

A Comprehensive Analysis of Many Non-Functional Requirement Prediction Techniques based on Machine Learning

Naina Handa

Dav College, Amritsar
naina_pisces17@yahoo.com

Abstract

The most critical and crucial field in computer science is SE (software engineering). NFRs (Non Functional Requirements) are very essential but are often overlooked. NFRs prioritization and prediction is required at large extent. ML (Machine Learning) models show the efficient way to predict NFRs and offer the better result outcomes as compared to NLP (Natural Language Processing). The focus of the study is to present various techniques to predict NFRs offered by several researchers. The work presented in the study focusing on classification and clustering techniques available to predict NFRs. But most of the researchers have used the Naïve Bayes algorithm. Precision and Recall have been used as their performance measure by most researchers. Scientists have ignored machine learning methods such as Ensemble and Parameter Tuning. The ultimate objective is to determine the various vulnerabilities in predictive techniques based on machine learning NFRs and to draw correct future avenues.

Keywords: Machine Learning, NFR prediction, Requirement Engineering.

Introduction

SE plays a vital part in the process of software creation. RE (Requirement engineering) methods can be categorized into several stages includes RI (Requirement Interpretation), REL (Requirement Elicitation), DM (Device Modelling), RS (Requirement Specification), RV (Requirement Validation) [1]. RE is considered as the process of evaluating the stakeholders' services offered by the program, and the constraints imposed on system. Functional and non-functional specifications may be segregated [2].

This is a high-level summary of what the software will do e.g. Users can search, or a subset of, the entire database. Technical knowledge, data processing, and calculations required to accomplish a system can be a realistic necessity. That extracts architecture of the system for use. FR (Functional requirements) considered as the key feature which customer usually expect from the system, such as developing, upgrading, and deleting bank-system accounts, etc., while NFRs are not clear requirements [3]. NFRs are real constraints on the functions of the device, e.g. Timing constraints, limitation of the production method, limitation of the output to name a few [4]. NFRs are usually inconsistent, and have several implementation issues at the time of production and mainly evaluated shortly before distribution for stakeholders [5]. NFRs are termed as utilities, which may include several features like modifiability, usability, reliability, scalability, portability, maintenance, versatility, complexity, adaptability, customizability to name but a few. NFS (Non-functional specifications) are the product specifications which are implied or planned. Those are characteristics meant to be regarded as attributes of quality. Non Functional Specifications originate from the technological system's architecture. It works out system performance characteristics. NFRs are very important consideration for determining the performance &

any software project failure. It is therefore important to give equal priority to NFRs as to FRs [6].

Table 1: Comparative Analysis of NFR & FR

FR	NFR
FRs are represented in detailed manner in system design. These are detailed requirements.	In general, NFRs informally reported, subjective in nature and sometimes contrary to one another.
FRs are specifications unique to roles.	NFRs are described as criteria. All requirements for quality are NFRs.
Testing for FRs verifies the system is performing activities as it should.	NFRs testing verify whether the standards of stakeholders are fulfilled or not.
FRs are requirements which are defined by users. Those are discussed by the involved parties themselves.	Specific technical people typically describe NFRs e.g. Software developers, Designer and Community leaders etc.

Table 1 show that the FRs are the system's actual features. These are comprehensive and easy to check specifications. While the specifications are not clear as NFRs. Actually NFRs are criteria which are arbitrary, inconsistent and difficult to check. The topic of having correct system project specifications is a troublesome one. Missing requirements of the specifications also leads to project failure. NFR selection is a separate matter from Functional Requirements. Stakeholders are generally able to tell what they need from the system, but they often have little experience or knowledge about how to get it from them. But eliciting the NFRs in a complete, reliable and incontrovertible form is very necessary, and it will certainly enable the professional developers to be included in the early phases of the system development.

In both traditional and agile software development strategies, users and developers have dedicated their greatest efforts to developing FRs. NFRs are usually darken in FRs & neglected or avoided until the end of the SDLC(system development life cycle), and also considered as secondary requirement[7].It is tough to model, build, test and update NFRs late in the process of software development and may results in low reliability with hike in maintenance costs.

