International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 12– No.45, July 2024 – www.ijais.org

Mapping Software Requirements: An Overview of Classification Strategies

Tamanna Yesmin Rashme
Jagannath University
Dhaka, Bangladesh

ABSTRACT
Effective software development relies on accurately identifying system requirements, which form the foundation for subsequent activities in the software life cycle (SLC). Proper classification of these requirements is crucial for subsequent design and implementation stages and can be achieved manually or through automated means. Machine Learning (ML) algorithms have emerged as valuable tools for addressing challenges in requirement engineering, such as identifying and prioritizing requirements. This paper examines ten studies to analyze the applied algorithms, presenting their training processes and evaluation metrics. The analysis aims to assist software engineers and researchers in selecting suitable requirement classification techniques by providing a detailed overview of various machine learning approaches and their effectiveness. Although the study references ten papers, it focuses on five key papers to demonstrate the results.

Keywords
SLC, Software Requirement Classification, Machine Learning.

1. INTRODUCTION
The software development process consists of several stages, from planning to testing and deploying the software. These stages are carried out in various ways, which are collectively known as Software Development Lifecycles (SDLC). One critical step in SDLC is the identification of software requirements, which can significantly impact the success or failure of software development [1]. Requirements serve as the building blocks for software development projects and are typically categorized as functional and nonfunctional requirements. Functional requirements describe system functionality, while non-functional requirements specify system properties and constraints. Classifying software requirements is a crucial task, as it helps in arranging text documents into categories based on attributes and properties belonging to each text [2]. Text classification is used in various domains, including news categorization and spam identification. However, even with well-defined software requirements, the automatic classification of natural language requirements into functional and non-functional requirements remains a challenge. This is due to the fact that stakeholders and requirements engineers employ various terminology and sentence structures to define the same sort of demand, which results in a high level of inconsistency in the process of requirements elicitation. Therefore, it is essential to find optimal ways to achieve automated classification. Manual classification of software requirements is time-consuming task, particularly for large projects with a vast number of requirements. Stakeholders need good requirements engineering, which means getting, studying, and writing down their needs, making sure that the recognized needs match their needs, and working toward requirements evolution [3]. The success of a project heavily depends on the accuracy of its requirements, as they establish formal agreement between the client and software provider working towards the same goal. Machine learning and natural language processing have greatly aided software engineering advancements over the years. There are three types of methods for machine learning: guided learning, unsupervised learning, and reinforcement learning. Various techniques and algorithms have been created and adapted to construct automation for the extraction and classification of requirements, using supervised and semi-supervised learning techniques. The current study aims to offer a preliminary understanding of the implementation of machine learning techniques for the automated classification of software requirements. To address the challenges involved, various natural language processing and ML techniques have been suggested in [4]. The first phase of this process involves text preparation, which is a crucial stage when using ML methods to elicit requirements [5]. Text preparation is made up of two key steps: text preprocessing and feature extraction. After this, the requirements are classified using ML techniques. To achieve this goal, the study conducts a systematic literature review that presents an overview of the latest developments in this field.

The following is the structure of this document. The relevant work is discussed in section II of this document. In Section III, the study methodology that was used to carry out the evaluation of the machine learning algorithms for needs categorization is outlined. The procedure for carrying out the systematic review and the main studies that were discovered are both discussed in Section IV of the paper. The outcomes of the investigation are detailed in section V. In the last part of this article, section VI, a summary of the findings, conclusions, and insights is presented.

2. RELATED WORK
The field of Software Engineering has demonstrated great potential for automation, with several processes in the software lifecycle now incorporating Machine Learning techniques and approaches [6].

Canedo and et. Al [7] used the PROMISE_exp dataset, an extension of the PROMISE repository, which contains labeled software requirements. The documents in the dataset were normalized and feature extraction was carried out using BoW, TF-IDF, and CHI2. The classification algorithms employed were Logistic Regression, Naive Bayes, and k-Nearest Neighbors. The usage of this particular dataset, the processes that were carried out in order to classify the data, and the comparison between BoW, TF-IDF, and CHI2 is something that has not been done in any of the other studies that have been conducted. This is what makes this study unique. For the software engineering community, the outcome will serve as a reference and help others in understanding the requirement

classification process. The study found that TF-IDF followed using LR produced better classification results with an F-measure of 0.91 in binary classification (tying with SVM in that case), 0.74 in NF classification and 0.78 in general classification.

