IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 February 2025, accepted 27 March 2025, date of publication 11 April 2025, date of current version 21 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3558795

RESEARCH ARTICLE

Feature Selection Techniques for Enhancing App User Review Analysis

AMRAN SALLEH, MOHD HAFEEZ OSMAN, (Member, IEEE), SA'ADAH HASSAN, AND MAR YAH SAID

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor 43400, Malaysia

Corresponding author: Mohd Hafeez Osman (hafeez@upm.edu.my)

This work was supported by Universtiti Putra Malaysia.

ABSTRACT The rapid growth of user-generated content, particularly app user reviews, presents a significant challenge in analyzing and extracting useful insights. The unstructured nature, inconsistent quality, and large volume of these reviews make it difficult to identify relevant information for app maintenance and updates. This study addresses this challenge by evaluating the impact of different feature selection techniques on the performance of machine learning models in multi-label classification tasks for app user review analysis. Our findings indicate that the subset attributes derived from a combination of Information Gain, GINI Index, and Correlation Matrix can improve the performance of machine learning models. Using a Support Vector Machine with proposed innovative score-based zero shot technique, promising results were achieved on average with 96.75% precision, 69.39% recall, and 80.81% F1-Score. Additionally, high-quality features, such as the *percentageDifference* can enhance performance in multi-label classification tasks, providing valuable insights for practitioners and researchers. The research implications and significance highlight the practical applications and strategic value of these findings, contributing to the advancement of knowledge and practice in the field of software engineering, particularly for multi-label classification tasks.

INDEX TERMS Feature selection, app user reviews, machine learning, multi-label classification, text mining.

I. INTRODUCTION

Recognizing the significance of user reviews, recent research has focused on automating the classification of these reviews to extract actionable insights efficiently. These reviews serve as crucial indicators of user satisfaction, preferences, and pain points, providing developers with essential insights for enhancing app features and overall user experience. Recent studies have emphasized the importance of automating the analysis of user reviews to efficiently extract actionable insights from this wealth of data ([1], [2]). However, extracting and analyzing user requirements from these reviews can be very challenging [3]. This is because user reviews are often vast volume [4], [5], [6], unstructured nature [7], [8], and inconsistent quality of user reviews [9], [10] necessitate sophisticated methodologies to parse and interpret the information contained within them. The integration of advanced feature selection methods into app user review analysis offers multiple advantages, including improved model accuracy, reduced computational demands, and enhanced interpretability of results.

Feature selection for app user reviews involves identifying the most relevant features that contribute to the understanding of user feedback and app performance. Feature selection, also known as variable selection or attribute selection, is the process of choosing the most relevant attributes from the data for predictive modeling. Basha and Rajput [11] illustrated the efficacy of utilizing Term Frequency as a feature selection criterion, significantly improving classification accuracy. Aruna et al. [12] further validated this in the context of Twitter sentiment analysis, where the application of Naïve Bayes algorithms alongside optimized feature selection yielded favorable results. Such insights underscore the necessity of

The associate editor coordinating the review of this manuscript and approving it for publication was Biju Issac.

integrating robust feature selection methodologies can lead to more accurate sentiment assessments.

The implications of feature selection extend beyond sentiment analysis into broader text classification contexts. Thatha et al. [13] noted the increasing relevance of feature selection methodologies in addressing the challenges posed by the extensive and often noisy textual data generated through user reviews. These reviews frequently contain informal language, grammatical errors, and other forms of noise that can obscure meaningful insights [14]. The identification and application of effective feature selection techniques are thus paramount for extracting valuable insights from diverse data sources, ranging from textual documents to multimedia content.

Further advancements in feature selection techniques have been proposed for multi-label classification problems. Pereira et al. [15] categorized various methods for this task, pointing out the lack of a unified framework for objective comparison. Mishra and Singh [16] introduced a clustering-based feature selection method for multi-label classification, demonstrating significant improvements in precision and recall through dimensionality reduction and ultimately leading to improved user satisfaction and app quality.

Recent developments also include the integration of novel techniques like Brain Storm Optimization (BSO) for data classification, as demonstrated by Pourpanah et al. [17]. Their hybrid model, which combines incremental learning neural networks with BSO, achieved superior results compared to traditional optimization techniques. Furthermore, Koka and Fang [18] proposed visualization-based feature selection methods, which leverage visual analytics to help users identify significant features while discarding irrelevant ones. Such advancements suggest that the utilization of innovative feature selection approaches could substantially enhance the analysis of app user reviews.

Therefore, a focused effort is required to develop tailored feature selection strategies that can effectively address the distinct challenges presented by the analysis of app user reviews. The objective of this study consists of two,

1) to evaluate the impact of different feature selection techniques on the performance of machine learning models in multi-label classification tasks for app user review analysis; and
2) to identify and compare the performance of various machine learning models in multi-label classification tasks when feature selection techniques are applied.

To achieve the objectives mentioned, we formulated two research questions to guide our study, as outlined in the following items.

1) How does feature selection impact the performance of machine learning models in multi-label classification tasks for user review analysis?
2) Which machine learning model performs best in multi-label classification tasks when feature selection is applied?

This study contributes to the body of knowledge in software engineering by:

1) Proposing a novel features selection technique that combines Information Gain, Gini Index, and Correlation Matrix.
2) Proposed Score-Based Zero-Shot technique through the introduction of a classification framework, which evaluates multi-label user reviews by calculating score differences between problem discovery (PD) and feature request (FR) categories. By using percentage-based thresholds, the method adaptively assigns primary or combined labels, ensuring nuanced classification and improved handling of overlapping label categories.
3) Empirically demonstrating the importance of feature selection in improving model performance.

In the rest of this paper: Section II is about related work. Section III provides an overview of the material and method. Section V explains the potential threats to validity. Section IV presents the result and discussion of the experiment. Finally, we conclude this work in VI and further work directions in Section VII.

II. RELATED WORK

Current research in feature selection for text classification, particularly in the context of multi-label classification tasks like app user review analysis, presents a variety of approaches. These include filter methods, wrapper methods, and hybrid techniques that aim to optimize the balance between feature reduction and model performance.

A. FILTER METHODS

One widely discussed technique is the application of filter-based methods for feature selection, which measure the relevance of each feature independently of the model. Kim and Zzang [19] compared eight prominent filter-based feature selection techniques such as Information Gain (IG), Chi-squared (CHI), Gini index, and normalized difference measure (NDM) using multinomial Naïve Bayes and Support Vector Machines. Their study demonstrated the superiority of certain methods over others in text categorization tasks, highlighting the effectiveness of Gini Index and Information Gain for feature selection in large-scale datasets. Other studies also support the utility of these methods, particularly IG and CHI, which have been successfully applied in various text mining tasks to improve classifier performance in multi-label settings (Zaman et al. [20]). These findings emphasize the importance of selecting relevant features to enhance classification accuracy and reduce computational cost.

B. WRAPPER METHODS

Another major category in feature selection techniques is wrapper methods, which evaluate subsets of features based on the performance of a specific learning algorithm. Methods like the Brain Storm Optimization (BSO)-based feature selection proposed by Pourpanah et al. [17] combine incremental learning models with feature optimization techniques to

enhance classification outcomes. While filter methods are computationally less expensive, wrapper methods often yield more accurate results, as they tailor the feature selection process to the specific machine learning model in use. However, these methods can be computationally intensive, particularly when dealing with high-dimensional datasets, such as those generated from app user reviews.

C. HYBRID TECHNIQUES

In addition to filter and wrapper methods, hybrid approaches have been proposed to combine the strengths of both categories. For instance, Khan et al. [21] introduced an intelligent hybrid feature selection technique for sentiment analysis, which effectively reduced the high-dimensional feature space while enhancing the predictive performance of the machine learning models used. This method leverages both feature extraction and feature selection to optimize sentiment model learning in textual data, a process that is critical for tasks like user review analysis, where textual features are often highly correlated and redundant.

D. COMPARISON WITH PREVIOUS SCHOLARS

Despite these advancements, gaps remain in the application of feature selection techniques specifically tailored to multi-label classification tasks in app user review analysis. Many existing studies focus on binary or multi-class classification tasks, leaving multi-label classification less explored. Pereira et al. [15] categorized feature selection methods specifically for multi-label classification and noted the lack of a unified framework to compare these methods objectively. This observation underscores the need for further research into feature selection methods that are optimized for multi-label tasks, which are more complex due to the simultaneous prediction of multiple labels for a single instance.

Recent developments in feature selection for sentiment analysis and text mining have also introduced new approaches, such as clustering-based methods for multi-label classification (Mishra and Singh [16]), which further improve model performance by reducing the feature space based on feature similarity. However, these methods require further empirical validation, particularly in the context of app user review analysis, where the diversity and complexity of textual features can impact the effectiveness of traditional feature selection techniques.

