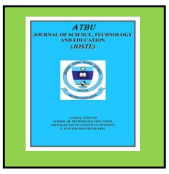
JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net



An Ensemble Learning Approach to Software Requirement Classification with Recursive Feature Elimination and Balance Bagging Classifier

Ahmed Umar, Imam Badamasi Ya'u, Fatima Umar Zambuk, Abuzairu Ahmad, Anes Aliyu Aihong.
Department of Mathematical Sciences,
Abubakar Tafawa Balewa University, Bauchi

ABSTRACT
Software development begins with the critical requirement phase, where user ideas are translated into structured requirements. Clear and accurate documentation of user requirements is fundamental for delivering a high-quality software product. However, this process involves classifying requirements into functional and non-functional categories, which can be time-consuming and error-prone when done manually. To address these challenges, this paper introduces a state-of-the-art model utilizing the Random Forest ensemble learning algorithm for the classification of software requirements. The aim is to enhance the efficiency and accuracy of this classification process. The proposed model was evaluated using precision, recall, F1-score and accuracy performance metrics. The results unveiled the model's exceptional performance, attaining a precision of 0.98, recall of 0.95, F1-score of 0.95, and an astounding accuracy of 0.99, representing a remarkable 99% performance rate. Notably, this performance outshines existing algorithms, specifically the Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), by significant margins, with differences of 0.08, 0.09, 0.09, and 0.24 in precision, recall, F1-score, and accuracy, respectively. In conclusion, the paper recommends the adoption of the Bagging Random Forest model for software requirement classification.

ARTICLE INFO
Article History
Received: June 2023
Received in revised form: July, 2023
Accepted: August, 2023
Published online: September, 2023

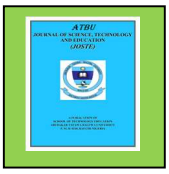
KEYWORDS
Ensemble learning, Software requirement classification, Bagging Random Forest model, SRS, Deep learning, Software development, Machine learning.

INTRODUCTION

This paper introduces the importance of the requirement phase in software development, emphasizing its user-centric nature and the need to translate user ideas into clear and well-documented requirements. It distinguishes between user needs and system requirements and mentions that these requirements can be classified into functional and non-functional categories. The creation of a software requirements specification document before starting a project is also highlighted. The initial step in the software development process is the requirement phase, which is user-focused and involves converting ideas or preferences into a requirements document (Mendes et al., 2020). It is crucial to clearly define and document the user requirements, as this is the first step in creating a high-quality product (Bandakkanavar, 2022). The requirements of a system refer to the various tasks that make up its behavior, such as responding to input values and evaluating data (Mendes et al., 2020). Needs are considered before software development begins, and common requirements are classified into functional, non-functional, and domain categories. A software requirements specification document outlines the intended purpose, requirements, nature, yield, and cost of the software to be developed, and is created before

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net



a project or application is launched (Bandakkanavar, 2022).

Statement of the Problem

Requirement engineering is a crucial first step in software development, and the industry is increasingly focused on classifying software requirements to meet the demands of larger projects. Classification helps separate user needs into functional and quality requirements based on client input, but the manual classification is time-consuming, expensive, and can lead to inaccurate results. Previous studies have attempted to automate the classification process using machine learning techniques, but improvements are still necessary for optimal performance (Mendes et al., 2020). The prominent dataset used in the state-of-the-art technique was PRMOISE_EXP, which was considered unbalanced (Mendes et al., 2020). Machine learning tools are considered essential for the automation of industrial systems, and ensemble learning techniques, such as the Random Forest algorithm, are a promising approach to software requirement classification (More, 2021). Therefore, we propose building a state-of-the-art model using the Random Forest ensemble learning algorithm to optimize software requirement classification in Software Requirements Classification Using Machine Learning Algorithm.

Aim and Objectives of the Study

The study is aim at to develop a Random Forest Ensembled Classifier for Software requirements using a Recursive Feature Selection technique. And the specific objectives are to:

i. Build an Advanced Software Requirement Classification using Ensembled Random Forest and Recursive Feature Elimination Techniques
ii. Evaluate the model against the existing model using the precision, recall, F1-score, and accuracy performance metrics.

LITERATURE REVIEW

This section forms the major contribution to reviewing the state-of-the-art Literature on the existing models used in System requirement classification in system engineering. Before the review of the general concept of System Requirement classification, different techniques used in the requirement classification, and the machine learning techniques used are all discussed in this section. The tools and techniques and the performance evaluation metrics are discussed also.

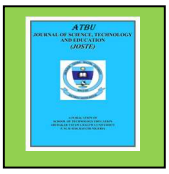
The three basic software requirements can be discussed as follows;

Functional Requirement (FR): Functional requirements are a crucial aspect of software development, as they outline the specific functions and behaviors that a system should perform to meet the users needs (Zubcoff et al., 2019). These requirements focus on defining the inputs, processes, and outputs required for specific functionality within the system. Functional requirements are typically expressed in natural language, allowing stakeholders to understand and communicate the desired system behavior effectively. Additionally, structured or formal specification languages, such as Unified Modeling Language (UML) or Z notation, can also be utilized to provide a more precise and unambiguous representation of functional requirements (Sommerville, 2016).

