Hybrid Feature Selection based Classifier for Non Functional Requirements

[1]Naina Handa, [2]Dr. Anil Sharma, [3]Dr. Amardeep Gupta
1Research Scholar, Lovely Professional University, Phagwara (Punjab) naina_pisces17@yahoo.co.in
2Associate Professor, Lovely Professional University, Phagwara (Punjab)
anil.19656@lpu.co.in
3Principal, JC D A V College, Dasuya (Punjab)
dramardeepgupta@gmail.com

Article Info
Page Number: 01 - 11
Publication Issue:
Vol 71 No. 2 (2022)

Abstract
Requirements can be segregated into Functional and Non Functional Requirements. The classification and sub-classification of requirements is a very crucial task. Feature Selection (FS) plays a significant role in classification. The goal of FS is to identify the most important aspects of a problem domain. It is quite beneficial in terms of increasing computing speed and accuracy. Identifying meaningful features from hundreds or even thousands of similar features is a difficult task. In this research, we present a hybrid FS technique for NFRs classification that incorporates two FS methods – filters and wrappers. Candidate features are initially chosen from the original feature set using two filter approaches, and then the intersection of these two sets is refined using more precise wrappers with the Bayesian Belief Model. Both the filters and the wrappers are used in this hybrid technique. The mechanism is investigated using a primary dataset of NFRs. The results of the experiments demonstrate that using a reduced feature set can improve prediction accuracy.

Article History
Article Received*: 29 November 2021*
Revised*: 15 December 2021*
Accepted*: 08 January 2022*
Publication*: 05 February 2022*

Keywords: Wrapper, Filter, Bayesian Belief Model.

Introduction

ML works on a simple rule “if you put garbage in, you will get the garbage out”. Feature Selection (FS) has been investigated by the data mining and ML sectors. FS is the process of identifying a small number of linked features that are sufficient for learning the target concept. FS refers to the concept of feature relevance, redundancy, and complementarity (synergy) [1]. FS has been investigated by the data mining and ML sectors. Features are sometimes referred to as attributes or variables. Features are the attributes of the system that have been measured and designed from the original input variables. It's a technique for extracting the most significant characteristics from the original features to reduce the dataset's size. In data mining, feature selection is a crucial pre-processing step that identifies useful feature subsets for classification. Furthermore, most classification algorithms presume that all features have the same relevance level when they are classified [2]. To improve classification results, dimension reduction strategies that identify a smaller collection of features are required. A dataset often has many logically redundant and interrelated characteristics, resulting in excessive computational complexity and poor interpretation. Many dimension reduction strategies, such as feature selection, feature extraction, and feature reduction, have been studied in the literature. Feature reduction reduces the original feature space

by removing conceptually redundant information without affecting classification accuracy [3]. By altering the original feature space, feature extraction converts a high-dimensional feature space into a unique low-dimensional feature space. FS for classification is a very active data mining research subject. The effort of a classifier employing FS is reduced by selecting only relevant features before classification. The classification performance has improved with the FS method. It detects only helpful features, discards irrelevant ones, and minimizes input dimensionality, making the classifier's implementation easier and the processing rate faster [4]. Many FS algorithms have been developed by researchers with various selection criteria. However, no single criterion is the best for all applications. FS seeks to either remove ineffective and superfluous features or pick effective and interconnected elements. Accurately measuring the correlations between candidate characteristics, selected features, and categories in the selection process, especially for high-dimensional and small-sample-size data, is a difficult challenge. In general, FS is important for preventing over-fitting, facilitating data visualization, lowering storage and processing costs, and boosting prediction algorithm accuracy. Furthermore, FS preserves the physical meaning of the original characteristics and improves the data model's readability and interpretability [5]. It is a method of discovering a small number of interconnected elements that are sufficient for learning the target notion. Features are sometimes referred to as attributes or variables. Features are the attributes of the system that have been measured and designed from the original input variables [6]. A dataset often has many logically redundant and interrelated characteristics, resulting in excessive computational complexity and poor interpretation. Many dimension reduction strategies, such as FS, feature extraction, and feature reduction, have been studied in the literature. Feature reduction reduces the original feature space by removing conceptually redundant information without affecting classification accuracy [7].