Therefore, the proposed study is different from prior studies (see Table 1) by concentrating on multi-label classification tasks with an innovative advanced feature selection techniques. We will evaluate these techniques across various machine learning methods. This focus addresses the limitations identified in previous research, particularly regarding the application of effective methods for complex classification scenarios, thus contributing valuable insights to the software engineering field.

In this study, a case study will be conducted to analyze user reviews in the transport domain. By applying the proposed feature selection techniques and machine learning models to this dataset, the study aims to show the practical benefits and improvements in classification accuracy, precision, recall, and F1-score. This real-world application will serve as a clear example of how the research can be useful for both practitioners and researchers in the field. Additionally, the study will help highlight the effectiveness of the proposed methods in improving the understanding and analysis of user feedback in Malaysian transport enforcement organization.

III. MATERIALS AND METHODS

This section discusses the material and methods we used to extract textual features up until classifying the user reviews.

A. MATERIALS

In this study, several instruments were used to collect data from app stores:

1) **Google play data scrapper tools:**
Google-Play-Scraper is the tool used to extract user feedback from the Google Play Store. Google-Play-Scraper is an official free and good API and it's available in both Python and Node.js without any external dependencies. This Google Play data scraper provide a various feature such as collect instant reviews, any region or language, search for any query and get multiple applications from any developer. This API will save the output into separate columns and rows for each item in the dataset.
2) **Programming languages:**
Programming languages Python were used in this study to write scripts for automated the data scraping process. Python is a versatile and powerful programming language and one of the top ten most popular programming languages.
3) **Machine learning tools:**
We utilized RapidMiner software to perform data mining tasks. For example, [1] and [29] used Naïve Bayes to classify user reviews into bug reports, non-functional requirements, usability, user experience, and ratings. Meanwhile, [30] used SVM to classify user reviews based on non-functional requirements and usability and [31] used neural networks to select top k features for model training and evaluation. Additionally, we are using a Deep Learning approach with a multi-layer feedforward artificial neural network, which is implemented using the H2O 3.42.0.1 algorithm [32].
4) **Text mining tools:**
Text mining tools such as Natural Language Processing (NLP) libraries or frameworks were used to preprocess and analyze textual data from app reviews. The text mining tools provided functions or algorithms for tokenization (splitting text into words or phrases), stemming (reducing words to their base form), lemmatization (reducing words to their dictionary form), part-of-speech tagging (labeling words with their grammatical category), sentiment analysis (determining

TABLE 1. Summary of related research studies.

Year	References	Research Objective	Problem Statement	Methods	Expected Results
2023	Kaur & Kaur [22]	Improve word representation for requirement classification	Traditional models fail to capture syntactic structure	LSTM-CNN, BERT enhancements	Enhanced classification accuracy
2023	Abdulhussien et al. [23]	Design feature selection models using quantum algorithms	Classical GA stagnation in local optima	QIGA, PSO	Higher verification accuracy
2023	Elhassan et al. [24]	Develop feature selection for Arabic sentiment analysis	Insufficient accuracy in sentiment analysis	Deep learning, PCA, GA	Improved sentiment classification accuracy
2023	Duarte & Berton [25]	Review semi-supervised learning for feature selection	Ineffective utilization of unlabeled data	Hybrid approaches	Robust sentiment analysis
2022	Zaman et al. [20]	Propose new approaches for feature selection	Challenges in streaming high-dimensional data	Online feature selection techniques	Adaptive models for real-time data
2021	Kadkhoda & Naderi [26]	Optimize feature selection in social networks	Existing methods do not prioritize critical features	AFIF, stepwise regression	Reduced overfitting, optimized selection
2021	Al-Hawari et al. [27]	Investigate the influence of feature selection on the performance of classifiers for app reviews	Classifying user reviews is challenging due to noisy and redundant information. Effective feature selection can boost classifier accuracy by filtering irrelevant features	Information Gain (IG), Chi-square, classifiers (e.g., J48, NB, RF)	Expect significant accuracy improvement in app review classification by applying IG and Chi-square feature selection techniques
2021	Tubishat et al. [28]	Extract explicit aspects in sentiment analysis	Lack of handling explicit aspects	WOA, optimal rules	Higher sentiment classification accuracy
2024	**Current Study**	**Develop a feature selection technique for multi-label classification**	**Gaps in multi-label feature selection techniques**	**Advanced feature selection (hybrid), supervised learning**	**Improved performance in multi-label tasks**

TABLE 2. Attributes of the user review data.

No.	Attribute Name	Explanation with Example
1	reviewId	A unique identifier for each review. *Example: 80da3dca-2ea3-4553-994f-14463d1e7be8*
2	userName	The name of the user who submitted the review. *Example: alleria Windr*
3	userImage	URL of the user's profile image. *Example: https://play-lh.googleuser../a-/ACB-R...*
4	content	The content of the review written by the user. *Example: Keep showing need to change my password, click on forget password then mentioned failed to send tac to email. pls have a proper message to the user if the service is not ready for use.*
5	score	The rating provided by the user. *Example: 3*
6	thumbsUpCount	The number of thumbs-up received by the review. *Example: 2*
7	reviewCreatedVersion	The version of the app at the time of the review. *Example: 2.0.0*
8	at	Timestamp when the review was posted. *Example: 12/2/2023 1:25:29 AM*
9	replyContent	replied by application administrator. *Example: We apologize for your difficulties in accessing the application and making transactions.*
10	repliedAt	Timestamp when the administrator was replied *Example :12/2/2023 8:30:30 AM*
11	appVersion	The version of the app referred to in the review. *Example: 2.0.0*

TABLE 3. Mining technique and features extraction.

Text Mining	Description
1. SA (Sentiment Analysis)	Determining the sentiment of the review by using VADER tools.
2. POS (Part-of-Speech)	Tagging words with their grammatical parts of speech.
3. TM (Topic Modeling)	Identifying topics present in the text by utilizing LDA.
4. keyBERT	A keyword extraction technique using BERT embeddings.
5. RAKE	Rapid Automatic Keyword Extraction algorithm.

sentiment polarity), topic modeling (identifying latent topics), etc. in this study, the different techniques are applied for mining and extracting features as explained in Table 3.

B. METHODS

This subsection, the user reviews from Google Play Store are analyzed through automated techniques, including zero-shot labeling, feature extraction, and classifier training. Preprocessing, feature selection, and evaluation of classifiers such as Naïve Bayes and Support Vector Machine are employed to categorize app reviews based on identified problem discovery and feature requests. The methodology integrates Google-Play-Scrape, data labeling, and advanced machine learning techniques to enhance review classification accuracy and performance. Figure 2 shows an overview of the classification process for multi-label tasks using a new technique called Score-Based Zero-Shot. The details of the process are explained in the following items.

1) **Data Collection:** Researchers collect a dataset consisting of app reviews from app stores. In our study, we used government apps as our dataset and focused on user reviews written in English (714 total of records, containing 1,705 sentences in total). To collect user reviews from the Google Play Store, we used the Google-Play-Scraper technique. The data contains several attributes, such as reviewId,

TABLE 4. Accuracy of the classification technique using IG+GI+CM on App reviews from the google store.

Multi-labeling	(a) Naïve Bayes				(b) Deep Learning			
	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score	Accuracy
Problem Discovery: Feature Request	86.51%	100.00%	92.77%	91.70%	90.43%	97.20%	93.69%	93.20%
Problem Discovery	100.00%	90.70%	95.12%		97.73%	98.85%	98.29%	
Feature Request; Problem Discovery	0.00%	0.00%			50.00%	9.09%	15.38%	
Multi-labeling	**(c) Decision Tree**				**(d) Support Vector Machine (SVM)**			
	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score	Accuracy
Problem Discovery: Feature Request	90.00%	95.19%	92.52%	92.20%	91.38%	99.07%	95.07%	94.60%
Problem Discovery	94.62%	100.00%	97.24%		98.86%	100.00%	99.43%	
Feature Request; Problem Discovery	100.00%	8.33%	15.38%		100.00%	9.09%	16.67%	

username, userImage, content, score, thumbsUpCount, reviewCreatedVersion, at, replyContent, repliedAt, and appVersion, as shown in Table 2. Additionally, some other attributes (marked with *) were created manually to assist in processing the user reviews, as shown in Table 6.