Li and et al.[8] proposed a method where they used two algorithms A1 And A2. A1 algorithm used for both functional and non-functional requirements where A2 represents the algorithm that classifies the non-functional requirements. In this research no natural language processing methos had been used. n-gram method was used for data preprocessing and after that two popular ML algorithms Random Forest and Gradient Boosting applied on the dataset. The results of this study have empirically demonstrated that, among the machine learning algorithms investigated, the gradient boosting algorithm provides superior prediction performance in terms of accuracy when sorting non-functional requirements than the random forest algorithm.

Quba and et al.[9] proposed a method to classify software requirements automatically using machine learning, specifically the Bag of Words (BoW) technique with either SVM or KNN algorithms. The PROMISE_exp dataset was used, and the software documents were preprocessed before classification. The results showed that using BoW with SVM was more effective than using KNN, with an average Fmeasure of 0.74. The study's findings can serve as a useful resource for future research in this field. However, the authors plan to further improve their technique by incorporating other algorithms such as Logistic Regression to enhance model accuracy and precision.

Saad Shafiq and et al.[10] presented a systematic mapping study on the applications of machine learning in software engineering. The study resulted in the creation of a taxonomy called Machine Learning for Software Engineering (MLSE), which classifies the state-of-the-art machine learning techniques based on their relevance to different software engineering stages. The study analyzed 227 articles, providing valuable insights for both academics and practitioners on the potential benefits of integrating machine learning techniques into software engineering projects.

R. Jindal and et al.[11] proposed a method that utilizes text mining and machine learning techniques to extract keywords from NFR descriptions and classify them into one of nine NFR types. The study evaluated the performance of eight ML models on a set of 326 NFR descriptions from 15 different projects. The results showed that the Naïve Bayes model performed the best in predicting "maintainability" and "availability" NFR types. The authors emphasize the need to analyze and classify NFR descriptions in the initial phases of software development and conducted a cost-benefit analysis to highlight the potential advantages of using the proposed models.

V. Patel et al. [12] proposed an approach to automate the classification of software requirements. The dataset used for the study is the PROMISE_exp dataset, which contains 444 functional and 525 non-functional requirements. The methodology section describes the process of cleaning the dataset, extracting features, and classifying the data using algorithms such as KNN, SVM, DT, and NB. Text pre-processing steps include extracting text, converting it to lowercase, removing noise and stop words, tokenization, and lemmatization. Feature extraction methods used are Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF). Performance metrics such as confusion matrix, accuracy, precision, recall, and F1-score are used to evaluate the models, with NB showing high values for all metrics.

The paper [13] by J. Manuel Pérez-Verdejo et al. presents a review of machine learning applications in classifying software requirements. It identifies 13 key studies that utilize algorithms such as Naïve Bayes, Decision Trees, and Natural Language Processing techniques. The review highlights common training datasets, primarily academic databases and user reviews, and evaluates the performance metrics of these models. The authors conclude that while there's significant academic interest, practical applications remain limited.

The study [14] by Du Zhang and Jeffrey J.P. Tsai examines how machine learning (ML) techniques can be applied to a range of software engineering (SE) tasks. It discusses the characteristics and applicability of different ML algorithms, summarizes existing research, and provides guidelines for using ML in SE. The paper emphasizes the potential of ML to address complex, poorly understood, and dynamic SE problems, offering a systematic approach to integrate ML methods into software development and maintenance processes.

3. RESEARCH METHOD

The research was conducted using a research method based on the guidelines outlined by Kitchenham and Charters[15]. Consequently, the procedure was divided into three stages: (1) Preparation (2)Execution; and (3)Results, which are described in the following sections.

3.1 Preparation

In this section, how research has been prepared for execution will be discussed.

