Software Quality Journal (2023) 31:1179–1209
https://doi.org/10.1007/s11219-023-09630-8

Machine learning for mHealth apps quality evaluation

An approach based on user feedback analysis

Mariem Haoues[1,2] · Raouia Mokni[3,4] · Asma Sellami[2]

Accepted: 13 April 2023 / Published online: 23 May 2023

Abstract
Mobile apps for healthcare (mHealth apps for short) have been increasingly adapted to help users manage their health or to get healthcare services. User feedback analysis is a pertinent method that can be used to improve the quality of mHealth apps. The objective of this paper is to use supervised machine learning algorithms to evaluate the quality of mHealth apps according to the ISO/IEC 25010 quality model based on user feedback. For this purpose, a total of 1682 user reviews have been collected from 86 mHealth apps provided by Google Play Store. Those reviews have been classified initially into the ISO/IEC 25010 eight quality characteristics, and further into Negative, Positive, and Neutral opinions. This analysis has been performed using machine learning and natural language processing techniques. The best performances were provided by the Stochastic Gradient Descent (SGD) classifier with an accuracy of 82.00% in classifying user reviews according to the ISO/IEC 25010 quality characteristics. Moreover, Support Vector Machine (SVM) classified the collected user reviews into Negative, Positive, and Neutral with an accuracy of 90.50%. Finally, for each quality characteristic, we classified the collected reviews according to the sentiment polarity. The best performance results were obtained for the Usability, Security, and Compatibility quality characteristics using SGD classifier with an accuracy equal to 98.00%, 97.50%, and 96.00%, respectively. The results of this paper will be effective to assist developers in improving the quality of mHealth apps.

Keywords Machine learning · mHealth apps · Natural language processing · User feedback · Sentiment analysis · ISO/IEC 25010 quality model · Quality characteristics

1 Introduction

Software development organizations compete to provide mobile applications (apps)[1] that successfully satisfy user needs. Worldwide, around 2.87 billion people use smartphones, where 47% say they cannot live without their devices Turner (2020). Mobile apps provide

[1] For simplicity, we will use the term “app” for application throughout this paper.

✉ Mariem Haoues
m.haoues@psau.edu.sa

Extended author information available on the last page of the article

interesting services for users; however, their quality is also important. Several quality characteristics should be provided by each mobile app, especially usability. The best way to evaluate the quality of mobile apps from a user perspective is to analyze his feedback.

Users review mobile apps they used or are currently using. User feedback contains usage scenarios, bug reports, and feature requests, that can help apps' developers to accomplish apps maintenance and evolution tasks Panichella et al. (2015). Hence, user feedback can be used by developers to early fix bugs and enhance the new release Maalej et al. (2015). The manual analysis of user reviews is unreasonable. Mobile apps' developers spend an important effort in collecting and analyzing reviews to better satisfy user needs.

Mobile apps are increasingly being adopted in the healthcare industry, by patients as well as medicinal experts. Statistics indicate that over 318,000 mobile apps for healthcare are available in major app stores with more than 270 million people having downloaded a healthcare app Mobius MD (2019). Mobile apps in healthcare are classified into different types such as health & fitness, stress, and diagnosis. Healthcare apps (mHealth apps) mostly provide assistance outside hospitals for patients and can help them manage their daily routine such as measuring vital parameters (e.g., pulse, blood sugar), taking medicine, etc. On the other hand, mHealth apps help healthcare providers conducting virtual visits and gathering data from their patients. For those reasons, healthcare organizations are increasingly adapting mHealth apps to improve the quality of their services. Currently, mHealth apps are used by a considerable number of users, which may lead to a large volume of reviews. Hence, due to the large volume of texts, the manual extraction of relevant information is an impracticable task Messaoud et al. (2019). In fact, manually analyzing user reviews is tedious and time-consuming, especially when looking for valuable reviews Tamjeed (2020).

Since its introduction in 1949 by the Canadian psychologist, Hebb (1949), machine learning algorithms have been increasingly being adopted in different domains (e.g., software engineering, healthcare) due to their problem-solving capacity Alpaydin (2020). Different software development and maintenance activities could be expressed through learning problems and solved by learning algorithms such as effort estimation Pospieszny et al. (2018), requirements classification Zhang and Tsai (2002), and so on. 40% of the United States companies use machine learning to improve sales and marketing, with 76% of them having exceeded their sales targets thanks to the use of machine learning Agrawal (2020). The promising results reported encouraged us to use machine learning algorithms in our study to evaluate mHealth apps' quality based on user feedback.

