https://doi.org/10.33472/AFJBS.6.9.2024.5087-5091

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Evaluation of Machine Learning Algorithms for Classifying Functional and Non-Functional Requirements

Sonali Idate[1], T.Srinivasa Rao[2]

[1]Research Scholar, Department of Computer Science and Engineering, GST, GITAM University, Visakhapatnam, India.
E-mail: sonaliidate@gmail.com

[2]Associate Professor, Department of Computer Science and Engineering, GST, GITAM University, Visakhapatnam, India.
E-mail: tsr.etl@gmail.com

Article History
Volume 6,Issue 9, 2024

Received:20 Feb 2024

Accepted : 28 Mar 2024

doi: 10.33472/AFJBS.6.9.2024.5087-5091

Abstract. The correct categorization of functional and non-functional requirements has become a significant task in requirement engineering (RE). The requirements for software are written in natural language text. Machine Learning (ML) - based approaches provide better results than traditional natural language processing. This paper evaluates Supervised Machine Learning (SML) algorithms that can automatically categorize requirements as functional (FR) and non-functional (NFR). However comprehensive evaluation of these ML approaches is still required. This paper selected Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT),and Random Forest(RF) ML algorithms. The application of ML in RE creates an interesting potential for the creation of new expert and intelligent systems to assist RE processes and tasks. In this paper, we discuss the treatment of the most distinctive features of FR and NFR, the sampling strategies used in additional data sets, and their impact on classification accuracy. The plan for future work is to study application of more algorithms and new features in order to improve the precision of the proposed models.

Keywords: Machine Learning, Natural Language Processing, Imbalanced Data, Classification, Functional Requirements, Non-Functional Requirements.

I. INTRODUCTION

In software designing requirement engineering is a crucial phase, in this phase tasks are to be performed by software are defined by functional requirements. Additionally, there are non-functional requirements, which outline how effectively software should perform those tasks. While the definition of FRs is generally accepted, this is not the case for NFRs. A FR describes the service that the software must provide[1]. It describes a software system or its component. A software system is set of a functions that are the inputs, the behavior, and the outputs, whereas NFRs specify limitations that have an impact on how the system should perform [1]. So to compare non-functional requirements, functional requirements describe how the system must function, and non-functional explain how the system should perform. Several requirements engineering researchers have aimed to automatically extract and categorize text of the requirements expressed in natural language. NFRs are silent with the elicitation to FRS and when structured in the text document after the elicitation phase [2]. Most of the NFRs become intertwined with FRs when requirement specification documents are commonly written and organized in accordance with FRS[3]. Further become difficult to distinguish FR and NFR, when designing software architectures, Architects must distinguish between NFRs and FRs to convert various requirements into various architectural aspects [4].

In the systematic review 24 existing ML-based approaches are studied to determine which Machine Learning (ML) algorithms have been used to classify FR, NFRs, or specific NFRs such as security, how they work, and how they have been evaluated [5]. This study identified a Support Vector Machine (SVM) as the most used algorithm followed by a Decision Tree (DT). This paper included two more ML algorithms Random Forest (RF)and Logistic Regression (LR) to evaluate the state of categorization of ML-based approaches and to determine which issues must be highlighted.

RQ1.Which feature model is to be used to cite the approach to get better results for classifying into FRs and NFRs?

RQ2.Among the selected ones when compared to other machine learning algorithms, which supervised ML algorithm is most effective on requirements classification tasks?

This work studied the comparison of four supervised classification algorithms –Random Forest (RF), Decision Tree(DT), Logistic Regression (LR), and Support Vector Machines (SVM). This research explores ways to enhance algorithms that automatically categorize software requirements. It also examines which popular machine learning techniques are effective for this purpose. This paper used supervised machine learning to examine how accurately requirements are classified as FR and NFR. This paper is organized as follows: In Section 2 contains a review of the literature. Section 3 presents the research methodology. Section 4 presented results answering research questions RQ1 and RQ2. And section 5 is a conclusion.