FS is an important part of any ML model since it offers the following benefits:

- FS can cut model training time in half.
- FS makes the model easier to understand by simplifying it.
- FS avoids the dimensionality curse.
- By eliminating over fitting, FS improves generalization.
- FS improves the performance of the ML model, which was created with all of the features.
- With unseen data, FS delivers more robust generalization and a faster reaction.FS improves the accuracy of the model by choosing the correct subset of data

The main goal of the FS is to retrieve the optimal subset from the feature set that yields the lowest generalization error. By altering the original feature space, feature extraction converts a high-dimensional feature space into a unique low-dimensional feature space It's a technique for extracting the most significant characteristics from the original features to reduce the dataset's size. In data mining, FS is a crucial pre-processing step that identifies useful feature subsets for classification. Furthermore, most classification algorithms presume that all features have the same relevance level when they are classified [8]. To improve classification results, dimension reduction strategies that identify a smaller collection of features are required. In data mining, FS is a crucial pre-processing step that identifies useful feature subsets for classification. Furthermore, most classification algorithms presume that all features have the same relevance level when they are

classified. To improve classification results, dimension reduction strategies that identify a smaller collection of features are required [9]. FS usually results in a feature subset, intending to obtain the most informative subset by selecting key features from the original feature set and removing unnecessary features. Although candidate traits and target classes are invariant during the selection process, the relationship between them is not. While adding features to the subset, it will be a constantly changing value. As a result, measuring the links between candidate features, selected features, and categories in the selection process is difficult. Classifier-independent Filter, classifier-dependent Wrapper, and Embedded techniques are three types of supervised algorithms that deal with diverse interactions between features and classifiers [10]. The explanation of various techniques is given below:

Filter Method:

Filters examine statistical aspects of data and rank features using heuristic scoring rules. Because this procedure is independent of classifiers, the characteristics chosen may not be the best for every classifier. As statistical and heuristic methods, however, the majority of them are computationally efficient and general [11]. Filtering methods have the advantage of being fast and scalable to huge datasets. Instead of employing classifiers, filters evaluate the relevance of characteristics individually based on predetermined metrics. Features are measured and rated according to their importance using simple measurements like distance, dependency, and information. Chi-square, Improved Gini index, and Mutual Information are the most commonly used filters. Each feature is assessed individually using its general statistical features in the filter approach. There is no unique learning model used in the filter technique [12]. As a result, it is unaffected by the classifier. The features were graded according to particular criteria, and the features with the highest scores were chosen. Following that, these features are fed into the classifier as input. The Filter technique is independent of the ML algorithm and relies on the statistical features of the training data. As a result, the computational cost is always minimal, but the outcomes aren't always adequate. It is a pre-processing step filter method that selects features independently of any ML algorithm. A statistical measure is assigned a scoring in each feature. The selection of features is not influenced by the nature of the classifier used. The ranking of the score is based on the relevance approach [13].

Wrapper method:

Wrappers employ training or test accuracy as the measure of feature subset when given a specific learning algorithm. Wrappers can choose the best features for a given classifier, but they are computationally expensive and more likely to overfit. The Wrapper technique, on the other hand, evaluates the selected features using the results of a present learning algorithm[14]. The wrapper technique is more efficient than the filter method because it analyses the interplay between feature subset search and the learning model. However, it takes time, especially when dealing with large datasets with many dimensions. The wrapper method selects the best feature subset using ML techniques. The wrapper technique's quality is determined by the accuracy of the model. The Wrapper technique, unlike the Filter method, chooses features using a classifier. It requires more excellent computational resources, and so it is expensive [10]. The important examples of Wrapper

techniques are Recursive Feature Elimination, Forward FS, and Backward Feature Elimination strategy.