The quality evaluation of mobile apps, in particular mHealth apps has been investigated recently by many researchers (cf., Al Kilani et al. (2019); Idri et al. (2018); Dewi et al. (2020), etc.). Several techniques have been used for this purpose such as quality metrics (cf., Zulfa et al. (2020); Dewi et al. (2020), etc.), quality assessment questionnaire (cf., Idri et al. (2017, 2018), etc.), and machine learning (cf., Al Kilani et al. (2019); Lu and Liang (2017), etc.). In fact, researchers evaluated the quality of mobile apps according to a set of quality characteristics (e.g., portability, maintainability, performance); however, none of these research studies respected totally the ISO/IEC 25010 quality model (cf., Idri et al. (2017, 2018), etc.). For instance, Idri et al. Idri et al. (2017) considered Operability as a quality characteristic, while Operability is a sub-characteristic of the Usability quality characteristic according to the ISO/IEC 25010 quality model ISO/IEC (2010). In addition, machine learning has been successfully used in previous works (cf., Al Kilani et al. (2019); Araujo et al. (2020), etc.) to classify user reviews into different categories for requirements engineering such as bug reports and enhancement reports. In fact, except for Kilani et al. Al Kilani et al. (2019), none of the previous studies proposed to classify user reviews on mobile apps according to the different ISO/IEC 25010 quality characteristics.

The main objective of this paper is to improve the quality of mental health apps based on user feedback. Hence, the main research question is that we address in this paper is *"How to evaluate the quality of mHealth apps based on user feedback according to ISO/IEC quality model?"*. The main contributions of this work can be summarized as follows:

- Firstly, we collect data from the user feedback on mHealth apps provided by the Google Play Store and apply natural language processing techniques to construct a classification system using machine learning algorithms.
- Secondly, we apply six machine learning algorithms (Random Forest, Decision Tree, Multinomial Naïve Bayes, K-Nearest Neighbors, Support Vector Machine, and Stochastic Gradient Descent) to classify the collected user reviews according to the eight quality characteristics of ISO/IEC 25010 quality model (Functional suitability, Reliability, Performance efficiency, Compatibility, Usability, Security, Maintainability, and Portability) as well as sentiment polarity (Positive, Neutral, and Negative).
- Finally, we conduct a set of experiments using our created dataset, as our proposed model yields the highest performance compared to other machine learning and deep learning models (BERT, RoBERTa, DistilBERT, and DistilBERT ML).

This paper is structured as follows: Sect. 2 presents the background information about ISO/IEC 25010 quality model ISO/IEC (2010), and reviews some related works. Section 3 describes how to use machine learning algorithms in evaluating mHealth apps' quality based on the user feedback (i.e., opinions). Section 4 presents and discusses the results of our conducted experiments. Section 5 discusses the obtained results in this paper. In Sect. 6, we highlight several threats to its validity. Finally, Sect. 7 summarizes this work with a set of future work directions.

2 Background and literature review

This section gives background information about the ISO/IEC 25010 quality model ISO/IEC (2010) and reviews some works studying the use of machine learning algorithms to evaluate the quality of mHealth apps using user feedback.

2.1 Software product quality: ISO/IEC 25010

ISO/IEC 25010 quality model, part of the SQuaRE (Software product Quality Requirements and Evaluation) series, presents a standardized way of defining and quantifying software/service quality characteristics. The ISO/IEC 25010 quality model is a *"set of characteristics, sub-characteristics, quality measures, quality measure elements and relationships between them"* ISO/IEC (2010). This model is composed of eight characteristics and 31 sub-characteristics that are related to the static properties of the software and dynamic properties of the computer system.

Compared to other quality models, ISO/IEC 25010 is more comprehensive and complete Herrera et al. (2010). For these reasons, we selected this model in this paper; however, we will focus only on the first level of the ISO/IEC 25010 quality model (i.e., quality characteristics level). This level includes the following eight quality characteristics: Functional suitability, Reliability, Performance efficiency, Compatibility, Usability, Security, Maintainability and Portability.

2.2 Related work

In this section, we review some related work that evaluated the quality of mobile apps, in particular mHealth apps. Then, we survey some previous work that used the user feedback analysis to improve mobile apps. Finally, we provide a discussion of the state-of-art.