II. LITERATURE SURVEY

This section included some papers associated to survey papers, and papers analyzing and classifying requirements. The survey paper found in general, 24 included studies of process patterns for applying machine learning-based algorithms to identify and categorize FRS and NFRs in textual requirements. Generally, ML-based approaches work well, attaining over 70% accuracy in identifying and classifying NFRs. In the review paper by Binkhonain et. al. automatically identifying requirements is of least important in their review; in contrast, their review focuses solely on identifying NFRs using ML algorithms. [5]. Kurtanovic et. al. use the Support Vector Machines (SVM) technique to categorize FR, NFR, and non-functional requirement subcategories [6].

The PROMISE repository, which has the drawback of being imbalanced in terms of functional and non-functional criteria, was used by researchers[6]. In another paper after downloading and using an app, users can rate it and leave written reviews or comments for the developer. Clement, J used these comments to offer insightful data that could aid engineers in better comprehending consumer requirements and grievances during software evolution. This feedback may provide developers with useful information that will assist them in better understanding NFRs such as reliability, usability, portability, and performance [7]. Jindal et al. used a single machine learning technique to perform an automated analysis of various software requirement specifications from the PROMISE repository and implement binary classification on multiple types of only security NFR. [8]. In some papers, the dataset used by many researchers only has a few NFRs [9][10].

Existing studies have work focusing mainly on identifying NFRs. Also, a dataset has a limited set of NFRs. This paper offers a specification of a feature that result in the classification of requirements as FR and NFR and a comparison of four popular ML algorithms to study the best performance on requirement classification into FR and NFR. The enhanced version of the original PROMISE dataset [13] includes more labelled requirement sentences, including 444 functional and 525 non-functional requirements [12].

The elicitation of requirements is the primary task of requirements engineering [20]. Following the elicitation task, requirements are documented in textual form and further broken down into categories, the major ones to be: Non-Functional Requirements and Functional Requirements [11].

Functional Requirements (FRs) are phrases and propositions that describe potential inputs and events from the system as well as the desired outputs for the software and/or software components [11].A system or system component must be able to perform the specified function in order to meet the requirements stated in FRs, according to IEEE et al. [14]. FRs should not take into account any technological concerns and should be independent of design and implementation factors. FRs define a feature that will be offered to the user as a service of the system, defining a portion of the system's behavior as a response. [15].

Non-Functional Requirements (NFRs) characterize the quality attributes of the system to be developed and frequently have a greater influence on system architecture than functional requirements. An NFR is a set of constraints placed on the system being designed that, for example, determines whether it is user-friendly, practical, quick, or reliable. An NFR is a term used to characterize a system's non-behavioral features. It encapsulates the characteristics and limitations that the system must adhere to[15]. The system quality attributes such as portability, reliability,efficiency, usability, testability, understandability, and modifiability, among others. NFRs are further classified into subcategories such as product requirements, organizational or process-related requirements, and external requirements [16].

III. DATASET

A system for requirements categorization needs to be able to learn. The challenge of requirements engineering is to take a given set of data and perform automated requirements identification/classification tasks on that data. The existing “nfr”dataset was expanded by adding requirement text from the PURE dataset [24][25].

The dataset has the label "F" for the FR category's required text and "A, L, LF, MN, O, PE, SC, SE, US, FT, PO" for the NFR category. However, for the category labelled with “F” we used 1 and for NFR categories used 0 as shown in Fig 1.

Existing label	Requirement text	Proposed label
F	The Disputes System shall record the name of the user and the date for any activity that creates or modifies the disputes case in the system. A detailed history of the actions taken on the case including the date and the user that performed the action must be maintained for auditing purposes.	1
F	For any systematic (non-user initiated) action that occurs on a case such as the disputes aging process a case activity will be generated. Upon the next logon the user that initiated the dispute case must be notified that a systematic action has occurred on the dispute case.	1
F	All letter requests must be formatted according to guidelines specified by the Print Letter Utility system.	1
F	Any disputes cases that have been closed for over 6 months must be purged from the online disputes database.	1
O	The product shall adhere to the corporate Architecture guidelines	0
LF	The product shall comply with corporate User Interface Guidelines	0
LF	The product shall comply with corporate color scheme	0
LF	The appearance of the product shall appear professional	0
PO	The software product is expected to run on Windows or Linux platforms.	1
US	The product shall be easy to use by Adjusters and Collision Estimators. 95% of Adjusters and Collision Estimators shall find the product easy to use.	0

