2021 International Conference on Information Technology (ICIT)

Software Requirements Classification using Machine Learning algorithm's

Gaith Y Quba, , Hadeel AL Qaisi, Ahmad Althunibat, Shadi AlZu'bi

Faculty of Science and IT, Al-Zaytooneh University of Jordan, Amman, Jordan

exam_pha@hotmail.com, hadeelalqaisi22@gmail.com, a.thunibat@zuj.edu.jo, smalzubi@zuj.edu.jo

2021 International Conference on Information Technology (ICIT) | 978-1-6654-2870-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIT52682.2021.9491688

***Abstract*— The world is growing and developing rapidly, and the demand for software has been increasing speedily, any software has many steps for building a program and all the steps are important for software requirements. Requirements classification can be applied manually, which requires great effort, time, cost and the accuracy may vary. Therefore, many previous researchv has been proposed to automate the classification process, but the automation process of the classification was not sufficient. In this study, we will propose a technique to automatically classify software requirements using machine learning to represent text data from software requirements specification and classify requirement to group Functional Requirement and Non-Functional Requirement. The experimented dataset in this study was the PROMISE_exp, which includes labeled requirements. All the documents of software from the database were changed (cleaned) with a set of steps (normalization, extractions, selection any techniques that will be used. The BoW used SVM algorithm or KNN algorithm for classification. This study used data from the PROMISE_exp to do the work, the information of the steps used to re-performed the classification, and the Measurement BoW, when using SVM and KNN algorithms the classification of requirements making can serve as a way and resources for another study. It can be seen that the use of BoW with SVM is better than use KNN algorithms with an average F-measure of all cases of 0.74. In future work we intend to improve to technique with make merge and change some algorithms as Logiest Regression to improve the Accuracy (Precision) of our model.**

***Keyword*: Rrequirement's engineering, machine learning, classification, KNN, SVM, Functional Requirement and Non-Functional Requirement, smart software engineering**

I. INTRODUCTION

A software life cycle gets up with different levels that starting in planning and finishing to testing and deploying software. All these steps are executed in different ways, as we need. Each way is called a Software Development Lifecycle (SDLC), Software requirements are one of the important steps that affect the building and Software Development (SD) and it plays a major role in the failure or success of software.

Classify the Requirements of any software into two main parts: Functional Requirement (FR) and Non-Functional Requirement (NFR) [1]. (FRs), which relate to the behavior of functions that the system implements, while NFR, which describe features (such as Quality, security, ease of use, privacy, etc.), In addition to constraints used in the application [2].

Manual classification of requirements is difficult and time-consuming, and costly especially on a mega project and written as a software requirements specification (SRS) document, for that reason, Automate the classification software requirements helps save time and effort, and accuracy is higher [4].

SRS document is most often a text, its need for classification is the try to arranging text documents into categories based on attributes and properties belonging to each text. Text classification is exploited in several areas, for example, news categorization and spam identification [2].

The stakeholders (user, customers, etc.) need to get effective requirements engineering, it's contains eliciting; analyzing, and studying the stakeholders' requirements, recording the requirements as requirements specification; validating that the recognized requirements match the stakeholders' requirements, and working the requirements evolution.

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 (such as pixel from an image, excel as CVS, data as text) 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 [6].

ML algorithms are organized into categories based on the desired result of the algorithm, it's divided into three types: Supervised Machine Learning, Unsupervised Machine Learning, and Reinforcement learning [7].

The above researches show that the Classification of the software requirements using (Bag of Words (BoW) with machine learning algorithms (KNN and SVM) and answer of question Which ML Algorithm (SVM and KNN) provides the better performance for the requirements classification task?

The rest of this paper is structured as follows: section

II introduces a literature review of related work, Section III describes Smart solutions for classifying requirements; The Result and Discussion are shown in section IV; Finally, Section V presents our conclusions and future works. doctors' use of accurate diagnosis, and patient ease of use of the health record.