2.2.1 Quality evaluation of mobile apps

The evaluation of mHealth apps quality has been investigated recently in several research studies (cf., Idri et al. (2017, 2018), etc.). Some of those studies will be detailed below.

Zulfa et al. (2020) proposed to evaluate the portability of the MyITS mobile app based on its three sub-characteristics: Adaptability, Installability, and Replaceability using six metrics, as provided by the ISO/IEC (2016). The weight results calculated from these six metrics on the three sub-characteristics reached maximum results of 1.0. The calculated weight for the adaptability sub-characteristic achieved 7.89, whereas the calculated weight for the installability sub-characteristic achieved 2. For the replaceability sub-characteristic, there is no calculated weight result since all attributes cannot be computed in quality. The obtained results proved that the MyITS mobile app can work appropriately on a variety of environments (e.g., Android, IOS). For the same mobile app, Dewi et al. (2020) evaluated and measured the maintainability quality characteristic based on its four sub-characteristics: Analysability, Modularity, Reusability, and Testability using 10 metrics, as provided by the ISO/IEC (2016). The results of this study showed that the maintainability of the MyITS mobile apps is good. The weight results calculated for the four sub-characteristics reached maximum results of myITS Lecturer at 2.670 and myITS Student at 2.083. The best weight value obtained for the Analysability sub-characteristic achieved 1.0. For the Modularity sub-characteristic, the best-achieved weight value is 0.75. For the Reusability sub-characteristic, the best-obtained weight value is 0.5. Finally, the best-obtained weight value for the Testability quality sub-characteristic is 0.67. It must be noted that this study did not include the Modifiability sub-characteristic.

Falih and Firdaus (2019) investigated the evaluation of mobile hybrid apps quality based on the ISO/IEC 25010 quality standard using three quality characteristics: Performance efficiency, Functional suitability, and Portability. The Functional suitability characteristic is evaluated according to the Functional implementation completeness and the Functional implementation coverage metrics. The Performance efficiency characteristic is evaluated according to the CPU usage (%), Memory usage (mb), API device execution time (ms), Screen first loading time (ms), and Screen resume loading time (ms) metrics, whereas the Portability quality characteristic is assessed using the Plugin compatibility and the Number of supported platform metrics. The authors used three case studies to empirically assess their proposed method, which are RocketChat, Fresh Food Finder, and Property Cross apps. The results obtained in this paper showed that, in terms of Functional suitability characteristic, both Fresh Food Finder and Property Cross apps provided good values, whereas for the Performance efficiency characteristic, both RocketChat and Property Cross apps provide a better response time. Finally, for the Portability characteristic, Fresh Food Finder and Property Cross apps give the best results. Hence, concerning the three selected quality characteristics, PropertyCross is better than Fresh Food Finder followed by the RocketChat app.

On the other hand, several researchers investigated the quality evaluation of mHealth apps. For example, Idri et al. (2017) evaluated the software quality of mobile Personal

Health Records (mPHRs) for pregnancy monitoring based on the ISO/IEC 25010 quality standard ISO/IEC (2010). This study selected 17 mPHRs apps available for iOS and Android users from Apple App and Google Play stores, respectively. Their evaluation is based on a quality assessment questionnaire that covers four selected External quality characteristics: Functional suitability, Operability, Performance Efficiency, and Reliability. For each quality characteristic, a set of questions has been proposed depending on the number of sub-characteristics. Each selected app is then evaluated according to the 5-interval scale classification (1–1.5: Very low, 1.6–2.5: Low, 2.6–3.5: Moderate, 3.6–4.5: High, and 4.6–5: Very high). This study showed that the majority of the selected apps offered the Functional suitability (satisfied by 16 from 17: 94.11%) and Reliability (satisfied by 17 from 17: 100%) quality characteristics more than the Operability (satisfied by 14 from 17: 82.35%) and Performance efficiency (satisfied by 7 from 17: 41.17%) quality characteristics. Moreover, this study used four classifiers (Iterative Dichotomiser 3, C4.5, K-nearest neighbors, and Naïve Bayes) to predict the quality in-use (i.e., user ratings) from the external quality of the mPHR apps. Among the 17 selected apps, 14 apps that include the user ratings, have been used in the classification. Each app is presented with the median scores from the selected four quality characteristics and described by its user rating (Moderate, High, and Very High). The experiment evaluation that has been conducted using a 2-fold cross-validation model showed that the K-nearest neighbors achieved the highest mean accuracy rate, followed by C4.5, Naïve Bayes, and Iterative Dichotomiser 3. This study did not provide the accuracy measures for the selected classifiers.

