and TF-IDF. TF-IDF shows the best performance for requirements classification and the reported accuracy, precision, and recall are 0.667, 0.667, and 0.539, respectively, where LSVC with TF-IDF recorded the highest accuracy score of 81.5%.

The paper was concluded that LSVC algorithm and TF-IDF routing technique is the best combination for classification of non-functional requirements.

ISSN (online): 2581-3048
Volume 8, Issue 8, pp 273-278, August-2024
https://doi.org/10.47001/IRJIET/2024.808032

Table 1: Summary of techniques used for classification by the researchers mentioned previously

	Authors	Technique used	Accuracy
1.	Ramadhani *et al.* [22]	FSKNN, Semantic-FSKNN	Semantic-FSKNN error rate=21.9%, accuracy=43.7%, precision=73.9%
			FSKNN performance of hamming loss=21.9%, accuracy =43.7% precision=73.9%
2.	Kurtanović *et al.* [23]	the Support Vector Machine	accuracy and recall = 92% precision=92% recall=90%
3.	Abad, S. *et al.* [24]	Latent Dirichlet Allocation(LDA), Biterm topic Modeling(BTM), Binarized Na¨ıve Bayes (BNB)	BNB correctly =94.40% average precision =0.95 recall = 0.94
4.	S Tiun *et al*. [25]	TFIDF& NB FastText	FastText F1 score=92.8%
			TFIDF& NB F1-score=91.13
5.	Canedo *et al.* in [26]	TF-IDF and LR	(TF-IDF & LR)F-measure binary classification=91%, 11-detail classification= 74%, 12-detail classification=78%
6.	Abdulmajeed *et al.*[27]	MNB LR	MNB (accuracy=95.55%, sensitivity=93.09%, accuracy=96.48%)
			LR (accuracy=91.23%, sensitivity=91.54%, accuracy=94.32%)
7.	Althunibat *et al.* [28]	(BoW&SVM) (BoW&KNN)	(BoW& KNN) F1 score (F-measure)= 90% binary classification =66% classification on the 12- subcategories =72%
8.	Khurshid *et al.* [29]	Logistic Regression (LR), Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), K-Nearest Neighbors (KNN), Ensemble,Random Forest (RF)	BoW & TF IDF, KNN rule-based hybrid classification, an av. accuracy = 85.7%
9.	Amin Khan *et al.* [30]	LR, SVM & SGD, Semantic, Similarity Distance, CNN, Norbert, XLNet	XLNet (accuracy=0.91 precision=0.91 F1-score scores=0.91)
			Norbert (accuracy=0.81 precision=0.91 F1-score scores=0.81)
10.	Abdur Rahman *et al.* [31]	Chi Squared, Hashing, BoW, TF-IDF,LSVC	accuracy=0.667, precision=0.667, recall=0.539, (LSVC&TF-IDF)accuracy score =81.5%

ISSN (online): 2581-3048
Volume 8, Issue 8, pp 273-278, August-2024
https://doi.org/10.47001/IRJIET/2024.808032

III. CONCLUSION

This study gives a good starting point for working with taxonomy to support researchers conducting software engineering surveys. The research focused on classifying functional and non-functional requirements using machine learning techniques and what are the best methods for this. They were used in classification to obtain the best accuracy, performance and quality with the least effort and time.

Through many researches, we found that MNB gives the best results for classifying requirements, taking into account accuracy of 95.55%, sensitivity of 93.09%, and measurement accuracy of 96.48%.

REFERENCES

[1] Rahimi, N., Eassa, F. and Elrefaei, L., 2020. An Ensemble Machine Learning Technique for Functional Requirement Classification. Symmetry, 12(10), p.1601.

[2] Dias Canedo, E. and Cordeiro Mendes, B., 2020. Software Requirements Classification Using Machine Learning Algorithms. Entropy, 22(9), p.1057.

[3] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, Non-Functional Requirements in Software Engineering. Kluwer Academic, 2000.

[4] Awad, E. Khamayseh, F. Arman, N. "Semi-Automated Classification of Non-Functional Arabic User Requirements using NLP Tools" Palestine Polytechnic University, 2024.

[5] S. Amasaki and P. Leelaprute, "The effects of vectorization methods on non-functional requirements classification," in 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Prague, Czech Republic: IEEE, 2018, pp. 175–182.

[6] D. Méndez-Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetro, T. Conte, M. Christiansson, D. Greer, C. Lassenius, T. Mannisto, M. Nayebi, M. Oivo, B. Penzenstadler, D. Pfahl, R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. Spinola, A. Tuzcu, J. de la Vara, and R.Wieringa, "Naming the pain in requirements engineering," Empirical software engineering, Vol. 22, No. 5, 2017, pp. 2298–2338.

[7] R. Svensson, M. Host, and B. Regnell, "Managing quality requirements: A systematic review," in 36th EUROMICRO Conference on Software Engineering and Advanced Applications. Lille, France: IEEE, 2010, pp. 261–268.

[8] Navarro-Almanza, R., Juarez-Ramirez, R. and Licea, G., 2017, October. Towards supporting software 2021 International Conference on Information Technology (ICIT) 689 engineering using deep learning: A case of software requirements classification. In 2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT) (pp. 116-120). IEEE.

[9] Hussain I, Kosseim L, Ormandjieva O. Using linguistic knowledge to classify non-functional requirements in SRS documents. In International Conference on Application of Natural Language to Information Systems 2008 Jun 24 (pp. 287-298). Springer, Berlin, Heidelberg.