NFR prediction process

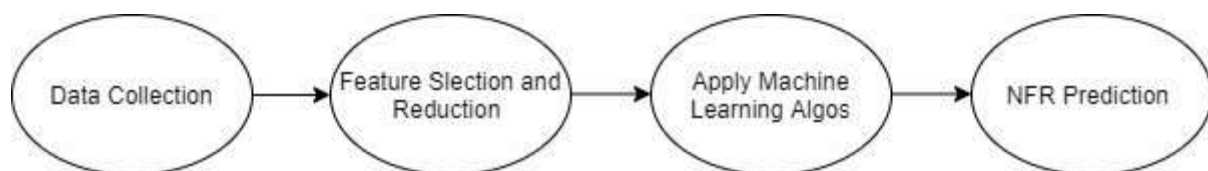


Fig 1

Data Collection: It is very foremost and primary step in Machine Learning Prediction .The data can be categorized into Primary data and Secondary data [8]. The Secondary dataset is a collection of data which is available on web and was published by someone. There are number of repositories like UCB, PROMISE and OPENDATA to name a few [9]. The data

can downloaded in various formats like excel and csv files etc. and also of different fields like tumor and government data sets to name a few. The Primary dataset has collected by the user itself for conducting the research. The secondary data can be used as a base for primary dataset [10].

Feature Selection(FS) and Reduction: It is a valuable filtering process, or selection of data set features [11]. FS algorithm provide result as weights (estimate importance of the feature) or subset of function which is selected [12]. It is an important pre-processing step which helps to overcome the problem of dimensionality. High-dimensional data can lead to increased complexities and reduced precision of the models in ML [13]. The goal of FS is to skillfully develop the prediction model and offer a superior performance.

Implementation of ML Algorithms: There are several ML algorithms like supervised[16], semi-supervised, clustering techniques or unmonitored[17] algorithms implemented on dataset [14],[15].

NFR Prediction: NFRs can be extracted from the bases of ML algorithms and predicted.

Related Work

[18] Casamayor et al . discussed the technique of text categorization based on semi-supervised methodology for classifying NFRs from a structured document using the Naïve bayes algorithm. To learn a classifier before seeking an appropriate NFR, the previously proposed supervised text categorization technique requires a lot of pre-categorization requirements, with supervised methodology requiring the researcher to manually categorize various requirements. This study tried to automate that process. The training process used in the process of classification decreases the number of criteria to be classified as compared to the supervised method.

[19] Rahimi et al extract NFR using data mining technique. The methodology proposed extracts from the document qualitative issues such as usability, efficiency and system protection. A hierarchy is being constructed in order to help the extracted NFRs model them according to the quality concerns. In the article, the sequence of machine learning and the data mining strategies are used to automatically detect various qualitative problems. A concrete hierarchy is suggested to coordinate this concern about consistency, some are linked together in such a way that certain important attributes are ignored at different stages of the hierarchy.

[20] Ramadhani et al. used a sentence-based classification algorithm for FSKNN (Fuzzy similarity based neighbor of K-nearest) recognition of NFRs. FSKNN algorithm does not take into account semantic considerations and the calculation of semantime relatedness. The system introduced in the text documents classifies various non-functional requirements. The system works on labeling the training data, classifying the data, measuring the semantic connection between the different groups and used words. The data method for automatic labeling learning saves time than for manually labeling the data. The results show the improved result with the use of Semantic Element.

Slanks and Williams [21] introduced a predetermined locator of NFR method to extract and classify sentences to 14 different categories. The proposed method distinguish several NFRs based on the classifications from accessible documents of natural language. A k-NN classifier

is proved as efficient method to differentiate related types of sentences within documents. The sentences are categorized according to different groups of NFR. It allows for the analyst to remove certain specific non-functional requirements. The paper uses multiple forms of classifiers and the result is that the classifier k-NN achieves high value in functional requirements being found.

[22] Mahmoud and Williams used a non-functional approach to classify and describe requirements which were not functional. The early methods used to classify and define NFRs utilize classification data manually to train model, classifier requires a large set of training data, but to achieve high precision, always there is unavailability of huge data. To allow traceability of NFRs, a technique is used to extract source code from the natural language content. In software specifications, semantic similarity approaches for Terms are used. The configuration of the clusters is used to construct the most logical word clusters possible.