Idri et al. (2018) used a quality assessment questionnaire to evaluate the requirements provided by 30 gamified blood donation apps concerning eight quality characteristics (Functional suitability, Reliability, Performance efficiency, Operability, Security, Compatibility, Maintainability, and Transferability). Then, each selected app has been evaluated according to a set of questions. According to its score, a gamified blood donation app is classified into five groups: Very high if the app's score ∈ [0.90, 1.00], High if the app's score ∈ [0.7, 0.89], Moderate if the app's score ∈ [0.4, 0.69], Low if the app's score ∈ [0.2, 0.39] and Very low if the app's score ∈ [0, 0.19]. The results of this paper showed that the majority of the selected apps satisfied the Functional suitability with 100%, Operability with 91%, Performance efficiency with 86%, and Reliability with 84%.

Davalbhakta et al. (2020) assessed the quality of the mobile apps currently utilized for COVID-19, using the Mobile Application Rating Scale (MARS) for overall Engagement, Functionality, Aesthetics, and Information sub-scales. This study selected 63 apps from Apple app and Google Play stores. The authors conducted their evaluation according to the app continent. The obtained results showed that apps from Asia are rated higher in functionality sub-scale (mean = 0.54; 95%), while the UK (8 of 17 from Europe) and North American apps together are rated higher in information sub-scale (mean = 0.6; 95%). Regarding the Aesthetics, engagement sub-scales, they did not vary between the western and Asian Apps. Generally, this study showed that COVID-19 mobile apps satisfied the functionality dimension with 91.87%, followed by the aesthetics dimension with 77.94%, then the information dimension with 72.58%, and finally the engagement with 64.12%.

Table 1 presents a summary of the research studies that evaluated the quality of mobile apps, including mHealth apps. As illustrated in this table, several researchers investigated the use of the ISO/IEC 25010 quality model to evaluate the mHealth apps. The majority of the research studies in Table 1 used the quality metrics (e.g., Zulfa et al. (2020); Dewi et al. (2020)) or a quality assessment questionnaire (e.g., Idri et al. (2017, 2018)) to evaluate mobile apps according to a set of quality characteristics (e.g., Functional suitability, Maintainability, Performance efficiency). Moreover, some of the above presented research

Table 1 Summary of the research studies that evaluated the quality of mobile apps

Proposal	Source	Quality characteristic	Method	Results
Idri et al. (2017)	Personal Health Records for pregnancy monitoring mobile apps	Functional suitability, Operability, Performance efficiency and Reliability	Quality assessment questionnaire	Functional suitability (94.11%), Reliability (100%), Operability (82.35%) and Performance efficiency (41.17%)
			Iterative Dichotomiser 3, C4.5, K-nearest neighbors and Naïve Bayes	Accuracy rate by K-NN > C4.5 > Naïve Bayes > Iterative Dichotomiser 3
Idri et al. (2018)	Blood donation mobile apps	Functional suitability, Reliability, Performance efficiency, Operability, Security, Compatibility, Maintainability, and Transferability	Quality assessment questionnaire	Functional suitability (100%), Operability (91%), Performance efficiency (86%), and Reliability (84%)
Davalbhakta et al. (2020)	COVID-19 mobile apps	Engagement, Functionality, Aesthetics, and Information	Quality assessment on using the MARS scale	Engagement (64.12%), Functionality (91.87%), Aesthetics (77.94%), and Information (72.58%)
Zulfa et al. (2020)	MyITS mobile app	Portability	Questionnaire, experiments, and metrics	Portability (1.0), Adaptability (7.89), Installability (2.00), and Replaceability (n.a)
Dewi et al. (2020)	MyITS mobile app	Maintainability	Metrics	Maintainability (2.67), Analysability (1.0), Modularity (0.5), Reusability (0.5), and Testability (0.67)
Falih and Firdaus (2019)	Mobile hybrid apps (FoodFinder, RocketChat, and PropertyCross)	Performance, Functionality and Portability	Metrics	PropertyCross > Fresh Food Finder > RocketChat

studies did not respect completely the ISO/IEC 25010 quality model (cf., Idri et al. (2017, 2018), etc.). For example, Idri et al. (2017) considered Operability as a quality characteristic. While Operability is a sub-characteristic of the Usability quality characteristic according to the ISO/IEC 25010 quality model ISO/IEC (2010).