[10] Osisanwo, F.Y., Akinsola, J.E.T., Awodele, O., Hinmikaiye, J.O., Olakanmi, O. and Akinjobi, J., 2017. Supervised machine learning algorithms: classification and comparison. International Journal of Computer Trends and Technology (IJCTT), 48(3), pp.128-138.

[11] Dash, R.Paramguru, R.L. and Dash, R., 2011. Comparative analysis of supervised and unsupervised discretization techniques. International Journal of Advances in Science and Technology, 2(3), pp.29-37.

[12] Abad, Z.S.H., Karras, O., Ghazi, P., Glinz, M., Ruhe, G., Schneider, K. What Works Better? A Study of Classifying Requirements. In: 2017 IEEE 25th International Requirements Engineering Conference (RE). pp. 496–501. IEEE, Lisbon, Portugal (2017).

[13] Robinson, W.N.: Two Rule-Based Natural Language Strategies for Requirements Discovery and Classification in Open Source Software ... Two Rule-Based Natural Language Strategies for Requirements Discovery and Classification in Open Source Software Development Projects. (2012).

[14] Sunner, D., Bajaj, H.: Classification of Functional and Non-functional Requirements in Agile by Cluster Neuro-Genetic Approach. International Journal of Software Engineering and Its Applications. 10, 129–138 (2016).

[15] Hussain, I., Kosseim, L., Ormandjieva, O. Using Linguistic Knowledge to Classify Non-functional Requirements in SRS documents. In: Kapetanios, E., Sugumaran, V., and Spiliopoulou, M. (eds.) Natural Language and Information Systems. pp. 287–298. Springer Berlin Heidelberg, Berlin, Heidelberg (2008).

[16] Casamayor, A., Godoy, D., Campo, M.: Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. Information and Software Technology. 52, 436–445 (2010).

[17] Mahmoud, M.: Software Requirements Classification using Natural Language Processing and SVD. International Journal of Computer Applications. 164, 7–12 (2017).

ISSN (online): 2581-3048
Volume 8, Issue 8, pp 273-278, August-2024
https://doi.org/10.47001/IRJIET/2024.808032

[18] Cleland-Huang, J., Settimi, R., Zou, X., Solc, P. Automated classification of non-functional requirements. Requirements Engineering. 12, 103–120 (2007).

[19] Tsai, C.-F. Bag-of-Words Representation in Image Annotation: A Review. ISRN Artificial Intelligence. 2012, 1–19 (2012).

[20] Zhang, W., Yang, Y., Wang, Q., Shu, F.: An empirical study on classification of non-functional requirements. of the 23rd International Conference on, (2011).

[21] Chong, T.Y., Banchs, R.E., Chng, E.S., Li, H. Modeling of term-distance and term-occurrence information for improving n-gram language model performance.

[22] Ramadhani, D. Rochimah, S. Yuhana, U. "Classification of Non-Functional Requirements Using Semantic-FSKNN Based ISO/IEC 9126" Dian Nuswantoro University Imam Bonjol, Semarang 50131, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia, 2013.

[23] Kurtanović,Z. Maalej, W. "Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning" Lisbon, Portugal, 2017.

[24] Abad, S. Karras, O. Ghazi, P. Glinz, M. Ruhe, G. Schneider, K. "What Works Better? A Study of Classifying Requirements" University of Calgary, Calgary, Canada, Leibniz Universitat Hannover, Hannover, Germany, University of Zurich, Zurich, Switzerland, arXiv:1707.02358v1 [cs.SE] 7 Jul 2017..

[25] S Tiun, U A Mokhtar, S H Bakar, S Saad, "Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text" Universiti Kebangsaaan Malaysia, 43600 Bangi, Selangor, Malaysia, 2020.

[26] Canedo, E. Mendes, B. "Software Requirements Classification Using Machine Learning Algorithms" University of Brasília (UnB), P.O. Box 4466, Brasília 70910-900, Brazil, 2020.

[27] Abdulmajeed, A. Younis, S. "Supporting Classification of Software Requirements system Using Intelligent Technologies Algorithms" College of Computer Sciences and Mathematics, University of Mosul, 2021.

[28] Yquba, G. AL Qaisi, H. Althunibat, A. AlZu'bi, S. "Software Requirements Classification using Machine Learning algorithm's" Al-Zaytooneh University of Jordan, Amman, Jordan, 2021.

[29] Khurshid, I. Imtiaz, S. Boulila, W. Khan, Z. Abbasi, A. Javed, A. Jalil, Z. "Classification of Non-Functional Requirements From IoT Oriented Healthcare Requirement Document" International Islamic University, Prince Sultan University, Riyadh, Saudi Arabia, Air University, Islamabad, Pakistan, 2022.

[30] AMIN KHAN, M. SOHAIL KHAN, M. KHAN, A. AHMAD, A. HUDA, S. "Non Functional Requirements Identification and Classification Using Transfer Learning Model" College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia, 2023.

[31] Rahman, A. Nayem, A. Siddik, S. "Non-Functional Requirements Classification Using Machine Learning Algorithms" Centre for Advanced Research in Sciences (CARS), University of Dhaka, Bangladesh, 2023.

Citation of this Article:

Sama Emad Sheet, & Ibrahim Ahmed Saleh. (2024). Software Requirement Specifications Using Intelligent Technical: Literature Review. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 8(8), 273-278. Article DOI https://doi.org/10.47001/IRJIET/2024.808032

****