3.1.1 Research Question:

The RQs that will be answered the following questions that inspired the research:

- RQ1: What are the existing techniques used for the automatic requirement classification?
- RQ2: What are the steps followed by ML approaches for identification and classification of FR and NFR?
- RQ3: what are the criteria used in comparing requirement classification techniques?
- RQ4: How do the performance and applicability of machine learning techniques for requirement classification vary across different domains of software projects?
- RQ5: What are the challenges and limitations of using machine learning techniques for software requirement classification?

3.1.2 Dataset

All the papers cited here used the programming language Python and the database PROMISE_exp to conduct the experiment [16]. The UCI Machine Learning Repository served as inspiration for this repository, which is meant to encourage reproducible, verifiable, disputable, and/or augmentable prediction models in software engineering. This repository is an extended version of the original PROMISE dataset. A total of 255 FRs and 370 NFRs, further categorized

into 11 subcategories, were already tagged and stored in the original repository.

3.1.3 Source Selection

There have been a lot of automatic searches done to find and pick the best study papers linked to the RQs listed.
International Requirements Engineering Conference (RE)

- Google scholar
- IEEE

3.1.4 Search Strategy

The search for studies was carried out in three academic databases mentioned above. Additionally, the search process was carried out guided by the three following search strings:

- "Requirements Engineering" AND "Machine Learning"
- software requirement classification using machine learning.
- ("Machine Learning" OR "Machine Learning Model") AND "Software Requirements Classification"

The first has to do with finding uses in Requirements Engineering that are linked to machine learning, so that a more general method can be found for these solutions. The second search string is set up because the analysis phase is thought to come right after the classification phase [1] Lastly, the third search string is meant to directly find papers that meet standards.

3.1.5 Advance Search

To limit the number of search results for the review, the initial phase involved using the search engines of academic databases and applying specific inclusion and exclusion criteria. Only papers that meet the following criteria were collected: they must have been published between 2010 and 2023, be accessible, either a journal or conference paper, and have the search terms included in either the title or abstract.

4. METHODOLOGY

This Research has been conducted by following stages.

4.1 Analysis of Current Preprocessing Methods

The branch of Machine Learning known as Natural Language Processing (NLP) focuses on teaching computers to interpret vast amounts of data expressed in natural language, allowing for better understanding of human speech. Through text preprocessing and feature extraction, raw text can be refined and fed into ML algorithms, making it easier to train models using supervised learning techniques. The process most the paper followed for requirement classification is shown in Fig 1.

Most prior research on this topic agrees that the initial step in data cleaning is normalization, which involves removing irrelevant words. Next, in the Vectorization stage, the corpus of requirements (which has been normalized) is transformed into mathematical vectors that better represent the data within them. These vectors are then used in the Organization stage, where the SVM and kNN algorithms are employed to train and predict the categorization models. Finally, in the Evaluation stage, the predicted categories (labels) of the requirements are compared to their true labels to assess the performance of the categorization process, Not All paper use feature selection process, but follow the same procedure for classifying the FR and NFR.

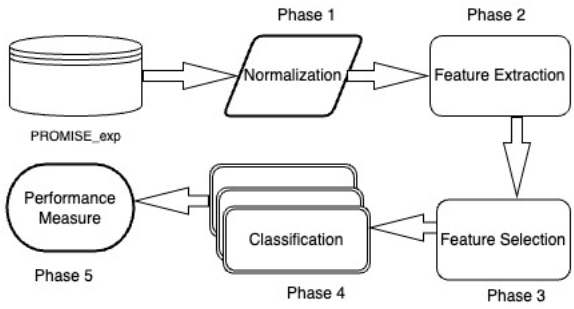

Fig 1. Method used for classification purpose. [7,9]

4.2. Execution

RQ1: What are the existing techniques used for the automatic requirement classification? Different machine algorithms have been used by different researcher for classifying the software requirements that already mentioned in section II.