Table 2

Reference No.	Description in form of Machine Learning Technique Used, Dataset and Validation of Model
[18]	Naïve Bayes Algorithm has used and 75% accuracy has achieved .The dataset has taken from the PROMISE Dataset and model has validated with the help of Experiment. The Security and Performance have focused in this paper.
[19]	Incremental diffusive clustering has used as Machine Learning Algorithm and the SRS document has used as input document .The Security, Performance and Usability NFRs are considered in this paper.
[20]	FSKNN has used and improved the accuracy by 44%.The 1342 Requirement sentences have used as input to the ML Model .The Performance and Access Control has focused in this research.
[21]	KNN and Naïve Bayes have used on the Health Care dataset which was taken from Promise dataset. The dataset has classified into 14 categories like Maintainability, Performance and usability to name a few.
[22]	Hierarchical clustering and Partition clustering ML techniques have used Smart Trip ,blue wallet and safe drink dataset has used and model has validated with the help of Experiment.
[23]	KNN and SMO Machine Learning algorithm have explored on Use Stories and Requirement document .Integrity, Confidentiality and Availability NFRs are focused in research.
[29]	Rule based algorithm has used on 625 requirement statements and achieve 91% accuracy. The model has validated using a case study. Promise dataset is used in this study.

Finding and Challenges of the Review

The analysis presented in the study concluded that several researchers offers classification and clustering techniques to extract and classify NFRs. Based on online reviews, SRS documents, Requirement document, User stories, Feature Request, and Web-based software, the NFRs are extracted from various software requirements for domains. Many reviewers took Pledge Server experiment results. The various NFRs are focused on various authors such as Performance, Protection, Accuracy, Health, Portability, Reliability, Legal, Availability, Privacy, Integrity, and Interoperability. But major focus of the work is on the performance

and safety prediction. It is observed in the investigation that the present methods have overlooked the use of Ensemble and Parameter tuning. There is an absence of NFRs dealing with standard datasets. Ensemble of different machine learning models was used by the majority of current researchers to achieve greater accuracy rate. Ensembling methods, however, are in essence computationally vast, and therefore incapable of achieving optimum precision. The basic dataset accessible for NFRs is missing. There is no regular NFR classification, in addition. The various authors gave different classifications.

Conclusion

This paper sets out a detailed analysis of NFR prediction techniques. It was found after literature survey that NFRd is considered as the major factor for cost effectiveness and error detection process. It may leads to problem if neglected. Requirement Engineering issue is very important for extraction and classification of NFRs. Various authors have used various machine learning algorithms to automate the extraction of NFRs. But most ignored Ensembling which was used to achieve higher speeds of precision. An effective ML model needs to be created to extract and classify NFRs in an efficient manner. Standard dataset still runs a shortage.

References

- [1]. Li, Yang, et al.: Automated requirements extraction for scientific software. *Procedia Computer Science* 51, 582-591 (2015).
- [2]. Alam, Sehrish, S. Asim Ali Shah, Shahid Nazir Bhatti, and Amr Mohsen Jadi: Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review. (2017).
- [3]. Davis, Alan M., and Dean A. Leffingwell.: Using requirements management to speed delivery of higher quality applications. Rational Software Corporation 20, 2004(1996).
- [4]. Martens, Nick: The impact of non-functional requirements on project success. Utrecht University, Msc Thesis, Utrecht(2011).
- [5]. Babar, Muhammad Imran, Masitah Ghazali, and Dayang NA Jawawi: Systematic reviews in requirements engineering: A systematic review. In 2014 8th. Malaysian Software Engineering Conference (MySEC), 43-48(2014).
- [6]. Abad, Zahra Shakeri Hossein, Oliver Karras, Parisa Ghazi, Martin Glinz, Guenther Ruhe, and Kurt Schneider: What works better? a study of classifying requirements. In 2017 IEEE 25th International Requirements Engineering Conference (RE), 496-501(2017).
- [7]. Khan, F., Jan, S. R., Tahir, M., Khan, S., & Ullah, F.: Survey: dealing non-functional requirements at architecture level. *VFAST Transactions on Software Engineering*, 9(2), 7-13(2016).
- [8]. Ezami, S.: Extracting non-functional requirements from unstructured text (Master's thesis, University of Waterloo) (2018).
- [9]. Kotsiantis, S. B., Zaharakis, I., & Pintelas, P.: Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160, 3-24(2007).
- [10]. Kiran, H. M., & Ali, Z.: Requirement Elicitation Techniques for Open Source Systems: A Review. *International Journal of Advanced Computer Science and Applications*, Pakistan, 330-334 (2018).