2) **Preprocessing:** Text preprocessing techniques such as transformation (remove noise and text lowercase), spelling correction (contraction), tokenization and normalization (stop-word removal, stemming) are applied to clean the text data (see Table 5) in order to enhance the learning efficiency of the machine learning model to be built later.
3) **Features Selection:** Feature selection is one of the important steps in data analysis. Feature selection, also called variable selection or attribute selection is the process of selecting the attributes from the data that are most relevant to the predictive modeling. A systematic review of existing feature selection methods for text mining applications reveals valuable insights into their comparative effectiveness. According to [33], techniques such as Information Gain (IG) feature selection have demonstrated promising results in datasets such as Reuters-21578 and 20 Newsgroup corpus, outperforming alternatives such as Chi-square, Document Frequency (DF), and Term Strength (TS). Therefore, this study will employ a combination of feature selection such as Information Gain, GINI Index, Manual Selection and Correlation Matrix.
4) **Features Extraction:** Features are extracted from the preprocessed text data. Bhatia et al. [34] used features such as review rating, review length, review tense, and review sentiment score in their study. In this study, we will utilize keyBERT for keyword extraction, which uses BERT embedding, and we will also employ the Rake algorithm (Rapid Automatic Keyword Extraction).
5) **Proposed Score-Based Zero-Shot Techniques:** This dataset will be labeled into three categories which are problem discovery [bug reports, issues, dissatisfaction emotion], features request [feature request, content request, promise, improvement request, idea, suggestion and features shortcoming] and mix. Some researchers like [9], [29], [35], [36], and [37] have manually labeled user reviews based on predefined categories, such as bug reports, non-functional requirements, usability, user experience, or ratings. Most deep learning models are supervised, meaning they require large amounts of labeled data specific to the domain. This creates an opportunity to develop new methods for zero-shot or few-shot learning [38]. Therefore, in this study, each of these user reviews was automatically labeled using the proposed score-based zero-shot technique as shown in Algorithm 1. The following sections explain the process in more detail, and a sample output is shown in Figure 1 for reference.

(a). Let $Score_{PD}$ and $Score_{FR}$ represent the sets of scores obtained from two datasets labeled PD and FR, respectively. Similarly, let $Label_{PD}$ and $Label_{FR}$ represent the corresponding sets of labels.

(b). Define $maxScore_{indexPD} = \arg\max Score_{PD}$ and $maxScore_{indexFR} = \arg\max Score_{FR}$ as the indices of the maximum scores in the PD and FR datasets, respectively.

(c). Let $highest_{slPD}$ and $highest_{slFR}$ denote the labels corresponding to the maximum scores in the PD and FR datasets, respectively, obtained as:

$$highest_{slPD} = Label_{PD} maxScore_{indexPD} \quad (1)$$

$$highest_{slFR} = Label_{FR} maxScore_{indexFR} \quad (2)$$

(d). The highest scores in the PD and FR datasets, denoted by $highest_{sPD}$ and $highest_{sFR}$, respectively, are:

$$highest_{sPD} = Score_{PD} maxScore_{indexPD} \quad (3)$$

$$highest_{sFR} = Score_{FR} maxScore_{indexFR} \quad (4)$$

(e). The percentage difference between the highest scores is calculated as:

$$diff = \frac{|highest_{sPD} - highest_{sFR}|}{\max(highest_{sPD}, highest_{sFR})} * 100 \quad (5)$$

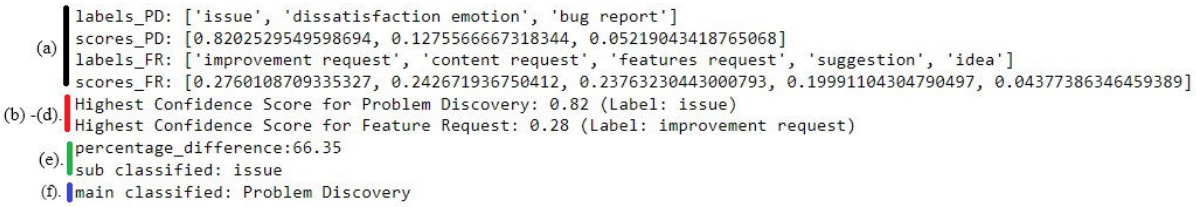

FIGURE 1. Sample output step a to f.

Based on the percentage difference, the datasets are classified into the following categories:

$$\mathrm{UR_{sc}} = \begin{cases} \left(\mathrm{highest_{slPD}}; \mathrm{highest_{slFR}}\right), & \text{if diff} \leq 50.00 \\ \mathrm{highest_{slPD}}, & \text{otherwise} \end{cases} \tag{6}$$

(f). Finally, the main labeling classification for the dataset is determined as:

$$\mathrm{UR_{mc}} = \begin{cases} \mathrm{mClass[highest_{slPD}]}, & \text{if diff} \leq 50.00 \\ \mathrm{mClass[highest_{slFR}]}, & \text{otherwise} \end{cases} \tag{7}$$

6) **Classifier Training:** Supervised machine learning models are trained on the extracted features to classify app reviews into predefined categories such as problem discovery, features request and mix. In this study, we applied information gain for feature selection and Optimized Parameter Grid which was used for hyperparameter tuning to find optimal values for the hyperparameters of a model. These techniques were applied to reduce time and resources and enhance performance for the built classifier model. Reference [34] utilized supervised machine learning techniques like Multinomial Naïve Bayes, Linear SVC, and CART Decision Tree for classification. Meanwhile, in our experiment, we applied classification algorithms such as Naïve Bayes, Deep Learning, Decision Tree and Support Vector Machine.
7) **Performance Evaluation:** The performance of the classification models is evaluated using metrics like precision, recall, and accuracy and the stratified 10-fold cross-validation method was applied in this study as well where K-fold cross-validation (K-fold CV) is a statistical method for selecting parameters and evaluating ML models [39].

The algorithm 1 presented is designed to categorize textual reviews into two primary classifications: Problem Discovery (PD) and Feature Request (FR). This classification is facilitated through a predefined pipeline (bugDetect) that assigns probability scores to each classification category. The methodology aims to enhance datasets by appending classification labels and hierarchical mappings to their respective broader categories. The algorithm follows a structured workflow to ensure a systematic classification of review texts.

1) Initialize label sets and mappings
Define *mainClassification*, which maps subcategories to main categories. Define label sets for *ProblemDiscovery*(*PD*) and *FeatureRequests*(*FR*).
Problem Discovery: [bug report, issue, dissatisfaction emotion] and Feature Request: [features request, content request, improvement request, idea, suggestion]
2) Loop through each review in the dataset
The algorithm iterates through each review contained within the dataset *df*. Each iteration processes an individual review, denoted as *r*, and extracts the textual content from the review for classification purposes.
3) Extract and classify the review
The extracted review text undergoes classification using the *bugDetect* pipeline that utilized zero-shot-classification and "facebook/bart-large-mnli" for PD and FR labels. Subsequently, the highest-scoring label for each category, along with its corresponding probability score, is identified.
4) Determine the main classification based on scores
If the **PD score** $\geq$ **FR score**, the review is categorized under **Problem Discovery**, and the corresponding label is mapped to its broader category using *mainClassification*. If the **FR score** surpasses the **PD score**, the review is categorized as a **Feature Request**, with a similar mapping procedure. If both scores are equal, the review is simultaneously assigned to both categories, and each respective label is mapped to its broader classification.
5) Calculate the percentage difference
A quantitative metric *diff* is calculated to measure the relative difference between the highest probability scores of PD and FR.
6) Refine the classification based on percentage difference
Based on the computed *diff* value, the sub-classification *UR_sc* and main classification *UR_mc* are updated accordingly. If *diff* ≤ 50, the classification retains both PD and FR labels. Otherwise, the classification retains only the dominant label (i.e., the one with the highest probability score).

IV. RESULT AND DISCUSSION

This section presents the results for research questions RQ1 and RQ2 derived from two experiments.