4.2.1 Study of existing ML & NLP method

Artificial intelligence (AI), and more specifically machine learning (ML), have seen significant growth in the context of data analysis and computing now-a-days. This growth often enables applications to perform in an intelligent way [17]. Therefore, different kinds of machine learning approaches may play a vital part in building successful models in different application areas according to their acquisition abilities, based on the nature of the data stated before and the desired output.

- KNN: K-Nearest Neighbors (KNN) is a type of nonparametric machine learning algorithm used for classification and regression analysis. KNN works by finding the K closest data points to a new input and then assigning a class or value based on most of the K neighbors. KNN is simple and effective for small datasets but can be computationally expensive for large datasets. It is widely used in recommendation systems, image recognition, and anomaly detection [18].
- SVM: Support Vector Machines (SVMs) are a type of machine learning algorithm used for classification and regression analysis. SVMs work by finding a hyperplane that best separates the data into different classes. This hyperplane is chosen to maximize the margin, or the distance between the hyperplane and the closest data points from each class. SVMs are effective in dealing with high-dimensional data and are widely used in text classification, image classification, and bioinformatics [19].
- NB: Naive Bayes (NB) is a probabilistic machine learning algorithm used for classification. NB works by assuming that the features in the data are conditionally independent given the class and using Bayes' theorem to calculate the probability of each class given the features. Despite the naive assumption of independence, NB is often surprisingly effective in practice, especially for text classification and spam filtering. It is fast and scalable and can handle high-dimensional data[20].
- LR: Logistic Regression (LR) is a statistical machine learning algorithm used for binary classification or multiclass classification problems. LR works by modeling the relationship between the input features

and the probability of a particular class using a logistic function. The parameters of the logistic function are estimated using maximum likelihood estimation. LR is widely used in fields such as medical diagnosis, credit scoring, and fraud detection[20].

- RF: Random Forest (RF) is an ensemble machine learning algorithm used for classification, regression, and anomaly detection. RF works by constructing multiple decision trees and then aggregating their outputs to produce a final prediction. Each decision tree in the forest is constructed using a random subset of the features and a random subset of the data. RF is often used for high-dimensional data and can handle noisy and missing data. It is widely used in fields such as bioinformatics, finance, and marketing[21].
- *GB:* Gradient Boosting (GB) is an ensemble machine learning algorithm used for regression and classification problems. GB works by constructing an additive model of weak learners, typically decision trees, and then optimizing the model using a gradient descent algorithm. The weak learners are trained on the residuals of the previous learners to improve the overall performance of the model. GB is a powerful and flexible algorithm that can handle a wide range of data types and is widely used in fields such as web search, recommendation systems, and image recognition.
- *BoW:* Bag of Words is a technique used for feature extraction in natural language processing. BoW works by representing text as a bag, or a set, of its constituent words, ignoring the order of the words but keeping track of their frequency[22]. BoW is a simple and effective technique that can be used for tasks such as text classification, sentiment analysis, and topic modeling.
- TF-IDF: Term Frequency-Inverse Document Frequency is a technique used for feature weighting in natural language processing. TF-IDF works by giving more weight to words that are frequent in a document but rare in the corpus, thus highlighting words that are most characteristic of a particular document [16]. TF-IDF is widely used in text classification, information retrieval, and search engines[23].
- Chi2: Chi-Squared is a statistical test used for feature selection in machine learning. chi2 works by measuring the dependence between a feature and the class variable, and selecting the features that are most independent of the class variable. chi2 is a useful technique for reducing the dimensionality of the data and improving the performance of the model [13].
- N-gram: N-gram is a technique used for feature extraction in natural language processing. N-gram works by representing text as a sequence of n consecutive words or characters, rather than individual words [17]. N-gram can capture more context and information about the relationships between words but may also increase the dimensionality of the data and require more computational resources. N-gram is widely used in tasks such as text classification, language modeling, and machine translation.

RQ2: What are the steps followed by ML approaches for identification and classification of FR and NFR? To identify the FR and NFR three cases have been determined. First case is the performance in requirement classification for FR and NFR. Second cases are classified the NFR into segments like availability, reliability, scalability, and security which can aid in prioritizing and addressing particular performance factors. And third case is for, to better identify and solve the system's unique requirements, it is possible to further subcategorize both FRs and NFRs. Steps has been described in fig 1.