Embedded methods:

FS should be included in the training process and guided by the unique structure. These methods have the advantage of interacting with learning algorithms, but constructing an appropriate optimization function is difficult. The classifier's embedded approach looks for the best feature subset. The hybrid or ensemble technique is another name for the embedded approach. Both the filter and wrapper approaches are merged in a hybrid approach. As a result, it integrates the computational effectiveness of the filter approach with the excellent performance of the wrapper approach. As previously stated, both the filter and wrapper methods have benefits and drawbacks. Combining these two FS methods to create a hybrid selection method has a wide range of applications for obtaining high-precision input features quickly. It starts by filtering out the majority of the redundant or collinear features. The remaining features and concentration data are then passed as input parameters to the wrapper method for FS. The hybrid method usually achieves high accuracy, a feature of wrappers that has high efficiency, characteristic of filters. FS is the process of picking a meaningful subset with the given variables to build a model. Its goal is to enhance accuracy by allowing ML algorithms to train faster, lower complexity, reduce overfitting, and extract datasets [15]. FS techniques include super greedy and greedy algorithms. The FS technique reduces the number of redundant or unnecessary features, resulting in less information loss. It is utilized in domains that are complex and have a small number of data samples. The Embedded approach is a combination of the Filter and Wrapper methods. Regularization or Penalization methods are the most common types of embedded methods. This method has its built-in FS methods. In the Embedded method, it is impossible to reflect upon the validity and relative ability of the techniques in the said scenario. Regularized algorithms are LASSO, Elastic Net, and Ridge Regression. FS can be performed using the mutual information test on features vs. classes and features vs. features. This is based on the information theorem, which examines the relationship between characteristics and classes to eliminate features that are either redundant or unrelated to the class. A greedy FS algorithm was proposed in their research. The filters go through three stages: feature set development, measurement, and learning algorithm testing. A feature subset is created during the feature set generation stage. The measurement phase follows, which measures the information contained in the current feature set. While the outcome does not meet the stopping requirement, the preceding procedures will be repeated. The stop criterion in this phase could be a measurement result threshold. A fresh feature set would be developed and the measurement would be repeated if the result did not meet the threshold. As a result, the most informative elements would be included in the final feature set. Finally, a learning algorithm precedes the testing process. The outcome provides the results of the testing of the selected feature. It demonstrates the operation of wrappers. It's the same as the filters, except that a learning algorithm replaces the measurement stage. And it is for this reason that the wrappers are always slow. Wrappers, on the other hand, might obtain higher FS outcomes in most circumstances because of the learning process. When the result starts to deteriorate or the quantity of features hits a predetermined threshold, the procedure is terminated.

Although the filters work rapidly, the outcomes are not always satisfactory. The

wrappers have high classification accuracy, but they process slowly. Furthermore, because the filters calculate information from features, their FS results will be influenced by the measured information of the features. Because the wrappers employ the learning algorithm to make their decisions, the learning algorithm biases their classification results [16].
The most essential artifacts produced during the software development life cycle are software requirements specifications, which are made up of both Functional and Non Functional Requirements (NFR). NFR indicates how the software system will give the tools to fulfil functional activities, whereas functional requirements provide software behaviour directly listed by stakeholders. Neglecting NFR has been linked to project failure or increased production costs [17]. As a result, early NFR detection is critical for developing high-quality software while also lowering development costs since it allows system-level restrictions to be evaluated and included in early architectural designs. However, in terms of time, budget, and accuracy, manually classifying requirements is not practical. In diverse disciplines, correlation, information gain, and the Bayesian Belief Method (BBM) are all frequently used FS algorithms. The hybrid technique we presented in this paper makes a significant contribution to the literature. The filter and wrapper benefits are combined in the hybrid algorithm. In comparison to existing FS techniques, the significant group selected by our system has higher accuracy identification rates.

Literature Review

This section provides a thorough examination of the present FS and NFRs elicitation methods. Though the review of the literature confirms ML, FS, and NFRs as established fields of study, it also highlights various limitations and undiscovered regions in these domains.