2.2.2 User reviews classification for mobile apps improvement

Reviewing user feedback is a pertinent solution to improve the apps according to the user needs. For this purpose, many researchers proposed to evaluate and analyze the user reviews to improve the mobile apps. For instance, Panichella et al. (2015) presented a taxonomy to classify mobile app reviews relevant to software maintenance (e.g., problem discovery, information seeking). In this paper, the authors collected 32,210 reviews from seven apps such as AngryBirds, Dropbox, and Evernote apps from Apple's App Store and TripAdvisor, PicsArt, Pinterest, and Whatsapp from the Google Play store. The authors applied natural language processing techniques, text and sentiment analysis with five selected machine learning classifiers (alternating decision tree, logistic regression, naive Bayes, support vector machine, and j48) to classify the collected app reviews into the identified categories. The best results have been provided by the j48 classifier with 75.20%, 74.20%, and 72.00% for respectively Precision, Recall, and F1-score measures.

Guzman et al. (2015) suggested a taxonomy for classifying app reviews relevant to software evolution. The taxonomy includes seven categories such as Bug report, Feature strength, Feature shortcoming, User request, Praise, Complaint and Usage scenario. This study collected 4550 reviews from seven mobile apps such as AngryBirds, Dropbox and Evernote apps from Apple App store and TripAdvisor, PicsArt, Pinterest and Whatsapp from Google Play store. To evaluate the performance of their proposed system, the authors used supervised machine learning algorithms (Naive Bayes, support vector machine, Logistic Regression and Neural Network) to classify user reviews into the identified categories. The best results have been provided by the Neural Network classifier with 74.00%, 59.00% and 64.00% for respectively the averages of Precision, Recall, and F-measure measures. The obtained results showed that when fusing the predictions of Logistic Regression and Neural Networks classifiers, the performance of Precision is still the same compared to the Neural Network performance; however, the recall performance is improved to 63.00%.

Al-Hawari et al. (2020) proposed an Associative Classification approach for Review Mining (ACRM) to classify user reviews into four maintenance tasks such as information giving, information seeking, problem discovery, and feature requests. They used the natural language pre-processing and text analysis techniques in the data pre-processing phase and applied several machine learning classifiers. In this study, the authors tested their proposed system using two datasets: Pan dataset and Maalej dataset that have been provided by Panichella et al. (2015) and Maalej et al. (2016), respectively. To evaluate the performance of their proposed method (ACRM), the authors used machine learning classifiers such as decision tree, naïve Bayes (NB), k-nearest neighbor (KNN), Gradient Boosting Trees (GBT), Classification based on Multiple Association Rules (CMAR), random forest (RF), support vector machine (SVM), and AC algorithms. The obtained results showed that the best averages of the accuracy performances have been achieved by GBT classifier with 79.00% and 80.00% over respectively Pan and Maalej datasets.

Aslam et al. (2020) proposed an approach for the classification of app reviews based on deep learning model into four categories: bug report, enhancement reports, user experiences, and ratings. The dataset used in this study has been provided by Maalej et al. (2016).

The proposed approach is evaluated using machine learning classifiers such as NB, Multinomial Naïve Bayes (MNB), DT, SVM, Convolutional Neural Network (CNN), and Long Short Term Memory (LSTM) techniques. The obtained results showed that the best performances are provided by the CNN model with respectively 95.49%, 93.94%, and 94.71% for the averages of Precision, Recall, and F1-score measures.

Finally, Kilani et al. Al Kilani et al. (2019) investigated the use of machine learning and natural language processing techniques to classify mHealth app reviews into initially: bugs, new features, and sentimental. Bugs are further classified into the following: general bug, usability bug, security bug, and performance bug. Same for the sentimental reviews, they are further classified positive, negative, and neutral. The proposed model in this paper includes four phases: data collection, data labeling, feature extraction, and data classification. In this study, the authors evaluated their model over 7500 reviews of 10 different health-related mobile apps using three supervised machine learning classifiers: MNB, RF and SVM. The collected reviews have been annotated manually by experts. The best performances have been provided by the MNB classifier with an F1-score of 72.00%, 52.00%, 90.00%, 86.00%, 12.00%, 11.00% and 21.00% for respectively bugs, new features, sentimental, general bug, security, performance, and usability classes. The averages of Precision, Recall, and F1-score are respectively 74.00%, 72.00%, and 73.00% for the first level classification (bugs, new features, and sentimental).