RQ3: What are the criteria used in comparing requirement classification techniques? Precision, Recall, and F-Score or F1 are the three metrics that have been agreed upon as the standard by which machine learning models should be evaluated. These measurements are used in model testing, where the expected and actual outcomes are compared.

Precision: Precision measures the accuracy of positive predictions made by a model.

$$Precision = \frac{\text{True Positives}}{True\ Positiives + False\ Positives}$$

Recall: To determine recall, divide the number of positive results by the total number of results (positive and negative).

$$Recall = \frac{\text{True Positives}}{True\ Positiives + False\ negatives}$$

F Score: The F1 Score is a balanced indicator of both Precision and Recall since it is the harmonic mean of these two measurements. A greater value indicates superior performance, and the scale runs from 0 to 1.

$$\text{F score} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

Accuracy: Accuracy is a fundamental performance metric used in machine learning to assess the effectiveness of a classification model. It measures the proportion of correctly predicted instances (both positive and negative) out of the total instances in the dataset. Accuracy is a simple and intuitive metric that is particularly useful when the classes are balanced.

$$Accuracy = \frac{True\ Positives\ (TP) + True\ Negatives\ (TN)}{True\ Positives(TP) + True\ Negatives(TN) + False\ Positives(FP) + False\ Negatives(FN)}$$

Where:

- True Positives (TP): The number of positive instances correctly predicted by the model.
- True Negatives (TN): The number of negative instances correctly predicted by the model.
- False Positives (FP): The number of negative instances incorrectly predicted as positive.
- False Negatives (FN): The number of positive instances incorrectly predicted as negative.

Even though the metrics are computed in each of the recognized studies, with the exception of the poll that was carried out in the manner in which they are compared to the primary database varies. In certain studies, a procedure known as cross-validation is carried out. This involves extracting several subsamples, also known as folds, from the dataset to divide it into training and test sets. The metrics that are generated therefore are the average metrics for each fold. However, the split between the training set and the test set may

be done once, and before the actual training process is carried out. In some instances, a subsample from the original database can be extracted, and in other instances, a comparison can be made to a separate source [24].

RQ4: How do the performance and applicability of machine learning techniques for requirement *classification vary across different domains of software projects?* This question aims to explore how the effectiveness of machine learning techniques for requirement classification can differ depending on the specific domain of the software project. It investigates the domain-specific challenges and adaptations required to optimize the classification accuracy and relevance of the ML models.

RQ5: What are the challenges and limitations of using machine learning techniques for software requirement classification? This question examines the various challenges and limitations encountered when applying machine learning techniques to software requirement classification. It focuses on issues such as data variability, inconsistency in terminology, availability of labeled data, domain-specific adaptations, and the adequacy of evaluation metrics.

5. RESULT

The primary goal of this study was to conduct a comprehensive literature research to classify the software requirement using the machine learning strategy in software development. This was accomplished using a technique of methodical evaluation proposed for in the field of software engineering. Different paper proposed their own thought to identify the applicable requirement classification technique. Table 1 shows the summary of the findings.

6. CONCLUSION

This paper represents overview of algorithms used for the classification of software requirements. The main involvement of this paper is to identify the techniques which were used for the software requirement classification. Some paper used feature selection method after preprocessing the dataset that made easier for machine learning method to give the output on timing.

Based on the review, it is seen that to identify the software requirements is difficult task because it needs text processing which is difficult and important for further procedure. Correct and proper techniques are needed for process the text which include lower case of letter, tokenization, lemmatization, noise remove, white space remove and extract the features.