Cleland et al. [18] established the NFR classification strategy and utilized an information retrieval technique. Researchers have suggested a method for detecting ambiguity, inconsistency, incompleteness, and redundancy of NFR in software requirement specifications.

Hussain et al. [19] provided a method for classifying text requirements as functional or non-functional based on linguistic information. A method for classifying text-based non-functional requirements was presented using a convolutional neural network.

Lu and Lang [20] have depended upon supervised learning methods for extracting and considering NFRs from user reviews. The four classification techniques BoW, CHI2, TF-IDF, and AUR-BoW, have been joined with three ML algorithms J48 and Bagging and Naive Bayes, to classify user reviews. Bagging has been inferred as the best technique for NFRs classification. This paper has used techniques such as feature selection, ensembling technique, and bagging to achieve higher accuracy.

Martino et al. [21] evaluate that the requirement specification is a complex task concerning, cloud computing, particularly with developing stakeholders who have ever-changing needs. So, the authors have proposed automatic modeling and classification of requirements stated in natural language form. The target data has been used from the Promise repository.

Abad et al. [22] contributed a methodology for pre-handling requirements that normalizes and standardizes requirements before applying classification algorithms and improves performance. Different ML techniques have been compared. The focused NFRs are usability, availability, or performance.

Kiran and Ali [23] explored that the requirement elicitation process is very complex and critical as engineers from various locales of the world build up the framework, so it's tough to accumulate requirements for such frameworks. This paper focuses on how the procedure of requirement elicitation is completed for open source software and the various ways utilized to streamline the process of requirement elicitation by using a variety of tools, strategies, and methodologies.

Tiwari and Rathore[24] presented an approach to pick a subset of techniques for an optimal output as part of the Requirement Elicitation process. Requirement engineering is heavily influenced by three factors: people, processes, and projects. This work aims to provide significant insights into the features of diverse requirements elicitation approaches. A series of case studies will be used to evaluate and offer context for the selection of the Requirement Elicitation technique.

However, no guideline has been found yet to select the best pair of feature extraction and machine learning methods for NFR classification by investigating their performance.

Proposed Framework

The whole process of this framework is divided into the following four steps Data Collection and Pre-processing, Feature Selection and Extraction, and Classifying requirements.

Data Collection and Pre-Processing phase: The dataset used in the current research has been gathered through the catalog from IT professionals and academicians. The collected The collected dataset has been converted into numeric form, missing data values are filled and repetition has been removed from the dataset. So that the data is ready for experiment purposes.

Feature Selection and Extraction Phase: A algorithm has been proposed for FS purposes for NFRs classification. In this algorithm advantages of the filter and wrapper method has explored and the accuracy of the algorithm has been validated with the help of the primary dataset. The irrelevant features are eliminated. The eliminated features are features that have no bearing on the output classes and are redundant in light of other input features. FS is developing in two major directions. The wrappers and the filters are the two types of filters. The filters are quick to operate and use a basic measurement, although the results are not always good. The wrappers, on the other hand, ensure good results by reviewing learning results, but they are extremely slow when applied to large feature sets with hundreds or even thousands of features. While the filters are quite effective at choosing features, they are unreliable when applied to large feature sets. This study attempts to address the issue by incorporating wrappers. It is a hybrid FS paradigm that uses both filter and wrapper techniques, rather than a pure wrapper approach. The working of the algorithm is explained in Figure 1 as shown below:

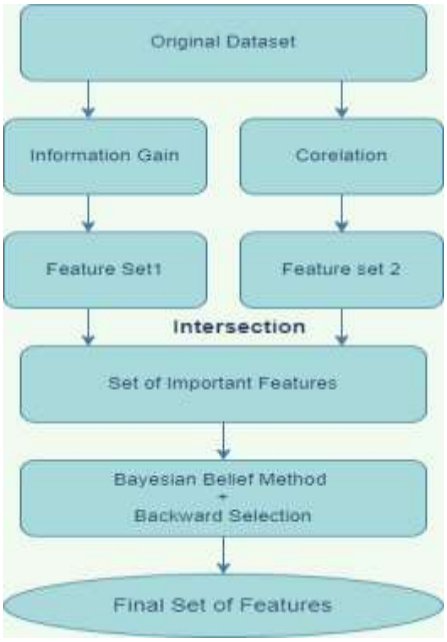

Figure 1

Two feature sets are first filtered out by correlation and information gain in our method. A wrapper process is used to integrate the feature sets and fine-tune them. We use both the filter and the wrapper to our advantage. It is not as fast as a pure filter, but it can produce better results than a filter. Most notably, when compared to a pure wrapper, the computing time and complexity can be lowered. In terms of NFR prediction, the hybrid technique is more feasible. To pick a smaller feature set that makes the classifier more accurate and faster, FS approaches have been applied to classification problems. The dataset is classified into two groups significant and non-significant NFRs. The Bayesian Belief Model (BBM) is used as the machine learning technique in the wrapper method. BBM is a probabilistic-based model. BBM is used in various applications for prediction purposes. The classification is represented in Figure 2. The backward search method has been explored as a search method.

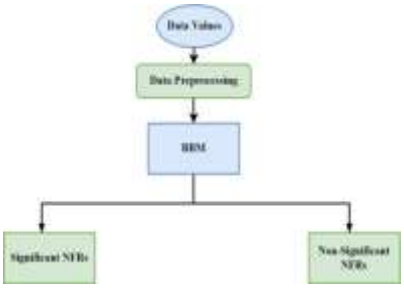

Figure 2

The hybrid FS technique has taken the advantage and increased the classification accuracy by pure filters while also reducing the processing time by utilizing pure wrappers. In the hybrid feature selection, the two filter techniques are chosen as starting steps and remove irrelevant and redundant features. When the preliminary procedure is complete, two feature subsets are selected by the information gain and correlation. These generated features by both of the methods are considered the most class-related features of all features. So selecting a final feature set for the

wrapper method is a problem now. The key to combining these feature sets is to perform the intersection means AND of these two feature sets. The feature set1 selected by information gain and feature set2 selected by correlation are intersected and only these features are selected which are common in both of the feature sets. The wrapper procedure with a machine learning algorithm would examine the features. The search method and machine learning algorithm are required to be selected for the wrapper method. The various kinds of search methods are available sequential backward and sequential forward search. Sequential search methods are characterized by a dynamically changing number of features included or eliminated at each step. Sequential forward search starts with an empty set and adds one feature at a time till the test result starts to get worse. Backward search has started with all features and removes one feature at a time, and then the learning model would be applied to test its result. The same process is performed iteratively till the number of features reaches a predefined threshold or the test result gets worse. The wrapper technique works with a machine learning algorithm. BBM has been explored in the current work and classified in the dataset. The algorithm of the present work is presented in algorithm 1as shown below:

Algorithm 1 Hybrid Feature Selection Procedure

1: Input: Feature Set FS=f1,f2,. . . . . ,fn
2: Output: S- the selected feature subset
3: Select the features based on the correlation filter method and store them in Feature Set 1
4: Select the features based on the Information Gain filter method and store them in Feature Set 2
5: Perform intersection between Feature Set 1 and Feature Set 2 and store in Feature Set 3
6: ** Extract common features**
7: Provide Feature Set 3 to wrapper method as input
8: Apply wrapper method with Bayesian Belief Model and Backward Selection method
9: Get S as the final selected feature subset

Experimental Setup

This part shows the experimental investigation, which used a usability-based primary dataset and analysed the results. To see how successfully the model learns to categorize software requirements, the accuracy metric has been used. Unlabeled text strings were utilized for testing and labeled data was used for training. The tests were run on a laptop with an Intel Core i7 processor with 32GB of RAM. The Pandas, NumPy, Sklearn, and Matplotlib packages were used to import the data and evaluate the findings. The prepared dataset is trained and evaluated to determine the proposed model's classification accuracy.