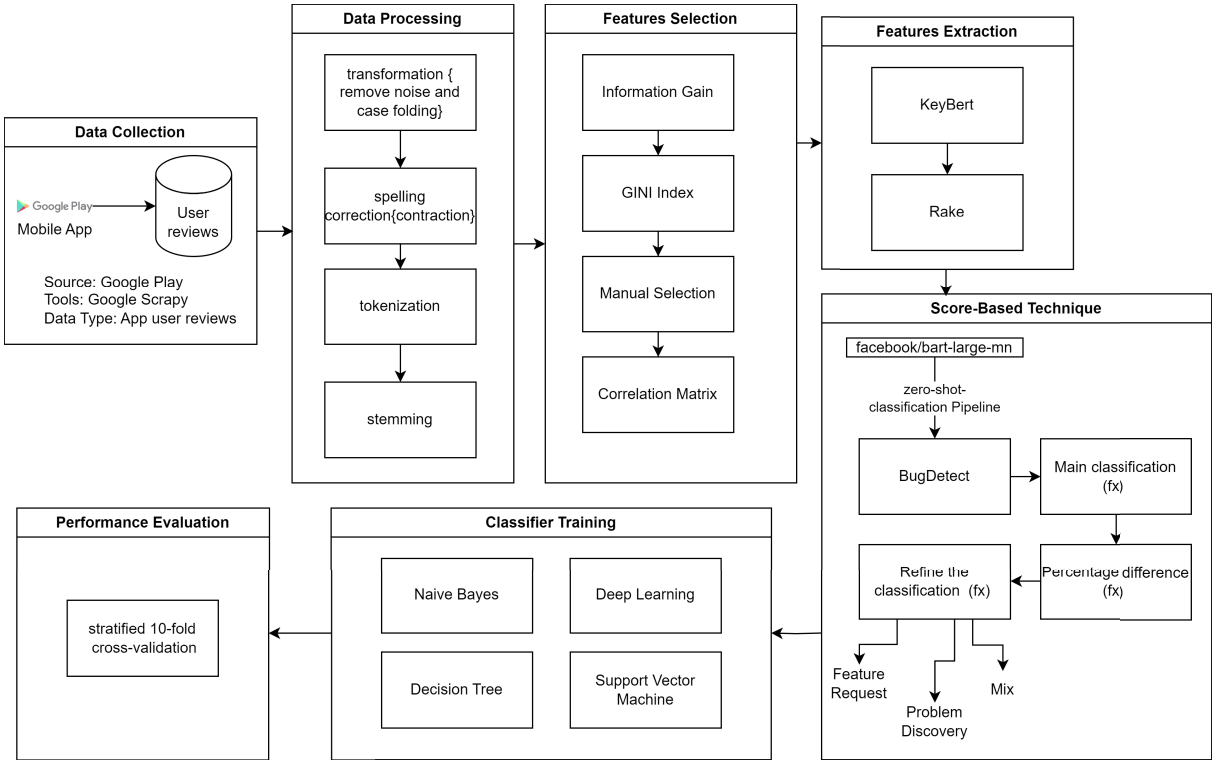

FIGURE 2. Overview user reviews analysis methodology.

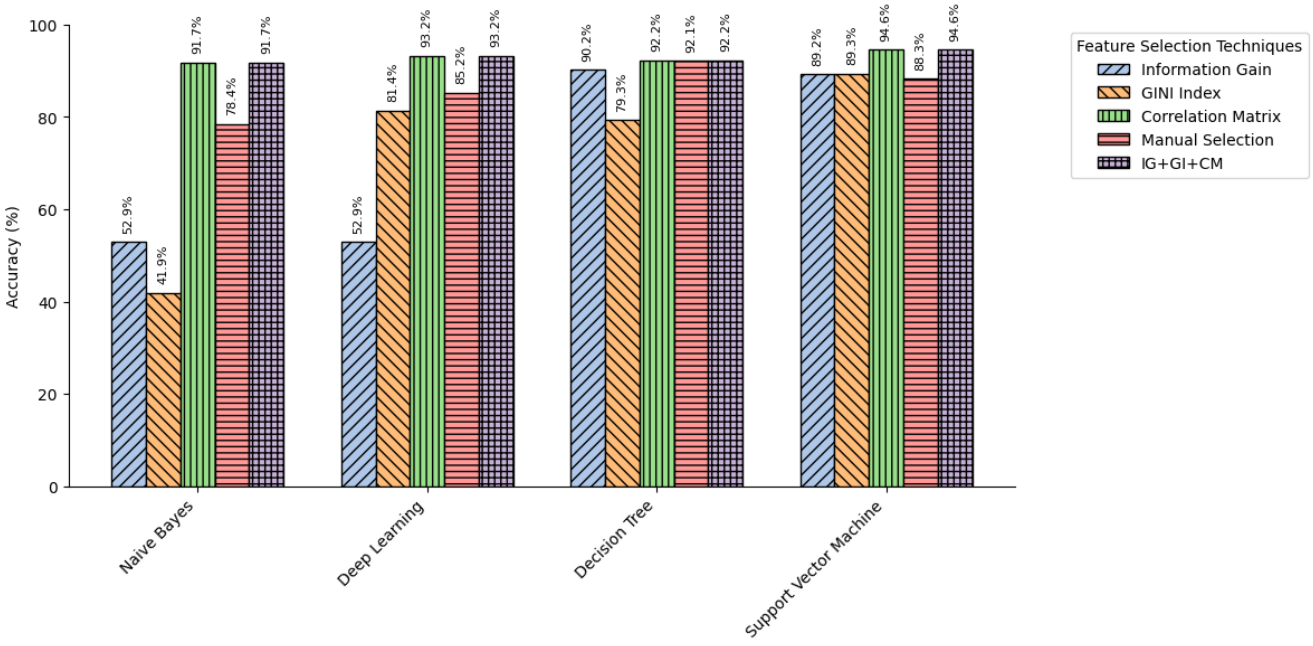

FIGURE 3. IG+GI+CM Feature selection across the Machine Learning.

A. IMPACT OF FEATURE SELECTION ON MACHINE LEARNING PERFORMANCE [RQ1]

1) MANUAL FEATURE SELECTION WAS APPLIED

The classification accuracies of all models for all sentence types are shown in Table 7. This accuracy result is derived from the manual selection of features without relying on feature selection techniques like Information Gain, GINI Index, Chi-square statistic, Correlation Matrix, or others. The Decision Tree model has the highest accuracy (92.1%), making it the most effective model among those tested in terms of accuracy.

Algorithm 1 Algorithm for Classification of Reviews Into Problem Discovery and Feature Request Categories

Input: Dataset *df* with review texts; Classification pipeline *bugDetect*
Output: Dataset *df* enriched with sub- and main classifications
Data: Label sets for Problem Discovery (PD) and Feature Requests (FR); Mapping dictionary *mainClassification*

```
/* Initialize label sets and mappings */
```
1. Define *mainClassification*, *problemDiscoveryLabels*, and *featureRequestLabels*

```
/* Loop through each review in the dataset */
```
2. **foreach** *row r in df* **do**

```
    /* Extract and classify the review */
```
    3. Extract review text *review* from *r*
    4. Classify *review* using *bugDetect* for both PD and FR labels
    5. Retrieve the highest-scoring label and score for each category (PD and FR)

```
    /* Determine the main classification based
       on scores */
```
    6. **if** *Highest PD score* > *Highest FR score* **then**
        7. Assign *review* to Problem Discovery
        8. Map the PD label to its main category using *mainClassification*
    9. **else if** *Highest FR score* > *Highest PD score* **then**
        10. Assign *review* to Feature Request
        11. Map the FR label to its main category using *mainClassification*
    12. **else**
        13. Assign *review* to both categories (PD and FR)
        14. Map both labels to their respective main categories

```
    /* Calculate the percentage difference */
```
    15. Compute the percentage difference:

$$\text{diff} = \frac{|\text{highest}_{\text{PD}} - \text{highest}_{\text{FR}}|}{\max(\text{highest}_{\text{PD}}, \text{highest}_{\text{FR}})} \times 100$$

```
    /* Refine the classification based on
       percentage difference */
```
    16. Update sub-classification (UR_{sc}):

$$UR_{\text{sc}} = \begin{cases} (\text{highest}_{\text{PD}}, \text{highest}_{\text{FR}}), & \text{if diff} \leq 50 \\ \text{highest}_{\text{PD}}, & \text{otherwise} \end{cases}$$

    17. Update main classification (UR_{mc}):

$$UR_{\text{mc}} = \begin{cases} \text{mClass}[\text{highest}_{\text{PD}}], & \text{if diff} \leq 50 \\ \text{mClass}[\text{highest}_{\text{FR}}], & \text{otherwise} \end{cases}$$

Meanwhile, the Deep Learning model shows a good balance with a relatively high accuracy (85.2%). The average

TABLE 5. Text Pre-processing techniques.

Pre-Processing	Description
1.RN (Remove Noise)	Cleansing the text from irrelevant characters or information.
2.LW (Lowercase)	Converting all text to lowercase for uniformity.
3.CTR (Contraction)	Resolving contractions to their expanded forms.
4.Tokenization	Breaking down the text into individual tokens or words.
5.SW (Stop-word)	Removing common words that do not contribute to the meaning.
6.ST (Stemming)	Reducing words to their root form.
7.LM (Lemmatization)	Converting words to their base or dictionary form.