II. Related Literature

This section overview and summarizes the previous research on Software Development Life Cycle, Support Vector Machine (SVM), classifying FR and NFR using traditional manual techniques or automated techniques and the works that utilized ML techniques to classify FR to various categories:

A. Software Development Life Cycle

Sabale et al provided in [9] the meaning of system lifecycle models into existence that emphasized the must need to follow part of structured approach towards improved system building. waterfall models were suggested like, prototype, rapid application development, V-shaped, etc. In the focus on the comparative analysis of these Software Development Life Cycle Models. With the growing operations of organizations, the need to automate the various activities increased. to make the system very easy they must some standard and structural procedures are introduced in the industry so that the change from manual to automated.

Mishra et al showed in [10] that SDLC used various models widely for developing software. SDLC models put pieces of information regarding the development of the software. SDLC models are important for improving developing the software in a manner organization such that it will be delivered on time and proper quality. Employing proper SDLC allows the project managers to regulate the whole development strategy of the software. Each SDLC has its advantages and disadvantages according to which they decide which model should be implemented under which conditions. For this, they need to compare SDLC models, they will compare different famous life cycle models.

Verma, S. in [8] showed that Software Engineering (SE) is one of the application systematic and quantifiable approaches to the development, working it, and upgrade or maintenance of software. Various steps or processes and methodologies have been developed over the last decades to improve software or system, with varying degrees of success.

B. Support Vector Machine (SVM) algorithm

Do et al presented in [11] a new parallel and incremental (SVM) algorithm for the classification of very big datasets on image processing type. SVM has displayed to build Precision models but the learning task usually needs a quadratic program so that this task for big datasets requires large memory capacity and a long time. they extend a recent Least Squares SVM (LS-SVM) proposed by Suykens and Vandewalle for building incremental and parallel algorithms. The new algorithm uses image processors to take high performance at less cost. Numerical test output on UCI and Delve dataset repositories displayed that our parallel incremental algorithm using images processing units is about 70 times faster than a CPU implementation and often significantly faster (over 1000 times) than state-of-the-art algorithms like LibSVM, SVM-perf, and CB-SVM.

C. Classifying Software Requirements Manually

Ko et al showed in [13] software engineers must the right understand stakeholder requirements to capably develop big-measure and intricate software. The requirements in huge projects are gathering from users (stakeholders), and they are written in normal human language (NLP). The first data requirements should be organized into categories to mechanically organized requirements into each labeled category using only subject words as the representative of the political analyst views. The suggested method is verified through experiments using two requirements’ data sets with two languages (Korean and English). As an output, the suggested method can provide an effective function for an Internet-based requirements analysis-supporting system to efficiently gather and analyze requirements from different and distributed end user's by using the network (internet).

D. Automatic Software Requirements Classification

Wang et al showed in [12] perfect academic properties and good output on Support Vector Machine (SVM) algorithms attract more care to its. the first step is analyzing the properties of Support Vector, then show a new learning process, which covers the SVM classification algorithm to the growing learning part. The academic basis of this algorithm is the classification equality of the (SV, training) set. the data is accumulated in the process of incremental learning. also, unmastered samples are discarded optimally by the LRU scheme. The academic study and trial outputs display that this algorithm could not only accelerate the training process, nonetheless also decrease the storage price, while the organization precision is also guaranteed.

Almanza et al suggest in [4] to use of Artificial intelligence (AI) by Deep Learning (DL) to classify software requirements. The model that suggests is based on Convolutional Neural Network (CNN) that has been state of art in other natural language-related tasks. appraisal our suggested model, dataset PROMISE corpus was taken, and it having a set of groups (labeled) requirements in FR and NFR. They achieve promising results on SRC using CNN even without handcrafted form any human features.