Table 1. Result Analysis with their respective papers

paper	***ML Methods***	***NLP***	***Feature Selection***	***Result***
[7]	SVM, NB, KNN,LR	BoW, TF-IDF	Chi2	TF-IDF with LR high F_score
[8]	RF, GB	n-gram		GB for A1, RF for A2
[9]	SVM, KNN	BoW		BoW with SVM shows high Fmeasure
[10]	J48, RF, LB, AB, MLP, RBF, Bagging and NB	BoW	IG	NB with high score of AUC and ROC
[11]	KNN, SVM, DT, NB	BoW, TF-IDF		MNB showed high Accuracy, Precision, Recall , F1-score

A comprehensive search using the internet's four digital libraries was carried out to choose peer-reviewed publications that were published in either journals or conferences. These papers published between 2010 and 2023 were chosen, and five primary questions were derived from them.

Additionally, during research time it is found that some researchers had developed their very own NLP tools to carry out the need's categorization. On the other hand, other researchers made advantage of the built-in tools available to them, such as Weka. The quantity of research articles that are now accessible in relation to this study subject is still insufficient and calls for more contributions. Future work will involve applying and comparing different ML algorithms and ensemble methods to improve classification accuracy further.

7. REFERENCES

[1] Ashraf Abdulmunim Abdulmajeed, Younis S. Younis, "Supporting Classification of Software Requirements system Using Intelligent Technologies Algorithms," Technium Vol. 3, Issue 11 pp.32-39 (2021) ISSN: 2668-778X.

[2] J. Manuel Perez-Verdejo, Angel J. S anchez-Garcıa, "A Systematic Literature Review on Machine Learning for Automated Requirements Classification", 2020 8th International Conference in Software Engineering Research and Innovation (CONISOFT).

[3] Abualhaija, S., Arora, C., Sabetzadeh, M. et al. Automated demarcation of requirements in textual specifications: a machine learning-based approach. Empir Software Eng 25, 5454–5497 (2020). https://doi.org/10.1007/s10664-020-09864-1K.

[4] Du Zhang, Jeffrey, "Machine Learning and Software Engineering," Software Quality Journal, vol. 11, no. 2, pp. 87–119, 2003. [Online]. Available: https://doi.org/10.1023/A:1023760326768.

[5] Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN COMPUT. SCI.* **2**, 160 (2021). https://doi.org/10.1007/s42979-021-00592-x.

[6] P. Talele and R. Phalnikar, "Classification and Prioritisation of Software Requirements using Machine Learning – A Systematic Review," *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2021, pp. 912-918, doi: 10.1109/Confluence51648.2021.9377190s

[7] Dias Canedo, E.; Cordeiro Mendes, B. Software Requirements Classification Using Machine Learning Algorithms. Entropy 2020, 22, 1057. https://doi.org/10.3390/e22091057.

[8] Law Foong Li, Nicholas Chia Jin-An, Zarinah Mohd Kasirun and Chua Yan Piaw, "An Empirical Comparison of Machine Learning Algorithms for Classification of Software Requirements" International Journal of Advanced Computer Science and Applications(IJACSA),

10(11), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0101135.

[9] G. Y. Quba, H. Al Qaisi, A. Althunibat and S. AlZu'bi, "Software Requirements Classification using Machine Learning algorithm's," 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 2021, pp. 685-690, doi:10.1109/ICIT52682.2021.9491688

[10] Saad Shafiq, Atif, Christoph Mayr-Dorn, "Machine Learning for Software Engineering: A Systematic Mapping", e-Informatica Software Engineering Journal (EISEJ), Volume 15, Issue 1, 2021, pages: 85–114, DOI 10.37190/e-Inf210105.

[11] Rajni Jindal, Ruchika Malhotra, Abha Jain and Ankita Bansal, "Mining Non-Functional Requirements using Machine Learning Techniques", In e-Informatica Software Engineering Journal, vol. 15, no. 1, pp. 85– 114, 2021. DOI: 10.37190/e-Inf210105.

[12] V. Patel, P. Mehta and K. Lavingia, "Software Requirement Classification Using Machine Learning Algorithms," 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), Bangalore, India, 2023, pp. 1-6, doi: 10.1109/ICAIA57370.2023.10169588.

[13] Pérez-Verdejo, J. & Sanchez Garcia, Angel & Ocharán-Hernández, Jorge. (2020). A Systematic Literature Review on Machine Learning for Automated Requirements Classification.21-28.10.1109/CONISOFT50191.2020.00014.