Non-Functional Requirement (NFR): These requirements can encompass a wide range of attributes that are important for the system's overall quality and success, such as performance, security, reliability, usability, maintainability, and scalability (Abad et al., 2018). For instance, a non-functional requirement could be that a web application must be able to handle 10,000 simultaneous users without experiencing any downtime or significant slowdowns. Unlike functional requirements, which are usually stated in natural language, non-functional requirements often require more precise measurement and testing to determine whether they have been met or not. Non-

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net



functional requirements are critical to the overall success of the system and must be carefully considered and documented during the software development process. They primarily address matters like:
(a)Portability (b) Security (c) Maintainability (d) Reliability (e) Scalability (f) Performance (g) Reusability (h) Flexibility

Domain Requirement: Domain requirements are the specific requirements that must be met in order for the software to be effective in that domain (Sommerville, 2016). Domain requirements are specific to a particular domain or industry, such as healthcare or finance (Zubcoff , et, al., 2019). They are a set of requirements that apply to all applications within that domain, and they are often derived from domain knowledge or best practices. For example, a domain requirement for a healthcare application might be that it must comply with HIPAA regulations for protecting patient privacy. These requirements may be functional or nonfunctional, and they help ensure that all applications within the domain meet the same standards and requirements.

Importance of Software Requirement Classification

The SDLC is an essential process for producing high-quality software that satisfies client needs. Prioritization is crucial to the SDLC standards, helping to focus on the requirements that are most important in terms of significance, cost, penalty, time, and risk. Stakeholders, including users and developers, define the requirements, and the requirement engineering process involves classifying and prioritizing these requirements. It is essential to consider both functional and non-functional requirements when developing software, as they both contribute to the overall value of the system (Chung & Prado, 2009). The quality of the software requirements specification (SRS) document is critical to the success of a software project, as it is used as input during the design, coding, and testing phases. Additionally, user feedback is vital to identifying new or modified requirements for upcoming releases, and the functionality and non-functional qualities of a software system ultimately define its value (Ormandjieva et, al., 2007).

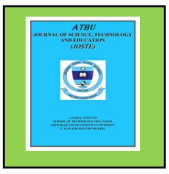
An Ensembled Learning Classifier

Ensemble approaches in machine learning involve combining insights from multiple learning models to obtain more accurate and reliable conclusions (More, 2021). Two types of ensemble learning are simple and advanced, with simple techniques including Max Voting, Averaging, and Weighted Average, and Advanced techniques including Stacking, Blending, Bagging, and Boosting. Various algorithms used in ensemble learning include Bagging-meta-Estimator, Random Forest Classifier, Adaboost, GBM, XGB, Light GBM, and CatBoost (Singh, 2018). In the proposed technique, bagging ensemble learning with Random Forest Classifier is used to enhance the classification of software requirements. The Bagging technique involves using homogeneous classifiers to improve their performance. The objective is to randomly generate training dataset replacement samples (subsets of the training data), and then train decision trees or other models using the subsets. This results in multiple models being combined, which reduces variance because the average prediction produced by several models is more reliable and robust than a single model or decision tree.

Support Vector Machine (SVM) is a widely used supervised learning algorithm for classification and regression problems in machine learning. Its main application is in classification problems. The primary objective of the SVM algorithm is to create the best decision boundary or hyperplane that can divide n-dimensional space into classes to classify future data points. The hyperplane is the best decision boundary that can segregate data points into two or more categories. In the SVM classifier, a linear hyperplane is usually

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net



used between two classes. However, the kernel trick is a technique that transforms non-separable problems into separable ones, making it useful in non-linear data separation cases. The SVM kernel is a function that raises the dimension of the input space to make it separable. It performs complex data transformations to determine how to divide the data according to the specified outputs or labels. (Ray, 2017)

K Nearest Neighbor (KNN) is a Supervised Learning algorithm that is considered one of the most straightforward Machine Learning algorithms. It works by assuming the similarity between a new data point and existing data points and then assigning the new data point to the category that is most similar to the existing ones. KNN is a versatile algorithm that can be used for both classification and regression problems in Supervised Learning (Harrison, 2018).

Logistic Regression (LR) is a widely used Machine Learning algorithm that belongs to the Supervised Learning technique. It is utilized to forecast the categorical dependent variable using a set of independent variables. Compared to linear regression, a logistic regression model has a more complex cost function, described by the sigmoid function or "logistic function" rather than a linear function (Ayush, 2019).

Multinomial Naive Bayes: the Multinomial Naive Bayes algorithm is a common probabilistic learning approach utilized in Natural Language Processing (NLP). The algorithm is founded on Bayes' theorem and is utilized to predict the tag of a text, such as an email or a newspaper article. The method calculates the probabilities of each tag for a given sample and then outputs the tag with the highest probability (Vadapalli, 2022).

RELATED WORK

Many surveys were carried out on the classification of software requirements in both ML and Deep learning techniques. A recent survey was carried out by (Khayashi, et, al., 2022) which was illustrated in Table 1 explain Table 1 here before the table.

Table 1: A taxonomy of ML Techniques Used for Requirements Classification (Khayashi, et, al., 2022)

Author (s)	Objective	Algorithm	Data Set	Best algorithm
Navarro-Almanza, et, al., (2017)	FR classification	DL	PROMISE	CNN
Rahman, et, al., (2019)	NFR classification	DL	PROMISE	LSTM
Winkler & Vogelsang (2016)	Categorizing requirements as requirements and Information	DL	DOORS	CNN
Baker, et, al., 2019	Division of the software requirements as FR and NFR, and classification of NFR	DL, ML	PROMISE	CNN
Rahimi, et al., (2021)	Classification of requirements as FR and NFR. Sub classification of FR and NFR	DL, ML	PROMISE	Ensemble models
Kici, et, al., 2021	Software requirements classification	DL	DOORS, PROMISE	DistilBERT
Ivanov, et, al.,2022	Classification of SRS as requirements and non Requirements	ML	PURE, RFI	BERT

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

Author (s)	Objective	Algorithm	Data Set	Best algorithm
Malik et, al., (2021)	Classification of nominal entities in the SRS	DL	DOORS	ML-CRF
Tiun, et, al., (2020)	FR Classification	ML	Dataset of 600 FR	Ensemble models
Tiun,	Classifying requirements into FR and NFR.	ML	PROMISE	LR
Haque, et, al., (2019)	Classification of NFR	ML	PROMISE	SGD SVM
Abad, et al., 2017	Classifying requirements into FR and NFR. Classification of NFR	ML	PROMISE	BNB
Lu & Liang (2017	Classification of users' opinions as FR, NFR, and others.	ML	Whatsapp, iBook	ARU-BoW, C4.5
Lu & Liang, (2017)	Classification of security requirements	ML	PROMISE	-
Dias & Cordeiro, (2020)	Classification of requirements as FR and NFR. Sub classification of FR and NFR	ML	PROMISE	LR
Deocadez, *et, al.,* 2017	Classification of requirements as FR and NFR	ML	User reviews	
Talele & Phalnikar (2021)	Classification and prioritization of software Requirements	ML	PURE, user comments	SVM and DT
Tóth & Vidács,(2018)	NFR classification	ML	PROMISE	MNB, SVM LLR, MNB
Tóth & Vidács, (2019)	NFR classification	ML	PROMISE, Stack Overflow	SVM, MNB, and LR.
Mahmoud & Williams, (2016)	NFR classification	ML	3 software req.	NGDWiki and LSA

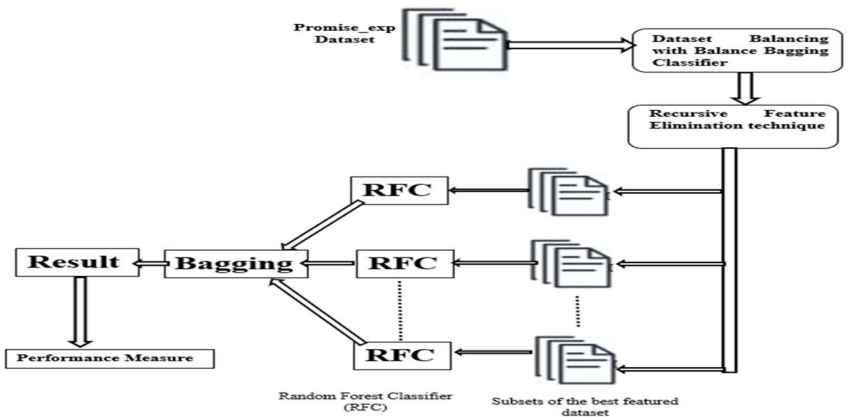
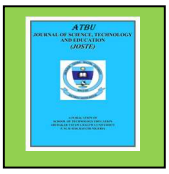


Figure 1: The model architecture of the proposed system

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net



The Proposed System Algorithm

In essence, the proposed system involves four main steps. The first step entails checking and balancing the dataset to ensure that it is suitable for the analysis. The second step involves feature selection, where the Recursive feature elimination technique is employed to select the most relevant features while eliminating those that are less important. Once the best features have been selected, the Bagging Ensembled learning technique is used in conjunction with the Random Forest Classifier model to classify the content of the dataset. The performance of the proposed model is then evaluated and compared to that of the existing system based on the results of the classification.

The general system Algorithm is as follows:

Step 1: Start.
Step 2: Let D = {d1, d2, d3, ...dn} be given dataset
Step 3: E = {}, the set of ensemble classifiers.
Step 4: S = {RFC1, RFC 2, RFC 3, ... RFC n}, the set of classifiers
Step 4: X = the training set, X D
Step 5: Y = the test set, Y D
Step 6: L = n(D)
Step 7: for i = 1 to L do
Step 8: S(i) = {Bootstrap sample I with replacement} I X
Step 9: M(i) = Model trained using C(i) on S(i)
Step 10: E = E C(i)
Step 11: Next i
Step 12: for i = 1 to L
Step 13: R(i) = Y Classified by E(i)
Step 14: Next i
Step 15: Result = max(R(i): i = 1, 2, ...n)
Step 16: Performance measure
Step 17: Stop

This algorithm encompasses the whole four phases of the proposed model development; hence it stands to be the general model algorithm.

The Model Classifier

The system development employs the Random Forest Classifier (RFC) as the classification method. RFC is a type of classifier that consists of multiple decision trees on different subsets of the dataset and then takes the average of their predictions to enhance the accuracy of the model. The implementation of RFC in the system development includes the following steps:

Step-1: Select random K data points from the training set.
Step-2: Build the decision trees associated with the selected data points (Subsets).
Step-3: Choose the number N for the decision trees that you want to build.
Step-4: Repeat Steps 1 & 2.
Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The Recursive Feature Elimination Technique

Step 1: Train a machine-learning model
Step 2: Derive feature importance
Step 3: Remove the least important feature(s)

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

Step 4: Re-train the machine learning model on the remaining features
Step 5: Calculate the change in performance of the second model concerning the previous one.
Step 6: If the performance decreases beyond a certain threshold, keep the feature from 3.
Step 7: Repeat steps 3-6 until all the features have been evaluated.

The Flowchart of the Proposed Model

Explain the flowchart steps here.

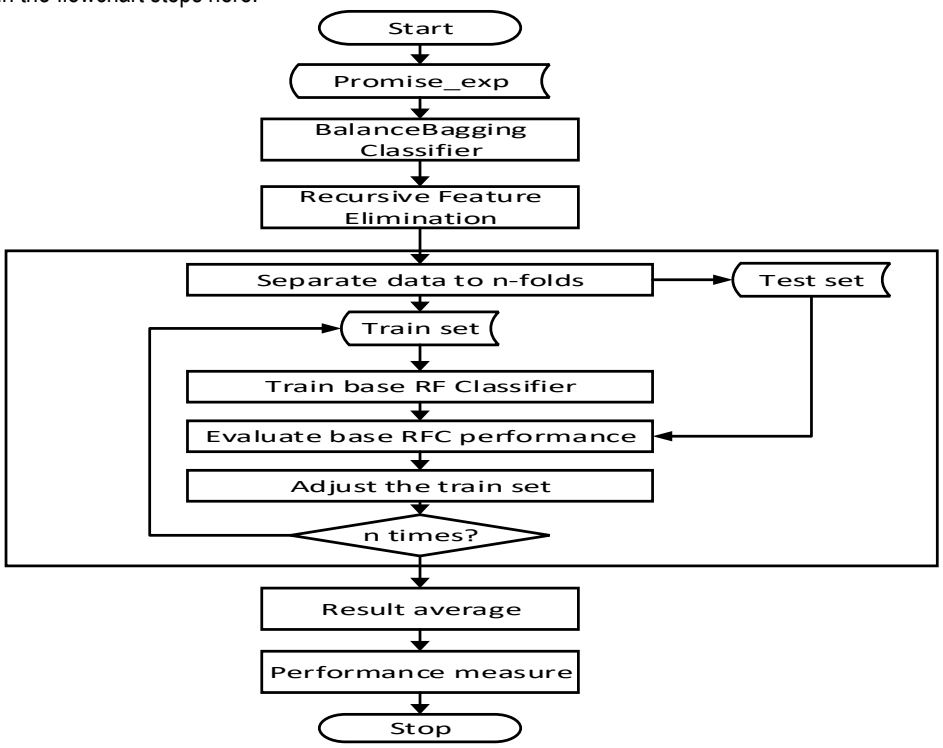


Figure 2: the flowchart of the proposed model

Dataset Description

Promise_exp dataset is made up of 625 labeled natural language needs in the original open source tera-PROMISE repository, 255 of which are functioning and 370 of which are not. The corpus was created by compiling the specifications from fifteen different projects. Available, legal, look-and-feel, maintainability, operability, performance, scalability, security, usability, fault tolerance, and portability are just a few of the non-functional needs that are covered

Figure 3 shows the number of each requirement in the Promise_exp dataset. The variations in the number flow from the highest to the lowest. It was shown that Functional Requirement cumulatively presented is the highest in the List with 444 count. While the lowest of all the requirements is Portability that is represented with only 12 count.

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

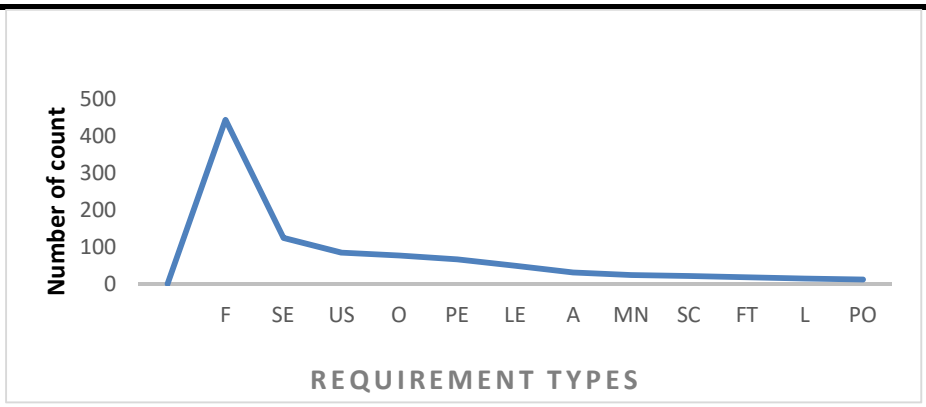


Figure 3: Requirement Types in the Promise_exp Dataset

Proposed System Model

The model as proposed used Random Forest Algorithm. A Random Forest Classifier creates an instance of Classification class from an ensemble sci-kit learn, that is used for the classification task. The number of decision trees used in the classification is indicated by the n_estimator parameter as 100. It means that the model consists of 100 ensemble decision trees, The random state of 42 means that the results are the same 42 times of reproduction. The parameter helps to maintain consistency. The train X_train and y_train arguments consider the independent and dependent variables respectively used for the model training.

During the training process, the Random Forest classifier creates multiple decision trees using different subsets of the training data. Each decision tree is trained to predict the target variable (class label) based on the provided features. The randomness introduced through bootstrap sampling and random feature selection helps the trees to capture different aspects of the data. Once all the individual decision trees are trained, their predictions are aggregated to make final predictions. In classification tasks, the class with the most votes from the ensemble's trees is chosen as the final prediction for each input sample.

Proposed Model Evaluation

The proposed model was evaluated with the benchmark evaluation metrics as suggested. Those metrics are; Accuracy, Precision, Recall, and F-Score. Table 2 illustrates the performance of the system.

Table 2: Performance Results of the Proposed System.

SN	Metrics	Performance
1	Accuracy	0.99
2	Precision	0.98
3	Recall	0.99
4	F-Score	0.99

From Table 2 above, the overall accuracy of the model on the entire dataset is 0.99, which means that the model's predictions match the true labels with high accuracy. This report provides metrics for each individual class, as well as aggregated metrics .High precision indicates that the model has a low false positive rate. The precision of 0.97, means that the

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

majority of the instances (i.e. 97%) predicted were correct. High recall indicates that the model has a low false negative rate. Recall of 0.99, meaning that 99%of actual "F" instances were correctly predicted. The high F1-score is 0.99, and it's the harmonic mean of precision and recall. Figure 4 illustrates the four metrics of the model.

Though Random Forest is a powerful algorithm, there are several technical challenges we faced in attempting to achieve high performance, among the challenges is; Hyper-parameter tuning. Random Forests have several hyper-parameters that need to be tuned for optimal performance. This includes the number of trees, the maximum depth of trees, minimum samples per leaf, and more. Hyper-parameter tuning can be time-consuming and requires careful experimentation to find the best configuration. Another challenge encountered is Random Forests can be prone to overfitting, especially if the maximum depth of the trees is not controlled. We used techniques like limiting tree depth, increasing the minimum samples per leaf, and using feature subsampling which helps us to mitigate overfitting. On the side of the Dataset (i.e PROMISE_exp), we have encountered the challenge of imbalanced dataset, where some classes have significantly fewer instances than others. Random Forests can struggle with imbalanced data, leading to biased results. Employing the Recursive feature selection technique helps to address this challenge.

Those Challenges were tackled in this study using Recursive Feature Elimination (RFE). Recursive Feature Elimination (RFE) is a technique used for feature selection, which involves iteratively removing the least important features from a model's feature set. The primary goal of RFE is to improve model performance, particularly by enhancing generalization, reducing overfitting, and potentially speeding up training times. Considering those solutions, almost all the challenges will be dealt with and an optimal performance is promising.

RESULT DISCUSSION

The result of the analysis given in Table 2 and graphically represented in Figure 5 indicates the expected improvement of the model. In a machine learning analysis, when a performance metrics reach 99%, indicates the highest performance, which shows that the model is to optimality. Apart from the accuracy those supporting metrics like, Recall, Precision, F1-Score shows a wonderful performance of the model.

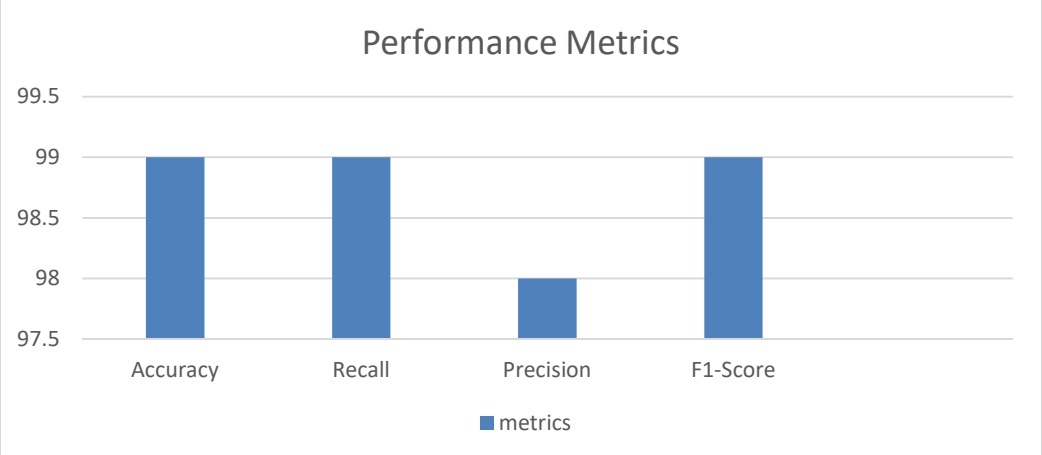


Figure 4: The performance Metrics of the Proposed System

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

Evaluation of the Proposed Model against the Existing Model

The result of the existing techniques, which uses SVM (Support Vector Machine), LR (Logistic Regression), Multinomial Naive Bayes (MNB) and KNN (K-Nearest Neighbour) algorithms to design the model, the performance of those algorithms, using same Promise_exp dataset is given in Table 3. The Existing system used only Precision, Recall and F1-Score to measure the performances of the algorithms, but the proposed system used Accuracy as well to measure the performance of the model.

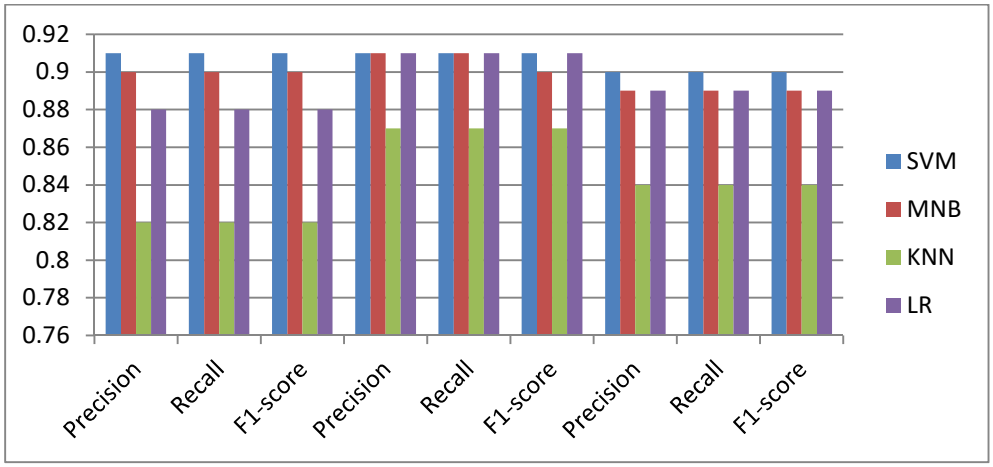


Figure 5: The performance evaluation of the existing system

Figure 5 shows the comparative analysis of the existing system with respect to precision, recall and F1-score, which where the equivalent metrics used in both the system using the prominent Promise_exp dataset.

Table 3: indicates clearly the differences between the performance metrics used in both the existing and the proposed system.

	BoW				**TF-IDF**				**CHI^2**			
	Prec	Rec	F1-sc	Accu	Prec.	Rec	F1-sc	Accu	Prec.	Rec	F1-sc	Accu
SVM	0.91	0.91	0.91	0.75	0.91	0.91	0.91	0.75	0.90	0.90	0.90	0.75
MNB	0.90	0.90	0.90	0.74	0.91	0.90	0.90	0.74	0.89	0.89	0.89	0.73
KNN	0.82	0.82	0.82	0.72	0.87	0.82	0.82	0.74	0.84	0.84	0.84	0.73
LR	0.88	0.88	0.88	0.73	0.91	0.88	0.88	0.75	0.89	0.89	0.89	0.72
Proposed Model (Bagging RF)	***0.98***	***0.99***	***0.99***	***0.99***	***0.95***	***0.99***	***0.99***	***0.99***	***0.94***	***0.95***	***0.95*** 0.95	***0.99***

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 11(4), DECEMBER, 2023
ISSN: 2277-0011; Journal homepage: www.atbuftejoste.net

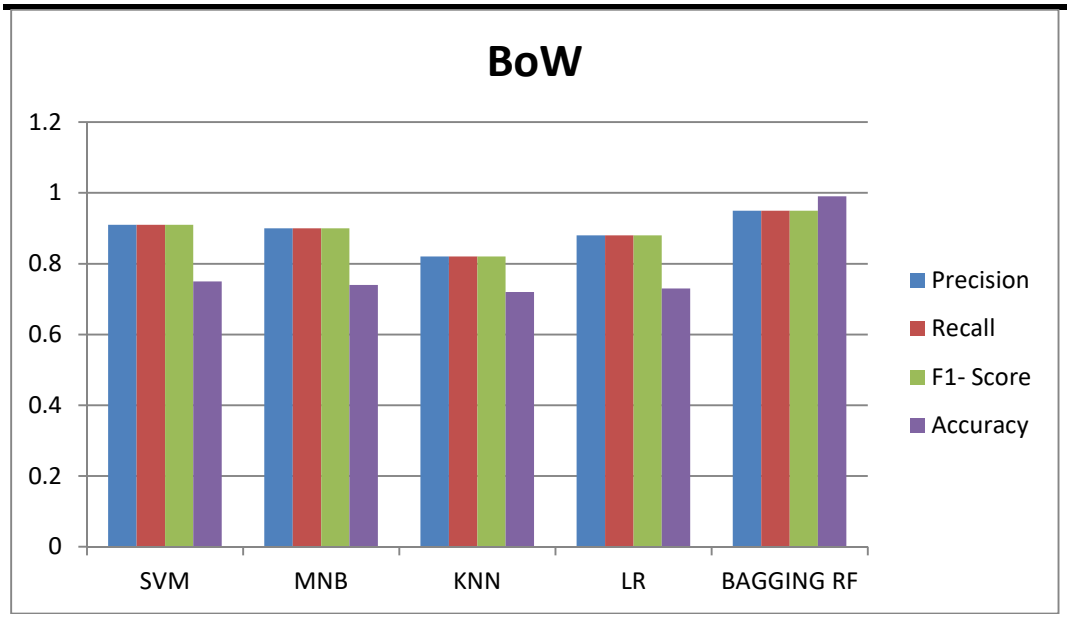


Figure 6: The Comparative Graph of the Systems Using BoW

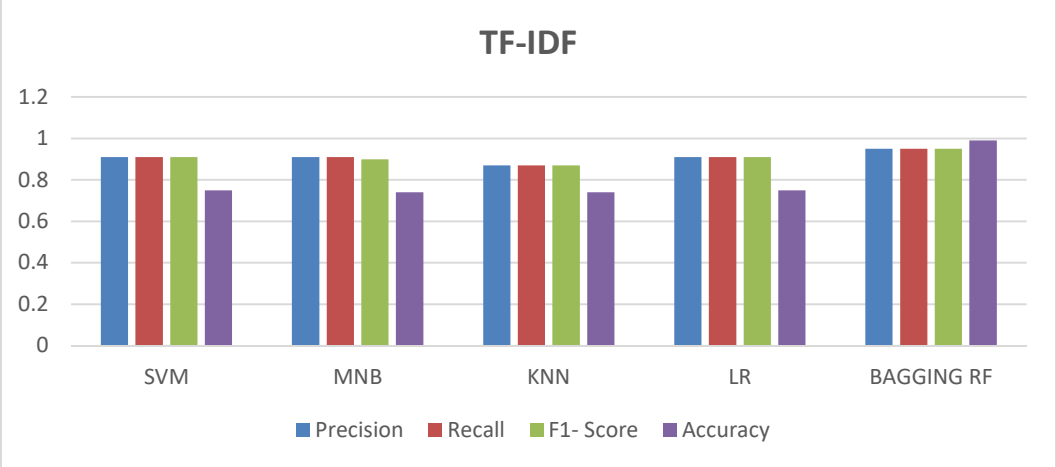


Figure 7: The Comparative Graph of the systems using TF-IDF

The Figure 7 above shows the comparative chart of the systems using TF-IDF. It indicates clearly that RF bagging classifier outperformed the SVM, MNB, KNN and LR.

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved

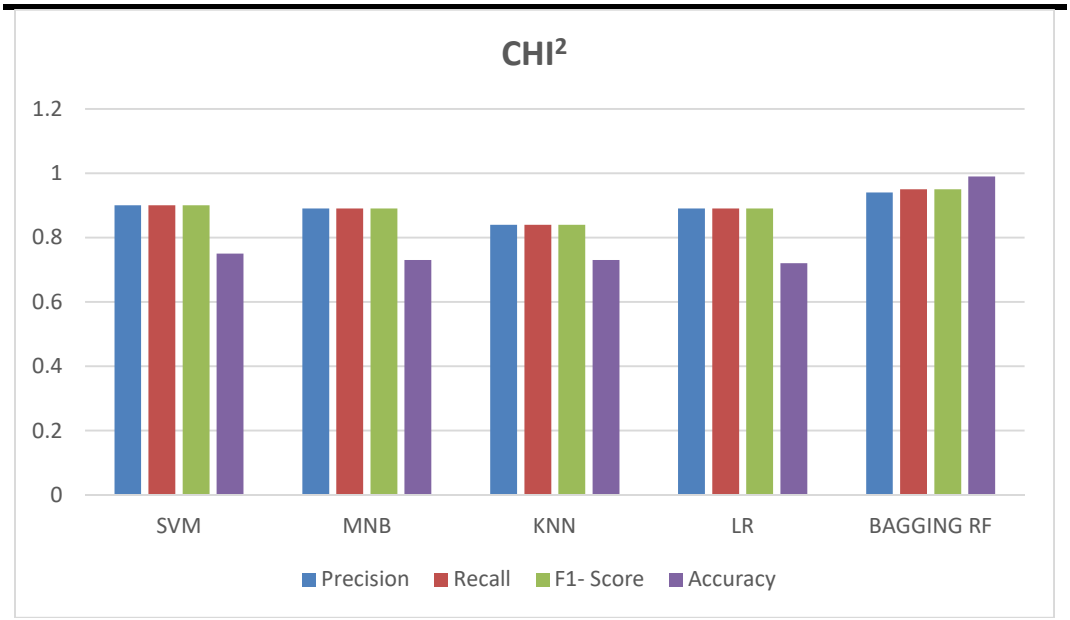


Figure 8: The Comparative Graph of the Systems Using CHI2

The figure 8 above clearly shows the comparative graph of the systems using CHI2. Again, the RF Bagging outperformed the SVM, KNN, MNB and LR.

CONCLUSION

In the conclusion, the paper presents the results of comparing the proposed Bagging Random Forest model with existing algorithms, highlighting the superior performance of the proposed model in terms of precision, recall, F1-score, and accuracy. The paper concludes by recommending the adoption of the Bagging Random Forest model for software requirement classification. The highest performed Algorithm in the existing system is SVM which has the differences of 0.08 in precision and 0.09 in both Recall and F1-score, when compared with the Proposed Model (i.e Bagging RF). This shows how Random Forest exceeds, KNN, SVM, LR and MNBin terms of Classification of Software Requirement Specification. Accuracy is one metric that was not used in the existing approach, it can be an indication for optimal performance of the model since its records up to 0.99 which is 99% rate of performance.

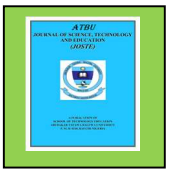
RECOMMENDATIONS

The recommendations suggest avenues for further research, including algorithm optimization, incorporation of unstructured data, real-world application and validation, human-in-the-loop approaches, long-term monitoring and adaptation, and making AI models more interpretable and explainable. These recommendations aim to advance the field of software requirement classification and contribute to more accurate and efficient solutions for software development.

REFERENCES

Abad, Z. S. H., Karras, O., Ghazi, P., Glinz, M., Ruhe, G., & Schneider, K. (2017, September). What works better? a study of classifying requirements. In *2017 IEEE 25th International Requirements Engineering Conference (RE)* (pp. 496-501). IEEE.

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved



Ayush, P. (2019, January 22). *Introduction to Logistic Regression*. Towards Data Science. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

Baker, C., Deng, L., Chakraborty, S., &Dehlinger, J. (2019, July). Automatic multi-class non-functional software requirements classification using neural networks. In *2019 IEEE 43rd annual computer software and applications conference (COMPSAC)* (Vol. 2, pp. 610-615). IEEE.

Bandakkanavar, R. (2022, January 30). *Software Requirements Specification document with example.* Krazytech. https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

Chung, L., & do Prado Leite, J. C. S. (2009). On non-functional requirements in software engineering. *Conceptual modeling: Foundations and Applications: Essays in honor of john mylopoulos*, 363-379.

Deocadez, R., Harrison, R., & Rodriguez, D. (2017, September). Automatically classifying requirements from app stores: A preliminary study. In *2017 IEEE 25th international requirements engineering conference Workshop (REW)* (pp. 367-371). IEEE.