M Lu et al provided in [5], user shows were mechanically classified into some types of NFRs (usability, portability, performance, and reliability), (FRs), and others. They join more than two organization techniques for example BoW, CHI2, TF-IDF, and AUR-BoW (planned in this work) with dissimilar machine learning algorithms as J48, Naive Bayes, and Bagging to categorize user analyses. They show trials to comparison the F-measures of the organization outputs through all the combinations of the techniques and algorithms. output in the study: they result from shows that evaluated user reviews can go to best classification output, and the ML algorithm Bagging is more appropriate for NFRs classification from user reviews.

Dias et al comparison in [7, 14, 15] between the text feature extraction techniques, and ML algorithms to the problem of requirements engineer classification to reply to the two major questions “Which works better (Bag of Words (BoW) or Chi-Squared (CHI2) or Term Frequency–Inverse Document Frequency (TF-IDF)) for organizing Requirements into (NFR and FR), and the sub-categories of NFR and “Which Machine Learning Algorithm give the better performance for the requirements classification?”. The outputs showed that the use of Term Frequency–Inverse Document Frequency with by the use of e Logiest Regression had the best organization output to Multiple requirements, with an F-measure more than 0.90 in two classifications (binary).

Rahimi et al introduced in [1,17,18] a new ML technique for classifying FR sentences to improve their accuracy and quality. This technique collection between ML models. The technique was executed, trained, and checkup (tested) using an aggregated dataset. The output is the accuracy of classifying FR and the required time of about 0.7 s.

Zaid investigated in [3,19,20] is the existing automatic classification techniques to help researchers and software engineers to choose the appropriate requirement classification technique, automatic classification of SW requirements is an important task; although, it has little work in this field. This SLR epitomizes all techniques found in this subject. The resulted of it paper is needed more effort and research papers to enhance and increase the performance as well as the quality of proposed tools

In this work, we will compare the performance of (Bag of Words (BoW) technique with two machine learning algorithms, which are SVM. and kNN to classify the FRs and NFRs.

III. SMART SOLUTIONS FOR CLASSIFYING REQUIREMENTS

A. Support Vector Machine (SVM)

SVM is a strong and different ML model, capable of performing linear or non-linear classification, regression, and even outlier detection. The algorithm makes the organization making a linear of largest margin that splits two categories. This margin makes there are few options for separating the data from the sample, and thus, there is little chance of misclassifying a new instance. [2].

B. k-nearest neighbors (KNN):

K-Nearest Neighbor (k-NN) the main idea in this algorithm is dependent on the instances within the old dataset contains records in close to second instances that have equal (same) properties. This algorithm categorizes data that will be inserted by calculating the distance of this data with the previously standing instances inside the database, then it takes the closest k instances and compute their middling (average), for regression problems, or gets the mod. [2, 16]

C. Performance Measures:

when we go to the evaluation of performance any technique and algorithms with different, we must use some of the factor’s metrics to evaluate its. in this study we will use:

The Precision: the result of the divide of the quantity of correctly classified samples on the total of samples, then after taking the percentage of the result.

$$\boldsymbol{Precision} = \frac{Classification\ True}{(Classification\ True + Performed\ Classification\ )}$$

Recall: It is calculated the number of times that a classification true (positives), divided by the number of times that the class appears in the test data.

$$\boldsymbol{Recall} = \frac{Classification\ True}{(Classification\ True + Test\ Data\ Class)}$$

F1 score it is convenient to combine recall and precision into a single metric.

$$\boldsymbol{F1-Score} = \frac{2 * \mathit{Classification\ True}}{(\mathit{Classification\ True} + \mathit{Performed\ Classification} + \mathit{Test\ Data\ Class})}$$

D. Studying, analyzing the proposed system

Studying, analyzing, and investigating the recent related studies that investigated smart requirements classification for any system.

Most of the previous studies agree that Normalization is the initial step where they clean the data from the word is over, then Vectorization (feature extraction) the requirements corpus (normalized) are changed into mathematical vectors that better represent the data contained in those requirements, the third step is Organization in it the vectors obtained in the Vectorization are used to train and predict the cataloging models of the two algorithms used in this work: SVM and kNN, the last step is Valuation(Evaluation) that is the final stage of an organization (classification) where the outputs of the requirement's organization categories (labels) predictions and the true labels of these requirements are used to compute the performance procedures. Figure 1 show these steps.