F1-Score, Precision, and Recall are 63.75%, 70.71%, and 62.16%, respectively. Despite, Deep Learning models are generally powerful, our results showed that they performed lower than the Decision Tree when manual feature selection was applied. This is consistent with findings by Saleem et al. [40], who reported that deep learning models tend to perform better when fed with features selected through filter-based methods. Additionally, deep learning models often require large amounts of data for training. In our study, we used a relatively small dataset with only 714 records. This limited dataset likely contributed to the poorer performance of the deep learning model, as deep learning algorithms typically struggle with smaller datasets. This finding supports the conclusions of Saleem et al. [40] and Liu et al. [41], who both found that deep learning models do not perform well with small datasets.

The Naïve Bayes model, has the lowest accuracy (78.4%) and the Support Vector Machine (SVM) provides a good accuracy (88.3%). These results suggest that the Decision Tree model is particularly effective for this multi classification task based on manual selection feature. This aligns with the research by Zhang et al. [42], which highlighted that Decision Tree algorithms are effective for both binary and multi-class tasks.

2) FEATURE SELECTION TECHNIQUE WAS APPLIED

Table 8, showing that the Support Vector Machine (SVM) model has the highest accuracy (94.6%) and on average scores 91.20% across all feature selections making it the most effective model among those tested in terms of both accuracy and learning feature selections from the data. This research is aligned with findings from Wongso et al. [43] explores SVM for text classification in Indonesian, and Bhatia and Sharma [31] use SVM with feature selection for sector classification of requirements. This validates the potential of SVM in conjunction with robust feature selection.

The Deep Learning model shows a good balance with relatively high accuracy (93.2%) and Naïve Bayes model, has the lowest accuracy (91.7%) if relies on Gini Index. The Decision Tree provides a good accuracy (92.2%) it represents

TABLE 6. Standard and additional mobile Apps reviews attribute.

Feature	Info Gain	Gini Index	Correlation Matrix	Manual Selection
reviewId	1.208			
userName	1.208			
userImage	1.208			
userReview	1.208			
keyBERT*	1.208			/
IOMRake*	1.208			/
NLTKRake*	1.208			/
TMLDA*	1.202			/
Noun Phrase*	1.152			/
Noun*	1.083			/
Verb*	1.026			/
%Difference*	0.976	0.449	0.827	/
VADERPolarity*	0.045	0.028		/
VADERScore*	0.029	0.011	0.024	/
Score	0.026	0.008	0	
At	0.024	0.015		
reviewCreatedVersion	0.018	0.005		
appVersion	0.018	0.005		/
thumbsUpCount	0.014	0.005	1	/
Reviewlength*	0.008	0.004	0.968	/

Note: Rows marked with an asterisk (*) are manually created.

TABLE 7. Machine learning - manual feature selection (%).

Classifier	Accuracy	Standard Deviation
1 Naïve Bayes	78.40	±7.3
2 Deep Learning	85.20	±3.0
3 Decision Tree	**92.10**	±3.2
4 Support Vector Machine	88.30	±4.3

TABLE 8. Accuracy performance based on feature selection technique (%).

Classifier	Info Gain(IG)	Gini Index (GI)	Correlation Matrix (CM)	Manual Selection (MS)	IG+ GI + CM
Naïve Bayes	52.9	41.9	91.7	78.40	91.7
Deep Learning	52.9	81.4	93.2	85.20	93.2
Decision Tree	90.2	79.3	92.2	92.10	92.2
Support Vector Machine	89.2	89.3	94.6	88.30	**94.6**

a second lower in average (89.20%) if compared to other classifiers.

Table 8 and Table 4 showing that the subset of features derived from Information Gain, GINI index, and Correlation Matrix which used a more refined feature selection process, resulted in improved performance across all models in terms of precision, recall, accuracy and F1-Score. Our research is aligned with findings from Al-Hawari et al. [27], where they explore Information Gain and Chi-square for app review classification, demonstrating improved performance. Meanwhile Bakar et al. [44] also explore Information Gain (IG) along with other features and how it enhances sentiment analysis models.

The Support Vector Machine (SVM) model showed the most significant improvement, achieving the highest accuracy (94.6%) in the combination of feature selection techniques. The Naïve Bayes model also showed a notable improvement in accuracy and gains, making it a more competitive model as shown in Figure 3.

In addition, to answer this research question, the hypotheses need to be tested based on the hypotheses declaration as shown inTable 9 to test the impact of feature selection on model performance, we used a paired t-test (Equation (8)) to compare the accuracy of models before and

TABLE 9. Hypothesis for RQ01.

Hypothesis	Description
Null Hypothesis (H0)	Feature selection does not significantly impact the performance of machine learning models in multi-label classification tasks.
Alternative Hypothesis (H1)	Feature selection significantly improves the performance of machine learning models in multi-label classification tasks.

TABLE 10. Paired t-test result.

Paired Samples T-Test

			Statistic	p
IG+GI+CM	Manual Selection	Wilcoxon W	10.0	0.125

Note. H_a μ Measure 1 - Measure 2 $\neq$ 0

Descriptives

	N	Mean	Median	SD	SE
IG+GI+CM	4	93.2	93.2	1.18	0.592
Manual Selection	4	86.0	86.8	5.80	2.900

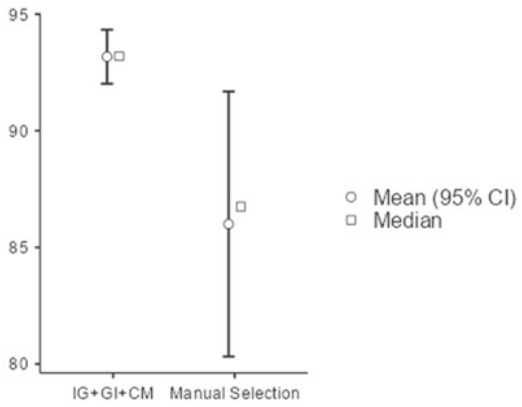

FIGURE 4. Paired t-test result.

after feature selection.

$$t = \frac{\overline{d}}{\frac{s_d}{\sqrt{n}}} \tag{8}$$

where, $\overline{d}$ is the mean difference in accuracy between the manual selection and combination features selection. Meanwhile, sd is the standard deviation of the differences and n is the number of models compared.

Based on the t-test calculation, the results of this study, while not statistically significant according to the p-value of 0.125, provide important insights that align with and complement existing research in the field of feature selection for text categorization. Although the statistical test did not reject the null hypothesis, indicating no significant improvement in model performance, the practical results tell a different story. For instance, the combination of Information Gain (IG), Gini Index (GI), and Correlation Matrix (CM) with feature selection methods demonstrated a notable increase in accuracy, particularly with the Support Vector Machine (SVM), where the accuracy rose from 88.3% to 94.6%. This outcome echoes the findings of Kim and Zzang [19], who also observed superior performance using similar feature selection techniques over various alternatives, such as Chi-squared and odds ratio. The observed increase in accuracy and the reduced computational time in this study reinforce the importance of feature selection, as highlighted by Zaman et al. [20], where methods like IG, chi-squared, and Gini Index have shown improvements in handling high-dimensional data. This work supports earlier hypotheses suggesting that feature selection can optimize model performance, even when statistical measures do not confirm significance. The results here extend the understanding established by Bakar et al. [44], who also confirmed the utility of methods like IG and GI in sentiment analysis, further reinforcing the growing body of evidence supporting feature selection in machine learning. Hence, this research corroborates findings from earlier studies, highlighting the practical advantages of feature selection, even when statistical tests do not fully capture its impact.

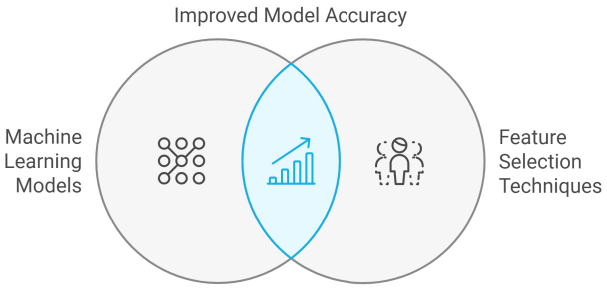

FIGURE 5. Enhancing Model Performance through Feature Selection.

In summary, we can conclude that feature selection has a significant impact on model performance, as shown in Figure 5. The details of the following item are provided to help with a better understanding.

1) This study examines how feature selection affects the performance of machine learning models. It compares the results of models that use feature selection techniques with those that do not. The findings show that applying feature selection methods like Information Gain, GINI Index, and Correlation Matrix can significantly improve model performance.
2) Among the models, the Support Vector Machine (SVM) achieved the best results. Its accuracy increased to 94.6% when feature selection techniques were used, with precision exceeding 90%. This demonstrates the effectiveness of feature selection in enhancing model accuracy and reducing computational demands.
3) The Decision Tree model also showed stable performance. Its accuracy was 92.1% without feature selection and slightly improved to 92.2% with feature

TABLE 11. Hypothesis for RQ02.