Table 2 presents a summary of the research studies that proposed to classify user reviews to improve the quality of mobile apps. As illustrated in this table, machine learning algorithms have been successfully used for this purpose. Researchers classified user reviews into different categories with respect to the maintenance phase such as bug reports and enhancement reports. Those categories can include both functional requirements and non-functional requirements; however, the distinction between the requirements types will be beneficial for the developers. In fact, those categories are restricted to the maintenance phase; however, the app quality must be kept during all the software life cycle phases. Moreover, except for Al Kilani et al. (2019), none of the previous studies proposed to classify user reviews according to the different quality characteristics.

2.2.3 Discussion

The quality of mobile apps is increasingly being investigated to increase user's satisfaction. User feedback provides information that express the users' opinions towards a specific mobile app. Several methods could be used to extract the user reviews since they help identify the app's issues and enhance its quality. Moreover, other users consider reviews a reliable source of information. In fact, reading negative feedback could alienate potential users from trying the app.

As illustrated in Table 1, several researchers investigated the use of ISO/IEC 25010 quality model to evaluate the quality of mHealth apps based on the quality metrics (cf., Zulfa et al. (2020); Dewi et al. (2020), etc.) or a quality assessment questionnaire (cf., Idri et al. (2017, 2018), etc.) according to a set of quality characteristics (e.g., Functional suitability, Maintainability, Performance efficiency). However, some of these studies did not use all the ISO/IEC 25010 quality model characteristics (cf., Idri et al. (2017, 2018), etc.). In addition, some researchers used the quality assessment questionnaire, which is very time-consuming and cannot capture perceptions and visualizations in real time.

On the other hand, in Table 2, several researchers (cf., Panichella et al. (2015); Aslam et al. (2020), etc.) investigated the user reviews classification into different categories

Table 2 Summary of the research studies that classified user reviews for mobile apps improvement

Proposal	Source	Method	Categories	Results (%)		
				Precision	Recall	F1-score
Panichella et al. (2015)	32 210 reviews	j48 classifier	Feature Request	70.40	22.50	34.10
			Problem Discovery	87.50	77.60	82.20
			Information Seeking	71.20	68.40	69.80
			Information Giving	68.00	90.40	77.60
			Weighted Avg	75.20	74.20	72.00
Guzman et al. (2015)	4550 reviews	Neural Network	Bug report	83.00	75.00	79.00
			Complaint	45.00	8.00	14.00
			User request	71.00	39.00	50.00
			Feature shortcoming	74.00	75.00	75.00
			Feature strength	70.00	50.00	59.00
			Noise	69.00	75.00	72.00
			Praise	76.00	73.00	74.00
			Usage scenario	73.00	27.00	39.00
Al-Hawari et al. (2020)	1390 reviews from	CMAR	Feature request	92.00	36.00	-
	Panichella et al. (2015)	ACRM	Information giving reviews	82.20	77.40	-
		CMAR	Problem discovery	94.66	60.00	
		ACRM	Information seeking	91.00	66.00	-
	3691 reviews from	ACRM	Problem discovery	77.78	77.78	-
	Maalej et al. (2016)	ACRM	Rating	82.97	75.40	-
		GBT	Feature request	81.06	54.36	-
		GBT	User experiences	89.80	70.23	-
Aslam et al. (2020)	4400 reviews	CNN	Bug report	94.69	94.02	94.35
	Maalej et al. (2016)		Enhancement reports	95.71	93.89	94.79
			User experiences	95.76	93.94	94.84

Table 2 (continued)

Proposal	Source	Method	Categories	Results (%)		
				Precision	Recall	F1-score
			Ratings	95.78	93.91	94.84
			Average	95.49	93.94	94.71
Al Kilani et al. (2019)	7500 reviews	Naive Bayes	Bug	68.00	78.00	72.00
		Multinomial	New Feature	52.00	52.00	52.00
			Sentimental	85.00	75.00	90.00
			General Bug	84.00	89.00	86.00
			Security	8.00	27.00	12.00
			Performance	12.00	9.00	11.00
			Usability	36.00	14.00	21.00
			Average	74.00	72.00	73.00