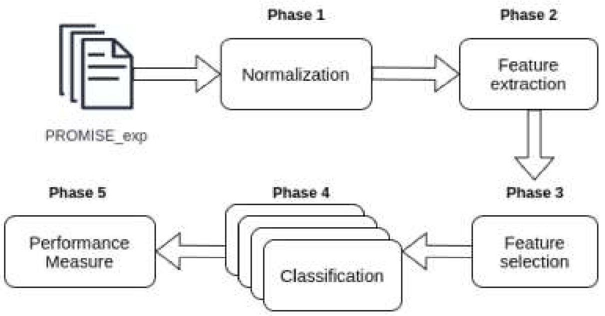

Figure 1. Phases of requirements classification

To make the testing, they used any software program for writing code, and the language can use for classification Python language and can also use a public database called PROMISE exp. This database's an expanded kind of the original PROMISE dataset. The original storage is made up of a collection of requirements labeled (255 FRs and 370 NFRs), the NFRs can divide into 11 different types. Figure 2 shows visually how deranged the modules are.

All requirements are written (documents) in the database have gone complete a normalization procedure, and implement all steps, after the last process (normalized) and vectorized corpus was used for another process(training and performance testing) using two algorithms(SVM and kNN). The SVM and KNN were used to categorize requirements into two different types of requirement (FR and NFR), then NFR classify into (11) sub-classes, now we make the testing in (3) cases:

- *Case 1 classification (2) group - Functional and Non-Functional Requirements*
- *Case 2 classification a Non-Functional Requirements (11 group)*
- *Case 3 classification with 12 groups*

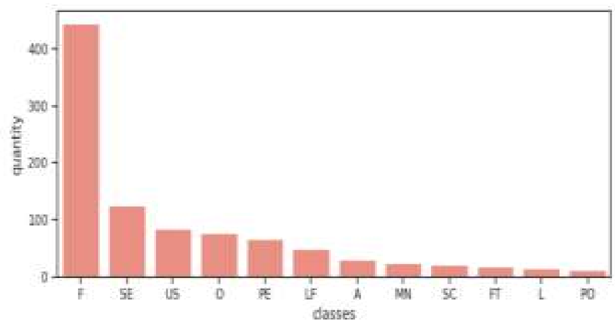

Figure 2. Distribution of requirement classes on dataset

IV. Result and Discussion:

We show the output of the studying and talk over its effect. We did three cases to determine the better use of BoW technique with ML algorithm (SVM or KNN) and dependent for classifying software requirements: the first: performance in classification of requirements (FR and NFR); secondary performance in subcategories (11 parts) classification of NFR; and third performance in subcategories (12parts) classification of requirements (FR and NFR). In the study of these three cases, we used the BoW technique with using (ML) algorithms (SVM and KNN) described in Section 3.2. The Performance Measures used to determine the algorithms were: Accuracy) Precision), Recall, and F1 score (F-measure).

The Effectiveness calculates of ML algorithms when classifying the requirements into two categories (FR and NFR) go to locating the quality of the requirements type between (FR AND NFR). Table 1 shows the output of case 1 (classification FR or NFR).

The Effectiveness calculates of ML algorithms when classifying the requirements to (11) subcategories classification of NFR (Availability, Performance, Security, etc.) go to locating the quality of these classification type (11) subcategories. Table 2 shows the output of case 2 (classification subcategories of NFR).