- [11]. Tiwari, Saurabh, and Santosh Singh Rathore.: A Methodology for the Selection of Requirement Elicitation Techniques. arXiv preprint arXiv:1709.08481 (2017).
- [12]. Asadi, M., Soltani, S., Gasevic, D., Hatala, M., & Bagheri, E.: Toward automated feature model configuration with optimizing non-functional requirements. *Information and Software Technology*, 56(9), 1144-1165 (2014).
- [13]. Groen, Eduard C., Sylwia Kopczyńska, Marc P. Hauer, Tobias D. Krafft, and Joerg Doerr.: Users—The hidden software product quality experts?: A study on how app users report quality aspects in online reviews. In 2017 IEEE 25th International Requirements Engineering Conference (RE), 80-89(2017).
- [14]. Pham, B. T., Bui, D. T., Prakash, I., & Dholakia, M. B.: Hybrid integration of Multilayer Perceptron Neural Networks and machine learning ensembles for landslide susceptibility assessment at Himalayan area (India) using GIS. *Catena*, 149, 52-63 (2017).
- [15]. Barzegar, R., Moghaddam, A. A., Deo, R., Fijani, E., & Tziritis, E.: Mapping groundwater contamination risk of multiple aquifers using multi-model ensemble of machine learning algorithms. *Science of the total environment*, 621, 697-712(2018).
- [16]. Kurtanović, Z., & Maalej, W.: Automatically classifying functional and non-functional requirements using supervised machine learning. In 2017 IEEE 25th International Requirements Engineering Conference (RE), 490-495(2017, September).
- [17]. Luo, M., Nie, F., Chang, X., Yang, Y., Hauptmann, A. G., & Zheng, Q.: Adaptive unsupervised feature selection with structure regularization. *IEEE transactions on neural networks and learning systems*, 29(4), 944-956 (2017).
- [18]. Casamayor, A., Godoy, D., & Campo, M.: Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. *Information and Software Technology*, 52(4), 436-445(2010).
- [19]. Rahimi, M., Mirakhorli, M., & Cleland-Huang, J.: Automated extraction and visualization of quality concerns from requirements specifications. In 2014 IEEE 22nd international requirements engineering conference (RE), 253-262(2014, August).
- [20]. Ramadhani, D. A., Rochimah, S., & Yuhana, U. L.: Classification of non-functional requirements using semantic-FSKNN based ISO/IEC 9126. *Telkomnika*, 13(4), 1456(2015).
- [21]. Slankas, J., & Williams, L.: Automated extraction of non-functional requirements in available documentation. In 2013 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE), 9-16(2013, May).
- [22]. Mahmoud, A., & Williams, G.: Detecting, classifying, and tracing non-functional software requirements. *Requirements Engineering*, 21(3), 357-381(2016).
- [23]. Riaz, M., King, J., Slankas, J., & Williams, L.: Hidden in plain sight: Automatically identifying security requirements from natural language artifacts. In 2014 IEEE 22nd International Requirements Engineering Conference (RE), 183-192(2014, August).
- [24]. Tóth, L., & Vidács, L.: Study of various classifiers for identification and classification of non-functional requirements. In *International Conference on Computational Science and Its Applications* (pp. 492-503). Springer, Cham(2018, May).
- [25]. Di Martino, B., Pascarella, J., Nacchia, S., Maisto, S. A., Iannucci, P., & Cerri, F.: Cloud Services Categories Identification from Requirements Specifications. In 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), 436-441(2018, May).
- [26]. Eyal Salman, H., Hammad, M., Seriai, A. D., & Al-Sbou, A.: Semantic Clustering of Functional Requirements Using Agglomerative Hierarchical Clustering. *Information*, 9(9), 222 (2018).

- [27]. Portugal, R. L. Q., Li, T., da Silva, L. F., Almentero, E., & do Prado Leite, J. C. S.: NFRfinder: a knowledge based strategy for mining non-functional requirements. In SBES, 102-111(2018, September).
- [28]. Bhowmik, T., & Do, A. Q.:Refinement and resolution of just-in-time requirements in open source software and a closer look into non-functional requirements. Journal of Industrial Information Integration, 14, 24-33(2019).
- [29]. Sachdeva, V., & Chung, L.: Handling non-functional requirements for big data and IOT projects in scrum. In 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence, 216-221(2017, January).