Fig1. Sample dataset requirements with proposed labelling

IV. RESEARCH METHODOLOGY

In the To address the research objectives (RQ1 and RQ2) the work was done in four phases to perform the classification of Functional Requirements and Non-Functional Requirements.

The first step is normalization, where the data is ready to be cleaned all unwanted words and a NULL value is removed. The second step is vectorization, where the data is converted into the format 0 & 1. The third step is classification, where we applied the algorithm, train & predict the classification model of the four algorithms i.e., RF, DT, LR, and SVM. The fourth step is Evaluation, this is the final phase where the result is calculated and then compared to all ML algorithms for precision and accuracy.

The experiment setup focused on two types of features – Bag of Word (BOW) and conventional term frequency–inverse document frequency (TFIDF) (statistical) text features to address the research objectives [17]. The experiment started with step1that involved removing symbols and lowercasing letters. Then feature extraction was performed. BOW and TFIDF were the extracted features. The Scikit-learn package's Count Vectorizer and TFIDF Vectorizer, respectively, were used to extract BOW and TFIDF.

For RQ2 selected four algorithms are Random Forest(RF), Decision Trees(DT), Logistic Regression(LR), and Support Vector Machines(SVM). Firstly, we take the same sample on these algorithms and compare all mentioned algorithms. After that, we compare the results obtained by RF,DT, LR, and SVM evaluating performance measures.

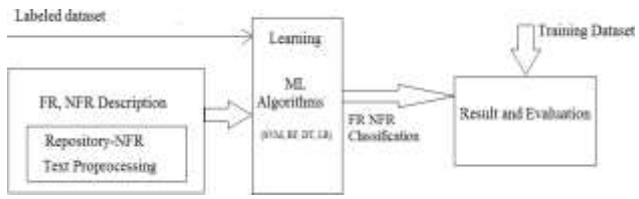

Fig2. Steps for classifying FR NFRs in textual requirements documents using selected ML algorithms.

V. Result

In this section, research questions are addressed, and findings are presented.

RQ1.What features are used to cite the approach to get better results for classifying into functional and non-functional requirements? The following Table2 shows the results for the first research question.

TABLE 2: EVALUATION OF CLASSIFIES BASED ON FEATURES

Feature and classifier	F1 score	Precision	Recall
BOW+SVM	90.30%	90.80%	89.80%
BOW+RF	84.47%	85.00%	83.80%
BOW+DT	89.30%	89.80%	88.80%
BOW+LR	90.80%	91.80%	89.80%
TF-IDF + SVM	88.53%	89.15%	87.90%
TF-IDF +RF	88.90%	90.60%	87.20%
TF-IDF +DT	90.30%	90.80%	89.80%
TF-IDF + LR	92.00%	92.90%	91.10%

The classical BOW feature appears to have the worst performance when employing LR as the classifier, scoring accuracy only 81.03% while DT is giving the best score of 89 % on the F1 scale. The low performance of BOW with LR can be explained by the fact that the requirement text for FR and NFR is relatively brief and typically just contains a single sentence. As shown in Fig. 1 and Fig. 2 TFiDF feature combined with all selected classifiers perform better over BOW on expanded dataset.

To answer the second research question (RQ2) in this study existing dataset was expanded and included 445 more requirements in the comparison study. The result with each classifier shows that SVM is most accurate over other classifiers.