Performance Measures	Case 1	
	SVM	KNN
Precision	0.90	0.82
Recall	0.90	0.82
F1-score	0.90	0.82

Table 1. The output of case 1 (classification FR or NFR)

Performance Measures	Case 2	
	SVM	KNN
Precision	0.68	0.56
Recall	0.67	0.48
F1-score	0.66	0.49

Table 2. The output of case 2 (classification subcategories of NFR)

The Effectiveness calculate of ML algorithms when classify the requirements to (12) subcategories classification of NFR (Availability, Performance, Security etc.) and FR go to locating the quality of these classification type (12) subcategories. Table 3 show the output of the case 3 (classification subcategories of NFR and FR)

Performance Measures	Case 3	
	SVM	KNN
Precision	0.73	0.67
Recall	0.73	0.69
F1-score	0.72	0.68

Table 3. The output of case 3 (classification subcategories of NFR and FR)

It is possible to notice that SVM had the better performance than KNN with the same technique (BoW) in the classification with its measuring of performance (Precision, Recall and F1-score) in the average value of (*0.76*). Table 3 show the output of the all cases. Proposing a technique to reduce the time used and increase the accuracy of the classification through merging and modifying the algorithms as Term Frequency–Inverse Document Frequency (TF-IDF) needed and analyzing them based on the two groups as FR and NFR.

V. Conclusion and Future Work

The study explains when we use the text Vectorization technique (BoW) with two ML algorithms (SVM and KNN) to Classifies requirements for two categories (NFRs and FR), We evaluate the use of these algorithms take the PROMISE_exp database. We doing some cases to compare the Performance Measures (recall, precision, and F1 score (F-measure) of the classification output). We found that when the use of (BoW) with SVM algorithms has the better performance measures for requirements classification from (BoW) with KNN, with an F1 score (F-measure) of 90% on the binary classification (FR or NFR),66% in (11) subcategories on type of NFR (Availability, Performance, Security, etc.), classification and 72% on the 12- subcategories classification. Studies have discovered that not Enough data set because the volume of FR and NFR (requirements) of some classification (labels) in it is smaller. classification Software Requirements using ML helps software engineer (developers) document their software or program more Precision and validation, decreasing making and rework the program more flexible to understand and use. We expect that the study helps programmers to use BoW techniques with SVM algorithms to classification software requirements categorization automatically and know what the stakeholders want. Furthermore, the classification of requirements making can serve as a way and resource for another study, aiding who search in the subjects to take the automatize classification SVM algorithm that considered the better accuracy and easy use this process (classification of requirements NFR and FR).

VI. 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] Zayed, M. and Adel, M. Automatic Software Requirements Classification: A Systematic Literature Review. Information Journal of computers and IT. pp.29-37.

[4] Navarro-Almanza, R., Juarez-Ramirez, R. and Licea, G., 2017, October. Towards supporting software

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.

[5] Lu, M. and Liang, P., 2017, June. Automatic classification of non-functional requirements from augmented app user reviews. In Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering (pp. 344-353).

[6] 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.

[7] 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.

[8] Verma, S., 2014. Analysis of Strengths and Weakness of SDLC Models. International Journal of Advance Research in Computer Science and Management Studies, 2(3).

[9] Sabale, R.G. and Dani, A.R., 2012. Comparative study of prototype model for software engineering with system development life cycle. IOSR Journal of Engineering, 2(7), pp.21-24.

[10] Mishra, A. and Dubey, D., 2013. A comparative study of different software development life cycle models in different scenarios. International Journal of Advance research in computer science and management studies, 1(5).

[11] Do, T.N., Nguyen, V.H. and Poulet, F., 2008, October. Speed up SVM algorithm for massive classification tasks. In International conference on advanced data mining and applications (pp. 147-157). Springer, Berlin, Heidelberg.

[12] Wang, Y., Zhang, F. and Chen, L., 2008, August. An approach to incremental SVM learning algorithm. In 2008 ISECS International Colloquium on Computing, Communication, Control, and Management (Vol. 1, pp. 352-354). IEEE.

[13] Ko, Y., Park, S., Seo, J. and Choi, S., 2007. Using classification techniques for informal requirements in the requirements analysis-supporting system. Information and Software Technology, 49(11-12), pp.1128-1140.