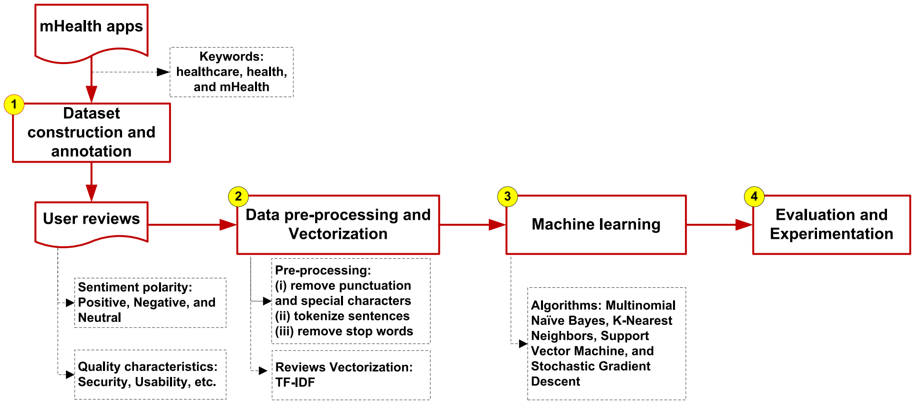

mHealth apps
Keywords: healthcare, health, and mHealth
1
Dataset construction and annotation
User reviews
Sentiment polarity: Positive, Negative, and Neutral
Quality characteristics: Security, Usability, etc.
2
Data pre-processing and Vectorization
Pre-processing: (i) remove punctuation and special characters (ii) tokenize sentences (iii) remove stop words
Reviews Vectorization: TF-IDF
3
Machine learning
Algorithms: Multinomial Naïve Bayes, K-Nearest Neighbors, Support Vector Machine, and Stochastic Gradient Descent
4
Evaluation and Experimentation

Fig. 1 The proposed approach to evaluate mHealth apps quality

within only the maintenance phase classes (e.g., bug reports, enhancement reports); however, apps quality is important for the developers as well as users throughout the software life cycle phases.

Compared to the state-of-art and the previous approaches that focused on the mHealth apps quality evaluation, our approach proposed in this paper used the user feedback analysis based on several supervised machine learning algorithms. Moreover, this paper adapts the ISO/IEC 25010 quality model ISO/IEC (2010), and hence used all the identified quality characteristics by this model. This analysis will help developers evaluate the quality of mHealth apps from the user's perspectives.

3 Research design

This section presents a precise description of the applied method in the evaluation of mHealth apps quality evaluation based on user feedback.

3.1 Machine learning for mHealth apps user feedback analysis

The main objective of this paper is the evaluation of mHealth apps quality based on user feedback using machine learning algorithms. Thus, user reviews will be collected from a set of selected mHealth apps. Then, we proposed to classify firstly each review according to the eight ISO/IEC 25010 quality characteristics (Functional suitability, compatibility, performance, portability, reliability, security, maintainability and usability) ISO/IEC (2010). Furthermore, the collected reviews are classified into positive, negative or neutral (sentiment polarity).

Figure 1 presents the main four steps followed in this research:

- **Step 1 — Dataset construction and annotation:** In order to create our dataset, we initially searched for the mHealth apps using the keywords "healthcare", "health", and "mHealth" available on Google Play store. Then, we applied a set of exclusion criteria to select the most relevant mHealth apps for this research. Thereafter, we used the Appbot tool Appbot (2021) to extract the recent reviews from the selected mHealth apps. This step will be described below in Sect. 3.2. Then, the collected user reviews have been cleaned by removing reviews written in a language different to English, duplicated reviews, and reviews provided non-relevant information. After that, we annotated the collected user reviews and classified them firstly according to the ISO/IEC 25010 quality characteristics ISO/IEC (2010). Then, the reviews are further classified into positive, negative or neutral opinions (sentiment polarity). In fact, this classification is based on the classification provided by Appbot tool Appbot (2021) and the authors' experiences. More details about this step are provided in Sect. 3.3.

- **Step 2 — Data pre-processing:** In the pre-processing of data, we adapted the natural language processing techniques (e.g., tokenize sentences, removing stop words). More details about this step are provided in Sect. 3.4. Furthermore, we applied the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer. More details about this step are provided in Sect. 3.5.
- **Step 3 — Machine learning:** We applied six machine learning algorithms (Random Forest, Decision Tree, Multinomial Naïve Bayes, K-Nearest Neighbors, Support Vector Machine, and Stochastic Gradient Descent) to classify the reviews according to the ISO/IEC 25010 quality characteristics ISO/IEC (2010) firstly, and secondly into sentiment polarity (positive, negative or neutral). More details about this step are provided in Sect. 3.6.
- **Step 4 — Evaluation and experimentation:** To evaluate the selected machine learning algorithms, we used: Accuracy, Precision, Recall, and F1-score. We also conducted a set of experiments. More details about this step are provided in Sect. 4.