[14] Zhang, D., Tsai, J.J. Machine Learning and Software Engineering. *Software Quality Journal* **11**, 87–119 (2003). https://doi.org/10.1023/A:1023760326768

[15] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Department of Computer Science University of Durham, Durham, UK, Tech. Rep., 2007.

[16] [Online]Avaible:http://promise.site.uottawa.ca/SERepository/datasets-page.html

[17] Rashme, T.Y., & Uddin, M.N. (2018). Self-Organizing Feature Map and K-Means Algorithm with Automatically Splitting and Merging Clusters based Image Segmentation. *International Journal of Image, Graphics and Signal Processing, 10*, 63-71.

[18] T. Y. Rashme, L. Islam, A. A. Prova and S. Jahan, "Autism Screening Disorder : Early Prediction," 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), Kuala Lumpur, Malaysia, 2021, pp. 1-6, doi: 10.1109/GUCON50781.2021.9573547.

[19] T. Y. Rashme, L. Islam, S. Jahan and A. A. Prova, "Early Prediction of Cardiovascular Diseases Using Feature Selection and Machine Learning Techniques," *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatre, India, 2021, pp. 1554-1559, doi: 10.1109/ICCES51350.2021.9489057.

[20] "Machine Learning and Software Engineering," Software QualityJournal, vol. 11, no. 2, pp. 87–119, 2003. [Online]. Available: https://doi.org/10.1023/A:1023760326768.

[21] Jahan, S., Islam, M.D.S., Islam, L. *et al.* Automated invasive cervical cancer disease detection at early stage through suitable machine learning model. *SN Appl. Sci.* **3**, 806 (2021). https://doi.org/10.1007/s42452-021-04786-z.

[22] Hasan, T., Matin, A. (2021). Extract Sentiment from Customer Reviews: A Better Approach of TF-IDF and BOW-Based Text Classification Using N-Gram Technique. In: Uddin, M.S., Bansal, J.C. (eds) Proceedings of International Joint Conference on Advances in Computational Intelligence. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-16-0586-4_19.

[23] Ahammad, Tanvir & Ahamed, Md. Khabir & Reshmi, Tamanna & Karim, Abdul & Halder, Sajal & Hasan, Md Mahmudul. (2021). Identification of Abusive Behavior Towards Religious Beliefs and Practices on Social Media Platforms. International Journal of Advanced Computer Science and Applications. 12. 2021. 10.14569/IJACSA.2021.0120699.

[24] Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. 2, 160 (2021). https://doi.org/10.1007/s42979-021-00592-x.

[25] Vijayvargiya, Sanidhya & Kumar, & Murthy, Lalita & Misra, Sanjay. (2022). Software Requirements Classification using Deep-learning Approach with Various Hidden Layers. 895-904. 10.15439/2022F140.

[26] Batta Mahesh, "Machine Learning Algorithms - A Review", International Journal of Science and Research (IJSR) ISSN: 2319-7064 ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426 Volume 9 Issue 1, January 2020

[27] [Online] Avaiable: https://pub.towardsai.net/difference-between-bagof-words-bow-and-tf-idf-in-nlp-with-python-97d3e75a9fd. Acces date: 5/5/2023.

[28] [Avaiable: https://www.analyticsvidhya.com/blog/2020/02/quickintroduction-bag-of-words-bow-tf-idf/, Acces date: 5/5/2023.

[29] John Violos,Konstantinos,"Text Classification Using the N-Gram Graph Representation Model Over High Frequency Data Streams," Front. Appl. Math. Stat., 11 September 2018 Sec. Mathematics of Computation and Data Science Volume 4 - 2018 | https://doi.org/10.3389/fams.2018.00041.

[30] Pratvina Talele, Rasgmi Phalnikar,"Classification and Prioritisation of Software Requirements using Machine lEarning- A Systematic Review", 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) | 978-1-6654-1451-7/20/$31.00 ©2021 IEEE |DOI:10.1109/Confluence51648.2021.9377190.