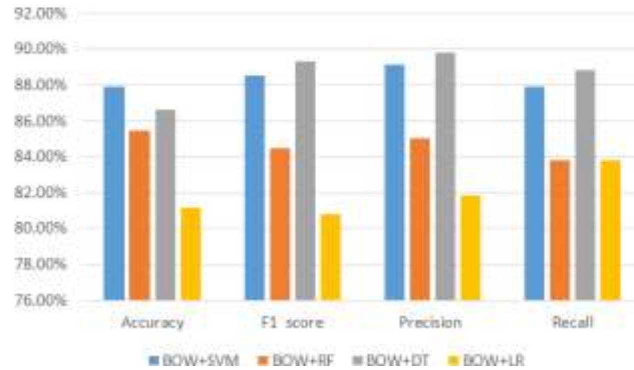

Fig 3. Evaluation of classifiers based with BOW

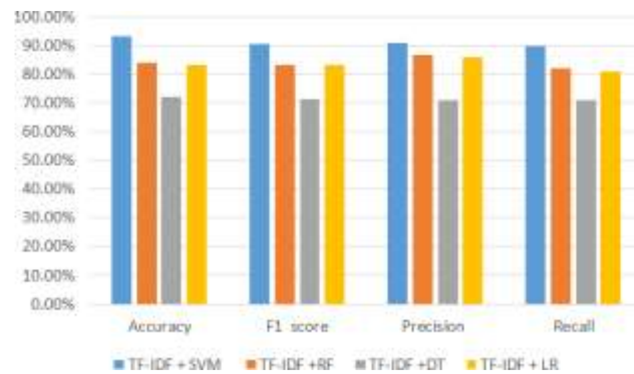

Fig 4.Evaluation of classifiers based with TF-iDF

Figures and tables must be centered in the column. Large figures and tables may span across both columns. Any table or figure that takes up more than 1 column width must be positioned either at the top or at the bottom of the page.

Graphics may be full colour. Graphics must not use stipple fill patterns because they may not be reproduced properly. Please use only *SOLID FILL*colours which contrast well both on screen and on a black-and-white hardcopy, as shown in Fig. 1.

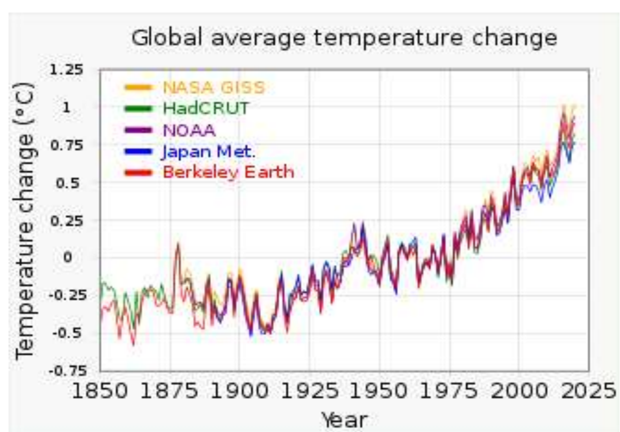

Fig. 1. A sample line graph using colours which contrast well both on screen and on a black-and-white hardcopy.

Fig. 2 shows an example of an image with adequate resolution. Check that the resolution is adequate to reveal the important detail in the figure.

Please check all figures in your paper both on screen and on a black-and-white hardcopy. When you check your paper on a black-and-white hardcopy, please ensure that:

- the colours used in each figure contrast well,
- the image used in each figure is clear,
- all text labels in each figure are legible.

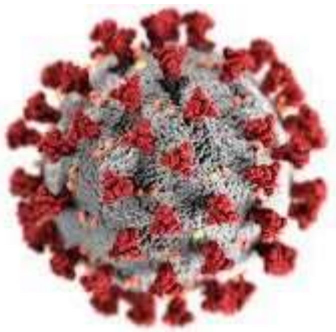

Fig. 2.Example of an image with acceptable resolution.

A. Figure Captions

Figures must be numbered using Arabic numerals. Figure captions must be in 8 pt Regular font. Captions of a single line (e.g. Fig. 2) must be centered whereas multi-line captions must be justified (e.g. Fig. 1). Captions with figure numbers must be placed after their associated figures, as shown in Fig. 1.