3.2 Dataset construction and data cleaning

As much as we know, there is no annotated dataset suitable for our research study. For this reason, we needed to prepare the dataset by ourselves. To collect the most appropriate mHealth apps for this study, we conducted a search on Google Play store using the keywords "health", "mhealth" and "healthcare". A total of 341 apps are provided by Google Play store, where 50 apps have been collected using the keyword "health", 46 apps have been collected using the keyword "healthcare", and 231 apps have been collected using the keyword "mhealth". A total of 14 apps are duplicated.

The 327 collected apps are available in many languages (e.g., English, French) and classified into different categories (e.g., Medical, Health & Fitness, News & Magazines).

To ensure the good quality of the reviews to be collected, we applied the following four Exclusion Criteria (EC):

- **EC1:** Exclude apps that are not in the Health & Fitness or Medical categories in Google Play store. The objective of this exclusion criteria is to keep only apps that are developed mainly for healthcare purpose. By applying EC1, a total of 43 apps have been excluded (e.g., Leap mHealth, WWE SuperCard - Collection de cartes multijoueur). Therefore, the number of the selected apps after employing EC1 is 284.
- **EC2:** Exclude mobile apps without English interfaces. The objective of this exclusion criteria is to ensure that the reviews to be collected later are written in English. By applying EC2, a total of three apps have been excluded (e.g., mHealth, CardioApp - Risco Cardiovascular Perioperatório). Therefore, the number of the selected apps after employing EC2 is 281.

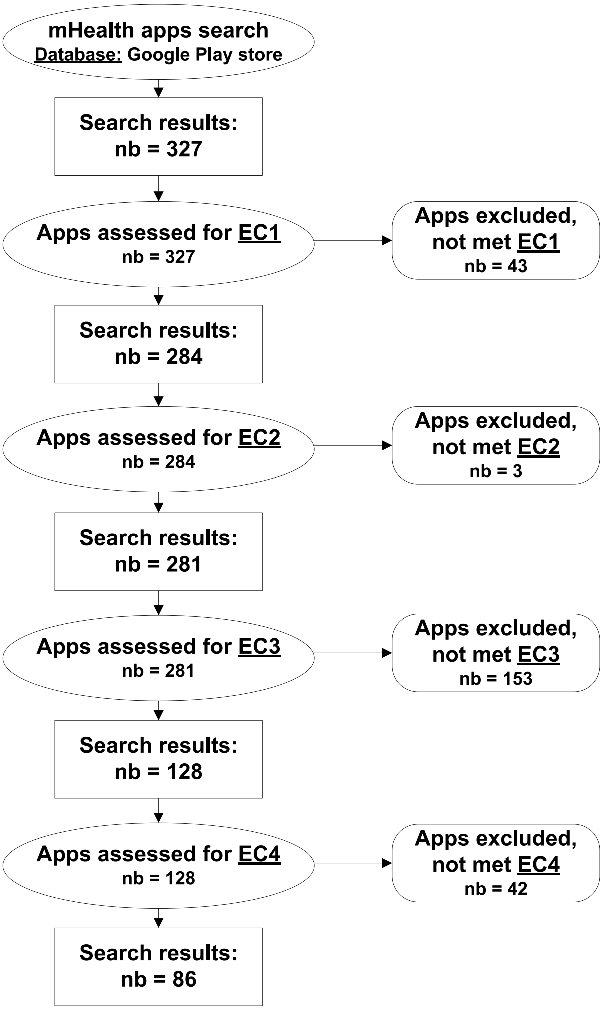

mHealth apps search
Database: Google Play store
Search results: nb = 327
Apps assessed for EC1 nb = 327
Apps excluded, not met EC1 nb = 43
Search results: nb = 284
Apps assessed for EC2 nb = 284
Apps excluded, not met EC2 nb = 3
Search results: nb = 281
Apps assessed for EC3 nb = 281
Apps excluded, not met EC3 nb = 153
Search results: nb = 128
Apps assessed for EC4 nb = 128
Apps excluded, not met EC4 nb = 42
Search results: nb = 86

Fig. 2 Representation of the mHealth apps selection process

- **EC3:** