Vol-2 Issue-2 2024
Scientific Research Journal of Science, Engineering and Technology
ISSN: 2584-0584, Peer Reviewed Journal

A Review on Classification of Extracted Features from Software Requirements Specification Documents using Support Vector Machine Learning Technique

Sadiq Mohammed Waziri[1], Fatima Umar Zambuk[2], Badamasi Imam Ya'u[3]

[1,2,3] Department of Computer Science, Faculty of Science, Abubakar Tafawa Balewa University Bauchi, Bauchi State, Nigeria

Correspondence: Sadiq Mohammed Waziri
Email id: swaziri.pg@atbu.edu.ng

Abstract
Manual classification of extracted features from large datasets can be tedious and time-consuming. This paper reviews the methods for classifying extracted features from SRS documents using Machine Learning (ML), with focus on linear Support Vector Machine (SVM) technique. We also explore other classification techniques, such as decision trees (DT), naïve Bayes (NB), and k-nearest neighbors (KNN)—for classifying the extracted features into mandatory and optional. Previous studies have compared different classification techniques for feature modeling. The primary goal of this review is to identify the best method for binary classification of features for software product lines engineering (SPLE). The proposed system will be tested on nine SRS documents that were chosen from the Public Requirements dataset with accuracy, precision, recall, and F1 scores used for evaluation.

Keywords – Requirements Feature, Extraction of Feature, Classification of Feature, Feature Modeling, Support Vector Machines

1 Introduction

The demand for software production has driven research into automating methods of extraction and classification of feature from different sources [1] — such as conversational agents, code, program documents, and user manuals—to enhance speed and precision while ensuring product quality [1]. Feature extraction in SPLE aims at facilitating feature modeling, where variations and similarities are identified within a Software Product Family (SPF). However, feature modeling cannot be fully realized without accurately classifying these features.

Classification in this context divides features into mandatory, representing commonalities within an SPF, and optional, identifying variations within the family, thus creating a binary classification. Previous studies have sought to determine the best technique for such classifications, with many finding SVM as a leading method. This review paper focuses on linear SVM for classifying extracted features from SRS documents, while also considering related studies. The article proceeds according to the following arrangement: Section 1 is the introductory section; Section 2 discusses methodology; Section 3 reviews related work(s); Section 4 discusses SVM technique; and, Section 5 offers a discussion. Finally, Section 6 concludes the review.

2 Methodology

We retrieved four additional documents, making nine in total, from the 79 documents in the PURE dataset to implement the existing system and classify features using the linear SVM technique. To ensure the validity of the proposed model we will employ three other classification techniques— DT, NB, and KNN—to classify features into mandatory and optional for comparison. We will evaluate the performance of these techniques using precision, accuracy, recall, and F1 scores metrics.

3 Related Works

Several studies have explored various classification techniques for different applications. For instance, [2] conducted comprehensive research on multiple classification methods, including Logistic Regression (LR), NB, DT, C4.5, C50, Artificial Neural Network (NN), KNN, XGBoost, and SVM. The study concluded that SVM and Logistic Regression were among the best-performing techniques with regards to accuracy and weighted F1-score.

In another study, [3] employed several ML techniques (SVM, KNN, DT, and NB) to classify publications into business, sciences, and social sciences categories, finding good performance across techniques except for Decision Trees. Similarly, [4] conducted a comparative study where a hybrid CNN-SVM model outperformed both individual techniques.

A study by [5] on classifying abstracts using SVM found that a linear kernel produced the highest accuracy of 58.3%. The accuracy was influenced by the number of features in the documents, where a higher number of features improved the classification outcome. This is consistent with [6], who emphasized that dataset quality significantly affects ML algorithm performance and highlighted the importance of enhancing classification algorithms.

Comparing ML techniques, [7] noted that while logic-based systems work best with discrete/categorical features, SVMs and NNs typically perform better with multidimensional and continuous features. SVMs require larger datasets for optimal prediction accuracy, unlike NB, which performs well with smaller datasets. The sensitivity of KNN to irrelevant attributes and the performance of decision trees with hyperrectangles were also discussed.
In the domain of textual classification, [9] used the C4.5 DT technique to predict the requirements' testability while [10] applied SVM for email spam filtering, finding it particularly effective in binary classification. Similarly, [11] analyzed the performance of different ML algorithms on IMDB and Spam datasets, where SVM ranked second on the IMDB dataset with an accuracy of 85.5%.

[12] analyzed public reaction to COVID-19 on social media using various ML algorithms, with SVM and LR delivering the best results. In another study, [13] compared SVM and KNN for categorizing software requirements into functional and non-functional requirements. SVM achieved an average F-measure of 0.74.

Finally, [14] evaluated sentiment classification techniques using NB, Maximum Entropy, and SVM on movie reviews, finding that ML approaches outperformed human-generated baselines, with SVM showing the highest performance among the three.

4 SVM Classification Technique

SVM is a contemporary supervised machine learning technique [13], closely related to classical multilayer perceptron NNs. SVMs operate by creating a "margin" on both sides of a line, called hyperplane, which separates two data classes. By maximizing this margin, SVMs aim to reduce predicted generalization error, thereby establishing the greatest distance between instances on either side of the hyperplane, [13] elaborated.

In SVM, the decision function for linear classification is represented as:

$$f(x) = w \cdot x + b \text{ [15]}$$

SVM is widely used in regression and classification, with Support Vector Classifier (SVC) being specialized for the latter.

The following are key concepts to provide a basic understanding of SVM:

a. Linear SVM for Binary Classification:
Decision Function: The decision function for a linear SVM is given by $f(x) = w \cdot x + b$ [15]

$f(x)$ = the decision function's output.
w = weight vector.
x = input feature vector.
b = bias term.

Prediction: To make a binary classification prediction, you typically use the $f(x)$ sign:

prediction = sign(f(x))

If $f(x)$ is greater than or equal to 0, the prediction is in one class; if it's less than 0, the prediction is in the other class.

b. Support Vector:
A support vector is any of the data points lying near the hyperplane and has non-zero weight in the SVM model [16]. Support vectors are important in defining the position and orientation of the hyperplane.

c. Margin:
Margin is the distance separating hyperplane and the support vector closest to it. In a well-trained SVM, the goal is to maximize the margin [17].

d. Kernel Trick:
SVMs handle non-linear classification, using a kernel function such as the radial basis function (RBF) and mapping input data into a higher-dimensional feature space. The kernel trick finds a hyperplane in the transformed space that corresponds to a non-linear decision boundary in the original space [17][8].

RBF Kernel: The RBF kernel is commonly used and is defined as:

$K(x_1, x_2) = e(-\gamma * ||x_1 – x_2||^2)$ [15]

Where $K(x_1, x_2)$ is the kernel function.

X_1 and x_2 are data points.

γ is a hyperparameter that controls the shape of the kernel.

e. C Parameter:
The C parameter in SVM controls the trade-off between minimizing the classification error(s) on the training data and maximizing the margin [8]. A small C value encourages a larger margin but allows some training points to be misclassified; a large C value tries to minimize misclassifications but may possibly result into a smaller margin [16].

SVMs become more complex when dealing with multi-class problems, non-linear kernels, and various parameters.

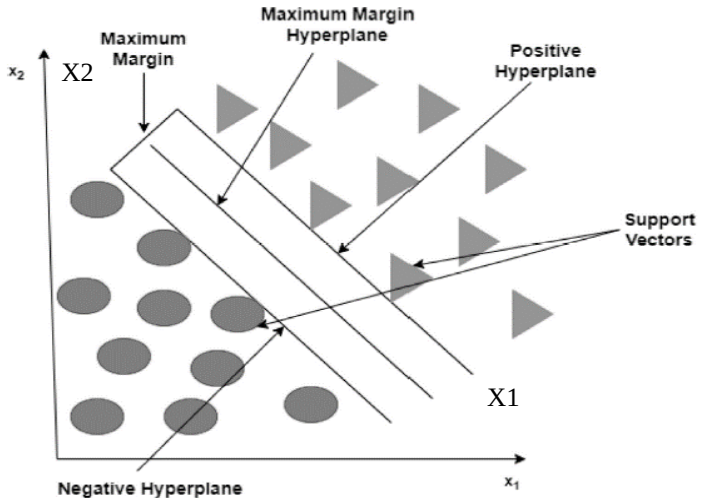

Fig. 1 SVM Classification [17new]

In classification, the steps include taking inputs and separating them into N classes they might belong according to training from each class's exemplars [17]. Classification is traditionally discrete in that exemplars belong to particular classes. All outcomes are covered by the predefined classes.

5 Discussion

The research presented in this review highlights the importance of feature classification in SPLE. Several studies have explored various machine learning techniques for this task, with SVM emerging as a promising technique. SVM has several advantages for feature classification in SPLE:

i. **Versatility:** SVM handles a linear and non-linear [18] relationship between features and classes, making it suitable for a wide range of datasets.
ii. **Efficiency:** SVM is computationally efficient [19], especially for smaller datasets [20].
iii. **Robustness:** SVM is relatively tough to noise and eccentric data [21].

However, SVM also has some limitations like sensitivity to tuning of hyperparameter. That is, SVM's performance is sometimes sensitive to the choice of hyperparameters, such as C parameter and kernel function; and scalability: SVM can be computationally expensive for enormous datasets.

While SVM has demonstrated promising results in many studies, it is important to consider specific characteristics of your dataset and the desired trade-off between accuracy and efficiency whenever you come to select a classification technique.

6 Conclusion

After implementing the proposed system, we found that SVM outperformed DT, NB, and KNN in terms of the average results shown in Table 1. This highlighted the potential of SVM as the most promising technique for feature classification in SPLE.