Dias Canedo, E., & Cordeiro Mendes, B. (2020). Software requirements classification using machine learning algorithms. Entropy, 22(9), 1057.

Haque, M. A., Rahman, ,I.0;M. A., & Siddik, M. S. (2019, May). Non-functional requirements classification with feature extraction and machine learning: An empirical study. *In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (pp. 1-5).* IEEE.

Harrison, O. (2018, September 10). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Towards Data Science. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Ivanov, V., Sadovykh, A., Naumchev, A., Bagnato, A., & Yakovlev, K. (2022, August). Extracting Software Requirements from Unstructured Documents. In *Recent Trends in Analysis of Images, Social Networks and Texts: 10th International Conference, AIST 2021, Tbilisi, Georgia, December 16–18, 2021, Revised Selected Papers* (pp. 17-29). Cham: Springer International Publishing.

Khayashi, F., Jamasb, B., Akbari, R., & Shamsinejadbabaki, P. (2022). Deep Learning Methods for Software Requirement Classification: A Performance Study on the PURE dataset. arXiv preprint arXiv:2211.05286.

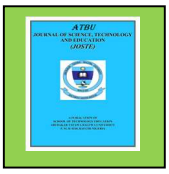
Kici, D., Malik, G., Cevik, M., Parikh, D., &Basar, A. (2021, June). A BERT-based transfer learning approach to text classification on software requirements specifications. In *Canadian Conference on AI*.

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

Mahmoud, A., & Williams, G. (2016). Detecting, classifying, and tracing non-functional software requirements. *Requirements Engineering*, *21*, 357-381.

Malik, G., Cevik, M., Khedr, Y., Parikh, D., &Basar, A. (2021). Named Entity Recognition on Software Requirements

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved



Specification Documents. At *Canadian Conference on AI*

More, V. (2021, March 26). What Is Ensemble Learning? Understanding Machine Learning Techniques. https://www.simplilearn.com/ensemble-learning-article

Navarro-Almanza, J., Alba-Castro, J. L., &Pelechano, V. (2017). Applying machine learning techniques to automatically classify requirements. Information and Software Technology, 82, 31-43.

Ormandjieva, O., Hussain, I., &Kosseim, L. (2007, September). Toward a text classification system for the quality assessment of software requirements written in natural language. In *Fourth international workshop on Software quality assurance: in conjunction with the 6th ESEC/FSE joint meeting* (pp. 39-45).

Rahimi, N., Eassa, F., &Elrefaei, L. (2021). One-and-two-phase software requirement classification using ensemble deep learning. *Entropy*, *23*(10), 1264.

Rahman, M. A., Haque, M. A., Tawhid, M. N. A., &Siddik, M. S. (2019, August). Classifying non-functional requirements using RNN variants for quality software development. In *Proceedings of the 3rd ACM SIGSOFT International Workshop on Machine Learning Techniques for Software Quality Evaluation* (pp. 25-30).

Ray, S. (2017, September 12). *SVM | How to Use Support Vector Machines (SVM) in Data Science*. How to Use Support Vector Machines (SVM) in Data S. https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson.

Talele, P., &Phalnikar, R. (2021). Software requirements classification and prioritization using machine learning. In *Machine Learning for Predictive Analysis: Proceedings of ICTIS 2020* (pp. 257-267). Springer Singapore.

Tiun, S., Mokhtar, U. A., Bakar, S. H., & Saad, S. (2020, April). Classification of functional and non-functional requirements in software requirements using Word2vec and fast Text. In the *journal of Physics: conference series* (Vol. 1529, No. 4, p. 042077). IOP Publishing.

Tóth, L., &Vidács, L. (2018). Study of various classifiers for identification and classification of non-functional requirements. In *Computational Science and Its Applications–ICCSA 2018: 18th International Conference, Melbourne, VIC, Australia, July 2-5, 2018, Proceedings, Part V 18* (pp. 492-503). Springer International Publishing.

Vadapalli, P. (2022, October 3). *Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2023*. UpGrad blog. https://www.upgrad.com/blog/multinomial-naive-bayes-explained/

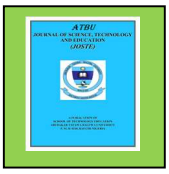
Wiegers, K. E. (2007, August 6). *Software requirements specification and IEEE standards.* TechTarget. https://www.techtarget.com/searchsoftwarequality/answer/Software-requirements-specification-and-the-IEEE-standard

Wiegers, K., & Beatty, J. (2013). *Software requirements.* Pearson Education.

Winkler, J., & Vogelsang, A. (2016, September). Automatic classification of requirements based on convolutional neural networks. In *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)* (pp. 39-45). IEEE.

Yang, Z., Chen, Y., Zhou, Y., Li, M., & Jiang, Y. (2015). On the localness of software.

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved



IEEE Transactions on Software Engineering, 41(3), 287-306.

Zubcoff, J., Garrigós, I., Casteleyn, S., Mazón, J. N., Aguilar, J. A., & Gomariz-Castillo, F. (2019). Evaluating different i*-based approaches for selecting functional requirements while balancing and optimizing non-functional requirements: A controlled experiment. *Information and Software Technology,* 106, 68-84.

Corresponding author: Ahmed Umar
ahmedumar589@gmail.com
Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi.
© 2023. Faculty of Technology Education, Abubakar Tafawa Balewa University, Bauchi. All rights reserved