Hypothesis	Description
Null Hypothesis (H0)	There is no significant difference in the performance of different machine learning models in terms of accuracy for multi-label classification tasks when the IG+GI+CM as feature selection technique is applied.
Alternative Hypothesis (H1)	The Support Vector Machine (SVM) model performs significantly better in terms of accuracy for multi-label classification tasks when feature engineering is applied.

TABLE 12. ANOVA single factor test.

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Manual Selection	4	344	86	33.63333333
IG+GI+CM	4	373	93.175	1.4025

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	102.961	1	102.961	5.877	0.052	5.987
Within Groups	105.108	6	17.518			
Total	208.069	7				

selection. This suggests that while Decision Tree models are already robust, they can still benefit from feature selection with small dataset.

B. BEST PERFORMING MACHINE LEARNING MODEL FOR MULTI-LABEL CLASSIFICATION WITH FEATURE SELECTION [RQ2]

In this research question, we need to execute an experiment to compare the performance of various machine learning models (Naïve Bayes, Deep Learning, Decision Tree, and Support Vector Machine) using the feature selection technique. Identifying the best-performing model can provide valuable insights for practitioners in selecting the most suitable model for multi-label classification tasks for user review analysis. To achieve this, we formulate the second hypothesis as shown in Table 11.

To compare the performance of different models, we used ANOVA (Analysis of Variance) to test (Equation (9)) if there are significant differences in accuracy among the models.

$$F = \frac{MS_{between}}{MS_{within}} \tag{9}$$

where, MS_{between} is the mean square between the groups and MS_{within} is the mean square within the groups.

The analysis of the raw data aligns with prior research and reinforces the importance of feature selection in multi-label classification tasks. With a p-value of 0.052, which is just slightly higher than the 0.05 threshold at the 5% significance level, this study supports the null hypothesis, indicating no significant difference in the performance of different machine learning models when the IG+GI+CM feature selection technique is applied as shown in Figure 3. However, this value is close to significance, suggesting a trend worth exploring further. This finding echoes the work of Sridharan and Sivakumar [33], who emphasized the critical role of feature selection in improving model performance, particularly in reducing dimensionality for text mining tasks. Moreover, our results align with the observations made by Mishra and Singh [16], who demonstrated improvements in precision and recall through feature selection in multi-label classification, which is consistent with the precision, recall, and F1-score trends observed in this study.

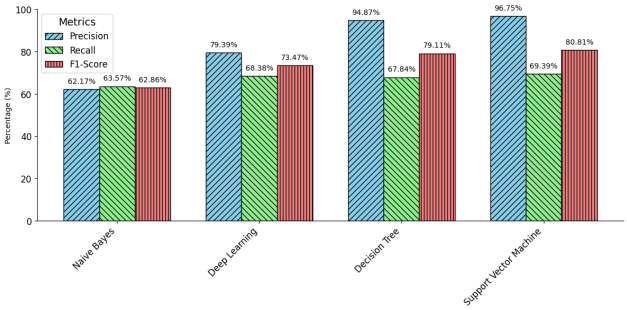

FIGURE 6. Precision, Recall, and F1-Score.

For instance Figure 6 shows the Support Vector Machine (SVM) showed a slight benefit over the Decision Tree model, a finding that complements the conclusions of Khan et al. [21], where the Intelligent Hybrid Feature Selection method also led to superior performance in sentiment feature extraction. These results add to the growing body of evidence from Aruna et al. [12] and Basha and Rajput [11], both of whom highlighted the positive impact of optimized feature selection on sentiment analysis. Thus, the convergence of our findings with these prior studies confirms the importance of feature selection in enhancing the accuracy and efficiency of machine learning models in complex, high-dimensional data scenarios such as app user review analysis. In summary, we can conclude from this experiment the following points:

1) This study compares the performance of four different models—Naïve Bayes, Deep Learning, Decision Tree, and Support Vector Machine (SVM)—using selected features. The results indicate that the SVM model performs the best in terms of both accuracy and overall effectiveness.
2) The Deep Learning model achieved a relatively high accuracy of 93.2% when feature selection was applied, which demonstrates its potential for handling complex multi-label classification tasks. However, it tends to overfit more than simpler models like Decision Tree, which limits its generalizability.
3) On the other hand, the Naïve Bayes model showed the lowest accuracy at 91.7%, even with feature selection. This suggests that Naïve Bayes has some limitations when it comes to effectively managing multi-label classification tasks compared to the other models.

C. DISCUSSION ON THE IMPLICATIONS OF THE RESEARCH FINDINGS ON FEATURE SELECTION TECHNIQUE AND MACHINE LEARNING MODEL SELECTION

The research findings have several important implications for both feature selection and model selection in the context of machine learning, particularly for multi-label classification tasks. These implications will be discussed in detail in the following items and derived significant details as shown in Table 13.

1) **Critical Role of Feature Selection**: The study demonstrates that feature selection techniques significantly improve the performance of machine learning models. This underscores the importance of investing time and resources into developing robust feature selection techniques. Practitioners should prioritize feature selection as a key step in the machine learning pipeline to enhance model accuracy and efficiency.
2) **Generalizability Across Models**: The findings suggest that effective feature selection techniques can lead to performance improvements across different types of models. This implies that well-engineered features can generalize well, making them valuable for a wide range of machine-learning tasks and models.
3) **Innovation in Feature or Attribute Creation**: The study encourages innovation in creating new features. Techniques such as polynomial features, cross-features, and domain-specific feature extraction can provide additional predictive power. In this study, we have developed a new feature or attribute called `%Difference` on top of standard attributes, which carries significant weight in all feature selection techniques, including information gain, GINI index, and correlation matrix.
4) **Machine Learning Selection**: The ANOVA test results indicate that there is no significant difference in accuracy among the different machine learning models when the right feature selection technique is applied. This suggests that the choice of machine learning model may be less critical than the quality of the features or attributes that are carefully designed and selected.
5) **Data Quality**: The effectiveness of feature selection methods in user review analysis heavily depends on the quality of the underlying data. Clean, well-preprocessed data enhances model outcomes by eliminating noise and ensuring the inclusion of meaningful features (Maskat and Munarko [45] and Ariffin and Tiun [46]). Preprocessing is vital to remove unnecessary elements, such as punctuation, redundant characters, or social media-specific tags, ensuring the dataset is fit for further analysis (Selvaraju and Anbananthen [47] and Kassim et al. [48]). Studies emphasize that thorough normalization and cleaning, especially in datasets containing informal or non-standard text, improve the accuracy of models used for sentiment analysis and other applications (Bakar et al. [49]; Alsaffar and Omar [50]). This highlights that high-quality data is essential to achieving reliable feature selection and optimizing the performance of machine learning models.

V. THREATS TO VALIDITY

In this section, we explain the potential threats to validity that we identified for this study.

1) **Conclusion Validity:** Although the study showed improvements in model performance using feature selection techniques, the p-values from statistical tests were not always significant (e.g., 0.125 for the t-test and 0.052 for ANOVA). This could suggest that some observed improvements may not be statistically reliable, despite practical relevance. The limited number of models (only four) may reduce the statistical power of the analysis, potentially affecting the accuracy of conclusions drawn about performance enhancements. Stratified 10-fold cross-validation, already employed in this study, further ensures robust performance evaluation by minimizing bias due to small datasets.
2) **Internal Validity:** Factor attribute like `%differences` may have affected the results, making it hard to clearly connect feature selection to improved performance. Additionally, the manual selection of features for baseline comparisons introduces subjectivity, which could bias the outcomes. To mitigate this, hyperparameter optimization through an optimized parameter grid was utilized, reducing variability caused by uncontrolled parameters. Automated feature labeling with the proposed score-based zero-shot technique helps mitigate subjective bias introduced by manual selection, thus improving consistency.
3) **Construct Validity:** The study uses performance metrics such as accuracy, precision, and F1-score to assess the model's effectiveness. However, these metrics might not fully reflect the impact of feature selection techniques in multi-label classification. In such cases, small changes in performance across different labels could go unnoticed, making it harder to understand how well the model performs for each specific label. To mitigate this, we proposed score-based zero-shot technique that consists of several key elements.
    a) **Capturing nuanced differences in multi-label classification:** The method calculates the percentage difference between the highest scores across labels, which helps identify subtle variations between categories such as Problem Discovery (PD) and Feature Requests (FR). This ensures that minor but meaningful changes are detected, addressing the limitation where traditional metrics like precision or F1-score might overlook small discrepancies.
    b) **Dynamic label assignment and decision logic:** By determining the primary classification label

TABLE 13. Derived significant details.