B. Table Captions

Tables must be numbered using uppercase Roman numerals. Table captions must be centred and in 8 pt Regular font with Small Caps. Every word in a table caption must be capitalized except for short minor words as listed in Section III-B. Captions with table numbers must be placed before their associated tables, as shown in Table I.

C. Page Numbers, Headers and Footers

Page numbers, headers and footers must not be used.

D. Links and Bookmarks

All hypertext links and section bookmarks will be removed from papers during the processing of papers for publication. If you need to refer to an Internet email address or URL in your paper, you must type out the address or URL fully in Regular font.

E. References

The heading of the References section must not be numbered. All reference items must be in 8 pt font. Please use Regular and Italic styles to distinguish different fields as shown in the References section.Number the reference items consecutively in square brackets (e.g. [1]).

When referring to a reference item, please simply use the reference number, as in [2]. Do not use "Ref. [3]" or "Reference [3]" except at the beginning of a sentence, e.g. "Reference [3] shows …". Multiple references are each numbered with separate brackets (e.g. [2], [3], [4]–[6]).

APA citation style should be followed. Examples of reference items of different categories shown in the References section include:

- example of a book in [1]
- example of a book in a series in [2]
- example of a journal article in [3]
- example of a conference paper in [4]
- example of a patent in [5]
- example of a website in [6]
- example of a web page in [7]
- example of a databook as a manual in [8]
- example of a datasheet in [9]
- example of a master's thesis in [10]
- example of a technical report in [11]
- example of a standard in [12]

VI. Conclusion

Most of the formatting instructions in this document are as per LaTeX style files and Microsoft Word. The Research Publication has used its best efforts to ensure that the templates have the same appearance.

Acknowledgment

The heading of the Acknowledgment section and the References section must not be numbered. The Research Publication wishes to acknowledge Causal Productions and other contributors for developing and maintaining the LaTeX style files which have been used in the preparation of this template.

References

[1] Metev, S. M., & Veiko, V. P. (1998) R. M. Osgood, Jr. (Ed.). *Laser assisted microtechnology* (2nd ed). Berlin, Germany: Springer-Verlag.
[2] Breckling, J. (Ed.). (1989). Applications to wind speed and direction. *ser*, The Analysis of Directional Time Series. *Lecture Notes in Statistics*. Berlin, Germany: Springer, *61*.
[3] Shengdong Zhang, Chunxiang Zhu, Sin, J. K. O., & Mok, P. K. T. (November 1999). A novel ultrathin elevated channel low-temperature poly-Si TFT. *IEEE Electron Device Letters,* 20(11), 569–571. doi:10.1109/55.798046
[4] Wegmuller, M., von der Weid, J. P., Oberson, P., & Gisin, N. (2000). High resolution fiber distributed measurements with coherent OFDR, Retrieved from paper no.: 11.3.4. In *Proceedings of the ECOC'00* p. 109.
[5] Sorace, R. E., Reinhardt, V. S., & Vaughn, S. A. High-speed digital-to-RF converter U.S Patent 5 668 842. (September 16, 1997).
[6] (2007). The IEEE website [Online]. Retrieved from http://www.ieee.org/.
[7] Shell, M. (2007). IEEEtran webpage on CTAN [Online]. Retrieved from http://www.ctan.org/tex-archive/macros/latex/contrib/IEEEtran/.
[8] *FLEXChip signal processor (MC68175/D).* (1996). Motorola.
[9] PDCA12-70 data sheet. *Opto Speed SA, Mezzovico, Switzerland.*
[10] Karnik, A. (January 1999). "*Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP*," [MEng Thesis]. Bangalore, India: Indian Institute of Science.
[11] Padhye, J., Firoiu, V., & Towsley, D. (1999). "*A stochastic model of TCP Reno congestion avoidance and control.*" MA: University of Massachusetts – Amherst. CMPSCI Tech. Rep. 99-02.
[12] Wireless LAN medium access control (MAC) and physical layer (PHY) specification. (1997). *IEEE STD, 802,* 11.