[14] AlZu'bi, Shadi, Sokyna Al-Qatawneh, and Mohammad Alsmirat. "Transferable hmm trained matrices for accelerating statistical segmentation time." In 2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 172-176. IEEE, 2018.

[15] Al-Ayyoub, Mahmoud, Shadi M. AlZu'bi, Yaser Jararweh, and Mohammad A. Alsmirat. "A gpu-based breast cancer detection system using single pass fuzzy c-means clustering algorithm." In 2016 5th ICMCS, pp. 650-654. IEEE, 2016.

[16] Al-Zu'bi, Shadi, Mahmoud Al-Ayyoub, Yaser Jararweh, and Mohammed A. Shehab. "Enhanced 3d segmentation techniques for reconstructed 3d medical volumes: Robust and accurate intelligent system." Procedia computer science 113 (2017): 531-538.

[17] Alasal, S.A., Alsmirat, M., Baker, Q.B. and Alzu'bi, S., 2020. Lumbar disk 3D modeling from limited number of MRI axial slices. International Journal of Electrical and Computer Engineering, 10(4), p.4101.

[18] AlZu'bi, S. and Jararweh, Y., 2020, April. Data Fusion in Autonomous Vehicles Research, Literature Tracing from Imaginary Idea to Smart Surrounding Community. In 2020 FMEC (pp. 306-311). IEEE.

[19] Rezaee, Hamideh, Ah Aghagolzadeh, M. Hadi Seyedarabi, and Snadi Al Zu'bi. "Tracking and occlusion handling in multi-sensor networks by particle filter." In 2011 IEEE GCC Conference and Exhibition (GCC), pp. 397-400. IEEE, 2011.

[20] Jararweh, Yaser, Shadi Alzubi, and Salim Hariri. "An optimal multi-processor allocation algorithm for high performance GPU accelerators." In 2011 IEEE (AEECT), pp. 1-6. IEEE, 2011.

[21] Mansour, A.M., Abdallah, J., Obeidat, M.A.," An efficient intelligent power detection method for photovoltaic system," International Journal of Circuits, Systems and Signal Processing, vol. 14, pp. 686–699, 2020.

[22] Ayman M Mansour, " Cooperative Multi-Agent Vehicle-to-Vehicle Wireless Network in a Noisy Environment," International Journal of Circuits, Systems and Signal Processing, vol. 15, 2021.

[23] Suhail Sharadqah, Ayman M Mansour, Mohammad A Obeidat, Ramiro Marbello and Soraya Mercedes Perez, "Nonlinear Rainfall Yearly Prediction based on Autoregressive Artificial Neural Networks Model in Central Jordan using Data Records: 1938-2018" International Journal of Advanced Computer Science and Applications(IJACSA), 12(2), 2021.

[24] Hawashin, B., Lafi, M., Kanan, T. and Mansour, A., 2020. An efficient hybrid similarity measure based on user interests for recommender systems. Expert Systems, 37(5), p.e12471.

[25] Hawashin, B. and Mansour, A., 2016. An efficient agent-based system to extract interests of user groups. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 1).

[26] Hawashin, B., Fotouhi, F. and Truta, T.M., 2011, March. A privacy preserving efficient protocol for semantic similarity join using long string attributes. In Proceedings of the 4th International Workshop on Privacy and Anonymity in the Information Society (pp. 1-7).

[27] AlZu'bi, Shadi, Sokyna AlQatawneh, Mohammad ElBes, and Mohammad Alsmirat. "Transferable HMM probability matrices in multi-orientation geometric medical volumes segmentation." Concurrency and Computation: Practice and Experience 32, no. 21 (2020): e5214.

[28] AlZu'bi, Shadi, Darah Aqel, Alaa Mughaid, and Yaser Jararweh. "A multi-levels geo-location based crawling method for social media platforms." In 2019 Sixth SNAMS, pp. 494-498. IEEE, 2019.