No.	Item	Description
1.	Impact of Feature Selection Techniques	The study evaluated various feature selection techniques, including Information Gain, Gini Index, and Correlation Matrix. The findings revealed that the combination of these techniques significantly improved the performance of machine learning models in multi-label classification tasks.
2.	Performance Metrics	The study demonstrated that using a Support Vector Machine (SVM) with the proposed innovative score-based zero-shot technique yielded promising results. The average precision was 96.75%, recall was 69.39%, and the F1-Score was 80.81%.
3.	High-Quality Features	The study emphasized that high-quality features, such as the *percentageDifference* attribute, enhanced performance in multi-label classification tasks, providing valuable insights for practitioners and researchers.
4.	Proposed Techniques	The study proposed a novel feature selection technique combining Information Gain, Gini Index, and Correlation Matrix, as well as a Score-Based Zero-Shot technique for multi-label classification.

based on the highest scores and percentage difference thresholds (diff $\leq 50\%$), our approach offers adaptive label selection. This is more refined than standard binary outputs, enabling better reflection of multi-label behavior and enhancing the interpretability of the classification results.

c) **Supplementary evaluation through score-based decisions:** Incorporating intermediate calculations score (e.g., highest_{sPD}, highest_{sFR}) introduces a complementary perspective that supplements accuracy-based metrics. This provides additional performance insights, thereby enriching the evaluation process.

4) **External Validity:** The results may not generalize well to other domains, as the experiment focuses on app user reviews, which involve specific text characteristics. Additionally, with only four machine learning models tested, the findings may not extend to other model architectures or datasets, limiting broader applicability. To mitigate this, we apply multiple feature selection techniques such as Information Gain, GINI Index, and Correlation Matrix to establishes a foundation for broader applicability.

VI. CONCLUSION

In conclusion, this study has provided valuable insights into the impact of feature selection techniques and the performance of various machine learning models in multi-label classification tasks, particularly in app user review analysis. The findings highlight that, despite a lack of statistical significance, feature selection techniques such as Information Gain (IG), Gini Index (GI), and Correlation Matrix (CM) demonstrate clear practical benefits in improving model accuracy. The proposed score-based zero-shot method with Support Vector Machine (SVM) improves accuracy and highlights the important role of feature selection in optimizing machine learning models.

Although the statistical analysis yielded a p-value slightly higher than the significance threshold, the near-significant results suggest that feature selection may have a more substantial impact than the statistical tests indicate. Improvements in key performance metrics such as precision, recall, and F1-score underscore the effectiveness of applying techniques like IG+GI+CM in multi-label classification. These findings, while not statistically conclusive, provide further evidence of the practical advantages of feature selection, demonstrating its importance in enhancing the performance of machine learning models. Overall, this research reinforces the idea that feature selection is essential in improving model outcomes, even in cases where statistical significance may not be fully evident.

VII. FUTURE DIRECTIONS

The study acknowledges the need for further research to enhance the proposed technique effectiveness and generalizability. Future research directions should include:

1) **Expanding Machine Learning Models and Multiple Domain Datasets:** In the future, it is important to use more machine learning models and work with larger and more diverse datasets. Combining data from multiple domains can help improve the accuracy and performance of these models. This approach will enhance the statistical power of the findings and provide deeper insights into the effects of feature selection techniques on multi-label classification tasks.
2) **Hyperparameter Tuning:** Further research should explore the impact of hyperparameter tuning on model performance. This process plays an important role in reducing confounding factors that may influence the results. By carefully tuning hyperparameters, researchers can achieve more reliable and consistent outcomes, making it easier to see the real impact of different feature selection techniques.
3) **Exploring Additional Evaluation Metrics:** Traditional metrics such as accuracy, precision, and F1-score may not fully capture the nuanced improvements in multi-label classification tasks. In the future, researchers should look into using more evaluation metrics that can give a broader and clearer understanding of how well the model performs.

CONFLICT OF INTEREST

The authors declare no any conflicts of interest.

ACKNOWLEDGMENT

Authors express their sincere gratitude to Universiti Putra Malaysia (UPM) for providing financial support and resources throughout this research.