Table 1. Results of Performance Evaluation [22]

Model	Av. Accuracy	Av. Precision	Av. Recall	Av. F1-Score
SVM	0.86	0.89	0.83	0.86
DT	0.82	0.83	0.80	0.82
NB	0.80	0.82	0.78	0.79
KNN	0.81	0.82	0.79	0.81

"Av." means "Average".

Future research could focus on:

i. Developing more efficient SVM algorithms for large-scale datasets.
ii. Investigating the impact of different feature extraction techniques on classification performance.
iii. Exploring alternative machine learning techniques that may be better suited for specific features or datasets.

Future research can contribute positively to the development of software product lines if these areas are addressed.

7 Conflict of Interest

The authors declare that there was no conflict of interest.

8 Authors Contributions

Sadiq Mohammed Waziri conceptualized and led the research; Fatima Umar Zambuk supervised the write-ups; Badamasi Imam Ya'u is the second supervisor and has contributed immensely to the literature review.

9 Funding

There was no funding received for the research work.

10 Acknowledgements

The authors acknowledge the Computer Science Department of Abubakar Tafawa Balewa University Bauchi for providing the necessary support required for the research.

11 Data Availability

The data supporting the findings of this study is provided only upon reasonable request from the corresponding author.

12 Reference

[1] Pohl, K., Bockle, G., & van der Linden, F. (2005). Software Product Line Engineering: Foundations, Principles, and Techniques. Springer, Heidelberg, Berlin, Germany.

[2] Oncea, B. (2023). Automatic Classification using Supervised Machine Learning in Price Statistics. MDPI, Basel, Switzerland. Retrieved July 20, 2023 from https://www.mdpi.com/journal/mathematics

[3] Chowdhury, S. and Shoen, M. (2020). Research Paper Classification using Machine Learning Techniques. 10.1109/IETC47856.2020.9249211

[4] Macedo, D. (2020). Improving Image Classification Accuracy Using Hybrid Systems of SVM and CNN.

[5] Lumbanraja, F., Fitri, E. Junaidi, A. & Prabowo, R. (2022). Abstract Classification using SVM Algorithm (Case Study: Abstract in Computer Science Journal)

[6] Sukhpreet, S. & Malik, K. (2022). Feature Selection and Classification Improvement of kinnow fruits using SVM Classifier

[7] Osisanwo, F. et al (2017). Supervised Machine Learning Algorithms: Classification and Comparison. International Journal of Computer Trends and Technology (IJCTT) – Volume 48 Number 3 June 2017.

[8] Shawe-Taylor, J., & Cristianini, N. (2000). Support Vector Machines (Vol. 2). Cambridge: Cambridge University Press.

[9]. Wrinkler, J. & Vogelsang, A. (2017). Automatic Classification of Requirements Based on Convolutional Neural Networks. In Requirements Engineering Conference Workshops (REW), IEEE International. New York: IEEE. DOI: https://doi.org/10.1109/REW.2016.021.

[10] Pandey, S., Taralekar, A., Yadav, R., Deshmukh, S. & Suryavanshi, S. (2020). Email Spam Detection and Classification using SVM. Paper in Journal of Computer Science and Information Technologies, Vol. II (1)

[11] Hassan, S. et al. (2022). Analytics of ML-Based Algorithms for Text Classification. Published by Elsevier B.V. on behalf of KeAi Communications Co., Ltd.

[12] Aurnob, F. et al. (2022). Sentiment Analysis on Corona Virus Tweets. Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

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

[14] Pang, B., Lee, L. and Vaithyanathan, S. (2002). Thumbs Up: Sentiment Classification using ML Techniques. Appears in Proc. 2002 Conf. on Empirical Methods in Natural Language Processing (EMNLP).

[15] Mavroforakis, M & Theodoridis, S (2006). A Geometric Approach to Support Vector Machine(SVM) Classification. EEE Transactions on Neural Networks, Vol. 17, No. 3, May 2006. DOI: 10.1109/TNN.2006.873281

[16] Müller, A. C. and Guido, S. (2002). Introduction to Machine Learning with Python: A Guide for Data Scientists. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

[17] Simpilearn (2023). https://www.simplilearn.com/image-processing-article/

[18] https://www.techtarget.com/whatis/definition/support-vector-machine-SVM

[19] https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/#Pros_and_Cons_of_SVM

[20] https://www.geeksforgeeks.org/support-vector-machine-in-machine-learning/

[21] Awad, M., Khanna, R. (2015). Support Vector Machines for Classification. In: Efficient Learning Machines. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-5990-9_3

[22] Waziri, S. et al (2024). Classification of Extracted Features from Software Requirements Specification Documents using Support Vector Machine Learning Technique**.** https://15.207.161.74/journal/SRJSET/currentissue/classification-of-extracted-features-from-software-requirements-specification-documents-using-support-vector-machine-learning-technique