Results

In the preliminary screening process, the information gain and correlation are used to filter the features. The primary dataset based on NFRs has been used for experimental purposes. The hybrid feature selection algorithm has designed and evaluated the classification of NFRs into significant and non-significant groups. The performance of the proposed technique is compared with none feature selection approach and the other 2 frequently used filter-based feature selection approaches, such as information gain and correlation. Correlation-based Feature Selection is a basic filter technique that ranks feature subsets using a heuristic evaluation

function based on correlation [7]. The evaluation function is biassed in favor of subsets with attributes that are substantially correlated with the class but uncorrelated with one another. Irrelevant traits should be discarded because their correlation with the class will be low. Because they will be substantially associated with one or more of the remaining features, redundant features should be screened out. The extent to which a feature predicts classes in portions of the instance space not already predicted by other characteristics will determine its acceptability.

In the experiments, the prepared dataset is trained and tested to evaluate the classification accuracy of the proposed model. The key idea of the proposed method is to combine the efficiency of the filter method and the Accuracy of the wrapper method. A model has been designed where the two filter techniques, such as information gain and correlation, have been utilized as preliminary procedures for the model. The intersection of both feature sets has been computed. The resultant feature set further provides the wrapper method where the backward search technique and BBM were used as machine learning models. The results are depicted as a graph as shown in figure 3.

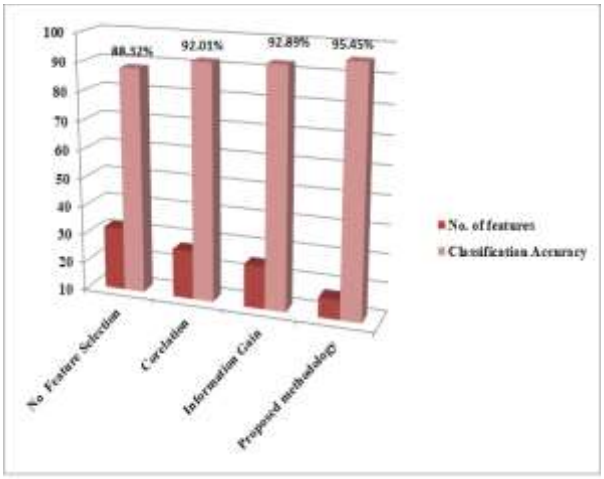

Figure 3

The results have shown that when no feature selection technique has been applied, and all 32 features are considered for classification the Accuracy is 88.32%. After applying the correlation and information gain with features set 27 and 25 the Accuracy is 92.01% and 92.89%. But for the proposed model, the features are reduced to 17 and the Accuracy is 95.45%, which is relatively higher than other techniques. Hence, the proposed hybrid feature selection model is relatively suitable for the NFRs classification. So it is concluded that the proposed technique is quite efficient and also saves the computation time of the model by removing the irrelevant features.

Conclusion

A hybrid FS technique has been applied and tested on the NFRs-based dataset in this paper. The key idea of the proposed method is to combine the efficiency of the filter method and the accuracy of the wrapper method. A framework has been designed where the two filter techniques such as information gain and correlation have been utilized as a preliminary procedure for the model and

the intersection of both feature sets have computed. The resultant features are further provided to the wrapper method and BBM is used as a machine learning model with a backward search technique. The experiment has performed on the primary dataset based on NFRs. The result has shown that the proposed model has outperformed other techniques with an accuracy score of 95.45%. So the present technique is quite suitable for NFRs classifications.

References

[1]. Casamayor, A., Godoy, D., & Campo, M. (2010). Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. Information and Software Technology, 52(4), 436-445.

[2]. Rahimi, M., Mirakhorli, M., & Cleland-Huang, J. (2014, August). Automated extraction and visualization of quality concerns from requirements specifications. In 2014 IEEE 22nd international requirements engineering conference (RE) (pp. 253-262). IEEE.

[3]. Condori-Fernandez, N., & Lago, P. (2018). Characterizing the contribution of quality requirements to software sustainability. Journal of Systems and Software, 137, 289-305.

[4]. Tamai, T., & Anzai, T. (2018). Quality Requirements Analysis with Machine Learning. In ENASE (pp. 241-248).