REFERENCES

[1] W. Maalej and H. Nabil, "Bug report, feature request, or simply praise? On automatically classifying app reviews," in *Proc. IEEE 23rd Int. Requirements Eng. Conf. (RE)*, Aug. 2015, pp. 116–125.
[2] S. Malgaonkar, S. A. Licorish, and B. T. R. Savarimuthu, "Prioritizing user concerns in app reviews—A study of requests for new features, enhancements and bug fixes," *Inf. Softw. Technol.*, vol. 144, Apr. 2022, Art. no. 106798.
[3] P. G. Manek, A. F. Septiyanto, and A. S. Nugroho, "A semantic comparison of feature requirements extraction methods," *IPTEK J. Technol. Sci.*, vol. 32, no. 3, pp. 176–183, Dec. 2021.
[4] M. Haering, C. Stanik, and W. Maalej, "Automatically matching bug reports with related app reviews," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng. (ICSE)*, May 2021, pp. 970–981.
[5] A. F. de Araújo and R. M. Marcacini, "RE-BERT: Automatic extraction of software requirements from app reviews using BERT language model," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 1321–1327.
[6] C. Gao, J. Zeng, M. R. Lyu, and I. King, "Online app review analysis for identifying emerging issues," in *Proc. IEEE/ACM 40th Int. Conf. Softw. Eng. (ICSE)*, May 2018, pp. 48–58.
[7] M. P. S. Golo, A. F. Araújo, R. G. Rossi, and R. M. Marcacini, "Detecting relevant app reviews for software evolution and maintenance through multimodal one-class learning," *Inf. Softw. Technol.*, vol. 151, Nov. 2022, Art. no. 106998.
[8] E. Guzman, M. Ibrahim, and M. Glinz, "Mining Twitter messages for software evolution," in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng. Companion (ICSE-C)*, May 2017, pp. 283–284.
[9] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How can I improve my app? Classifying user reviews for software maintenance and evolution," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep. 2015, pp. 281–290.
[10] Y. M. Aye and S. S. Aung, "Contextual lexicon based sentiment analysis in Myanmar text reviews," in *Proc. 23rd Conf. Oriental COCOSDA Int. Committee Co-Ordination Standardisation Speech Databases Assessment Techn. (O-COCOSDA)*, Nov. 2020, pp. 160–165.
[11] S. M. Basha and D. S. Rajput, "Evaluating the impact of feature selection on overall performance of sentiment analysis," in *Proc. Int. Conf. Inf. Technol.*, Dec. 2017, pp. 96–102.
[12] T. Aruna, K. Asha, G. Divyaraj, and P. K. Pareek, "Feature selection based naïve Bayes algorithm for Twitter sentiment analysis," in *Proc. 4th Int. Conf. Emerging Res. Electronics, Comput. Sci. Technology (ICERECT)*, Dec. 2022, pp. 1–7.
[13] V. N. Thatha, A. S. Babu, and D. Haritha, "An enhanced feature selection for text documents," in *Proc. 3rd Int. Conf. Smart Comput. Inform.*, vol. 2. Springer, 2020, pp. 21–29.
[14] R. R. Mekala, A. Irfan, E. C. Groen, A. Porter, and M. Lindvall, "Classifying user requirements from online feedback in small dataset environments using deep learning," in *Proc. IEEE 29th Int. Requirements Eng. Conf. (RE)*, Sep. 2021, pp. 139–149.
[15] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, "Categorizing feature selection methods for multi-label classification," *Artif. Intell. Rev.*, vol. 49, no. 1, pp. 57–78, Jan. 2018.
[16] N. K. Mishra and P. K. Singh, "FS-MLC: Feature selection for multi-label classification using clustering in feature space," *Inf. Process. Manage.*, vol. 57, no. 4, Jul. 2020, Art. no. 102240.
[17] F. Pourpanah, Y. Shi, C. P. Lim, Q. Hao, and C. J. Tan, "Feature selection based on brain storm optimization for data classification," *Appl. Soft Comput.*, vol. 80, pp. 761–775, Jul. 2019.
[18] K. Koka and S. Fang, "Online review analysis by visual feature selection," in *Proc. IEEE 15th Int. Conf Dependable, Autonomic Secure Comput., 15th Int. Conf Pervasive Intell. Comput., 3rd Int. Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Nov. 2017, pp. 1084–1091.
[19] K. Kim and S. Y. Zzang, "Trigonometric comparison measure: A feature selection method for text categorization," *Data Knowl. Eng.*, vol. 119, pp. 1–21, Jan. 2019.
[20] E. A. K. Zaman, A. Mohamed, and A. Ahmad, "Feature selection for online streaming high-dimensional data: A state-of-the-art review," *Appl. Soft Comput.*, vol. 127, Sep. 2022, Art. no. 109355.
[21] J. Khan, A. Alam, and Y. Lee, "Intelligent hybrid feature selection for textual sentiment classification," *IEEE Access*, vol. 9, pp. 140590–140608, 2021.
[22] K. Kaur and P. Kaur, "Improving BERT model for requirements classification by bidirectional LSTM-CNN deep model," *Comput. Electr. Eng.*, vol. 108, May 2023, Art. no. 108699.
[23] A. A. Abdulhussien, M. F. Nasrudin, S. M. Darwish, and Z. A. A. Alyasseri, "Feature selection method based on quantum inspired genetic algorithm for Arabic signature verification," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 3, pp. 141–156, Mar. 2023.
[24] N. Elhassan, G. Varone, R. Ahmed, M. Gogate, K. Dashtipour, H. Almoamari, M. A. El-Affendi, B. N. Al-Tamimi, F. Albalwy, and A. Hussain, "Arabic sentiment analysis based on word embeddings and deep learning," *Computers*, vol. 12, no. 6, p. 126, Jun. 2023.
[25] J. M. Duarte and L. Berton, "A review of semi-supervised learning for text classification," *Artif. Intell. Rev.*, vol. 56, no. 9, pp. 9401–9469, Sep. 2023.
[26] K. K. Mohammadmosaferi and H. Naderi, "AFIF: Automatically finding important features in community evolution prediction for dynamic social networks," *Comput. Commun.*, vol. 176, pp. 66–80, Aug. 2021.
[27] A. Al-Hawari, H. Najadat, and R. Shatnawi, "Classification of application reviews into software maintenance tasks using data mining techniques," *Softw. Quality J.*, vol. 29, no. 3, pp. 667–703, Sep. 2021.
[28] M. Tubishat, N. Idris, and M. Abushariah, "Explicit aspects extraction in sentiment analysis using optimal rules combination," *Future Gener. Comput. Syst.*, vol. 114, pp. 448–480, Jan. 2021.
[29] Z. Peng, J. Wang, K. He, and M. Tang, "An approach of extracting feature requests from app reviews," in *Proc. 12th Int. Conf. Collaborate Comput., Netw., Appl. Worksharing*. Beijing, China: Springer, Nov. 2017, pp. 312–323.
[30] M. Binkhonain and L. Zhao, "A machine learning approach for hierarchical classification of software requirements," *Mach. Learn. Appl.*, vol. 12, Jun. 2023, Art. no. 100457.
[31] K. Bhatia and A. Sharma, "Sector classification for crowd-based software requirements," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 1312–1320.
[32] RapidMiner. (2024). *Deep Learning—Rapidminer Studio Documentation*. Accessed: Feb. 25, 2025. [Online]. Available: https://docs.rapidminer.com/2024.1/studio/operators/modeling/predictive/neural_nets/deep_learning.html
[33] K. Sridharan and P. Sivakumar, "A systematic review on techniques of feature selection and classification for text mining," *Int. J. Bus. Inf. Syst.*, vol. 28, no. 4, pp. 504–518, 2018.
[34] M. P. S. Bhatia, A. Kumar, and R. Beniwal, "An optimized classification of apps reviews for improving requirement engineering," *Recent Adv. Comput. Sci. Commun.*, vol. 14, no. 5, pp. 1390–1399, Aug. 2021.
[35] M. Lu and P. Liang, "Automatic classification of non-functional requirements from augmented app user reviews," in *Proc. 21st Int. Conf. Eval. Assessment Softw. Eng.*, Jun. 2017, pp. 344–353.
[36] M. A. H. Wadud, M. M. Kabir, M. F. Mridha, M. A. Ali, M. A. Hamid, and M. M. Monowar, "How can we manage offensive text in social media—A text classification approach using LSTM-BOOST," *Int. J. Inf. Manage. Data Insights*, vol. 2, no. 2, Nov. 2022, Art. no. 100095.
[37] T. Wang, P. Liang, and M. Lu, "What aspects do non-functional requirements in app user reviews describe? An exploratory and comparative study," in *Proc. 25th Asia–Pacific Softw. Eng. Conf. (APSEC)*, Dec. 2018, pp. 494–503.
[38] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning–based text classification: A comprehensive review," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 1–40, Apr. 2021.
[39] T. Hirsch and B. Hofer, "Using textual bug reports to predict the fault category of software bugs," *Array*, vol. 15, Sep. 2022, Art. no. 100189.
[40] S. Saleem, M. N. Asim, L. V. Elst, and A. Dengel, "FNReq-net: A hybrid computational framework for functional and non-functional requirements classification," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 8, Sep. 2023, Art. no. 101665.
[41] L. Liu, W. Zhou, and M. Gutierrez, "Effectiveness of predicting tunneling-induced ground settlements using machine learning methods with small datasets," *J. Rock Mech. Geotechnical Eng.*, vol. 14, no. 4, pp. 1028–1041, Aug. 2022.

[42] Y. Zhang, Y. Xin, Q. Li, J. Ma, S. Li, X. Lv, and W. Lv, "Empirical study of seven data mining algorithms on different characteristics of datasets for biomedical classification applications," *Biomed. Eng. OnLine*, vol. 16, no. 1, pp. 1–15, Dec. 2017.
[43] R. Wongso, F. A. Luwinda, B. C. Trisnajaya, O. Rusli, and Rudy, "News article text classification in Indonesian language," *Proc. Comput. Sci.*, vol. 116, pp. 137–143, Jan. 2017.
[44] M. F. R. A. Bakar, N. Idris, L. Shuib, and N. Khamis, "Sentiment analysis of noisy Malay text: State of art, challenges and future work," *IEEE Access*, vol. 8, pp. 24687–24696, 2020.
[45] R. Maskat and Y. Munarko, "A taxonomy of Malay social media text," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, pp. 465–472, Oct. 2019.
[46] S. N. A. Noor Ariffin and S. Tiun, "Part-of-speech tagger for Malay social media texts," *GEMA Online J. Lang. Stud.*, vol. 18, no. 4, pp. 124–142, Nov. 2018.
[47] S. Selvaraju and K. S. Muthu Anbananthen, "Opinion extraction on online Malay text," *Amer. J. Appl. Sci.*, vol. 16, no. 4, pp. 134–142, Apr. 2019.
[48] M. N. Kassim, S. H. M. Jali, M. A. Maarof, A. Zainal, and A. A. Wahab, "Enhanced text stemmer with noisy text normalization for Malay texts," in *Smart Trends in Computing and Communications*. Berlin, Germany: Springer, 2020, pp. 433–444.
[49] M. F. R. A. Bakar, N. Idris, and L. Shuib, "An enhancement of Malay social media text normalization for lexicon-based sentiment analysis," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Nov. 2019, pp. 211–215.
[50] A. Alsaffar and N. Omar, "Integrating a lexicon based approach and K nearest neighbour for Malay sentiment analysis," *J. Comput. Sci.*, vol. 11, no. 4, pp. 639–644, Apr. 2015.

AMRAN SALLEH received the B.Sc. degree from Universiti Malaysia Pahang, in 2006, and the M.Sc. degree in computer science from Universiti Putra Malaysia, in 2015. He is currently pursuing the Ph.D. degree in software engineering with a focus on software requirements and machine learning. His work includes publishing in reputable journals.

MOHD HAFEEZ OSMAN (Member, IEEE) received the B.Sc. and M.Sc. degrees from Universiti Teknologi Malaysia (UTM), Malaysia, and the Ph.D. degree from Leiden University, The Netherlands. He is currently a Senior Lecturer with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. His research interests include reverse engineering and software design.

SA'ADAH HASSAN received the B.Comp.Sc. degree from Universiti Teknologi Malaysia (UTM), the master's degree in software engineering from the University of Malaya (UM), Malaysia, and the Ph.D. degree from the University of Ulster, Northern Ireland, U.K. She is currently an Associate Professor with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Her research interests include software engineering, intelligent systems, and management information systems.

MAR YAH SAID received the B.Sc. degree from Leeds Metropolitan University, the M.Sc. degree from Universiti Teknologi Malaysia (UTM), and the Ph.D. degree from the University of Southampton. She is currently a Senior Lecturer with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Her research interests include software engineering requirements and formal specifications.