[5]. Binkhonain, M., & Zhao, L. (2019). A review of machine learning algorithms for identification and classification of non-functional requirements. Expert Systems with Applications.

[6]. Riaz, M., King, J., Slankas, J., & Williams, L. (2014, August). Hidden in plain sight: Automatically identifying security requirements from natural language artifacts. In 2014 IEEE 22nd International Requirements Engineering Conference (RE) (pp. 183-192). IEEE.

[7]. Gazi, Y., Umar, M. S., & Sadiq, M. (2015). Classification of NFRs for Information System. International Journal of Computer Applications, 115(22).

[8]. Mateev, M., & Poutziouris, P. (Eds.). (2019). Creative Business and Social Innovations for a Sustainable Future: Proceedings of the 1st American University in the Emirates International Research Conference--Dubai, UAE 2017. Springer.

[9]. Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., & Gao, Z. (2017). A hybrid feature selection algorithm for gene expression data classification. Neurocomputing, 256, 56-62.

[10]. Ramadhani, D. A., Rochimah, S., & Yuhana, U. L. (2015). Classification of non-functional requirements using semantic-FSKNN based ISO/IEC 9126. Telkomnika, 13(4), 1456.

[11]. Gazi, Y., Umar, M. S., & Sadiq, M. (2015). Classification of NFRs for Information System. International Journal of Computer Applications, 115(22).

[12]. Sharma, V. S., Ramnani, R. R., & Sengupta, S. (2014, June). A framework for identifying and analyzing non-functional requirements from text. In Proceedings of the 4th international workshop on twin peaks of requirements and architecture (pp. 1-8). ACM.

[13]. Idri, A., Hosni, M., Abnane, I., de Gea, J. M. C., & Alemán, J. L. F. (2019, April). Impact of Parameter Tuning on Machine Learning Based Breast Cancer Classification. In World Conference on Information Systems and Technologies (pp. 115-125). Springer, Cham. 14. Ezami, S. (2018). Extracting non-functional requirements from unstructured text (Master's thesis, University of Waterloo).

[14]. Di Martino, B., Pascarella, J., Nacchia, S., Maisto, S. A., Iannucci, P., & Cerri, F. (2018, May). Cloud Services Categories Identification from Requirements Specifications. In 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA) (pp. 436-441). IEEE.

[15]. Kiran, H. M., & Ali, Z. (2018). Requirement Elicitation Techniques for Open Source Systems: A Review. International Journal of Advanced Computer Science and Applications, Pakistan, 330-334.

[16]. Kopczyńska, S., Nawrocki, J., & Ochodek, M. (2018). An empirical study on catalog of non-functional requirement templates: Usefulness and maintenance issues. Information and Software Technology, 103, 75-91.

[17]. J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, “The detection and classification of non-functional requirements with application to early aspects,” in 14th IEEE International Requirements Engineering Conference (RE’06). IEEE, 2006, pp. 39–48.

[18]. A. Hussain and E. O. Mkpojiogu, “An application of the iso/iec 25010 standard in the quality-in use assessment of an online health awareness system,” Jurnal Teknologi, vol. 77, no. 5, pp. 9–13, 2015.

[19]. M. Lu and P. Liang, “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, 2017, pp. 344–353.

[20]. B. Di Martino, J. Pascarella, S. Nacchia, S. A. Maisto, P. Iannucci, and F. Cerri, “Cloud services categories identification from requirements specifications,” in 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA). IEEE, 2018, pp. 436–441.

[21]. Z. S. H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe, and K. Schneider,“What works better? a study of classifying requirements,” in 2017 IEEE 25th International Requirements Engineering Conference (RE). IEEE, 2017, pp. 496–501.

[22]. H. M. Kiran and Z. Ali, “Requirement elicitation techniques for open source systems: A review,” International Journal of Advanced Computer Science and Applications, Pakistan, pp. 330–334, 2018.

[23]. S. Tiwari and S. S. Rathore, “A methodology for the selection of requirement elicitation techniques,” arXiv preprint arXiv:1709.08481, 2017.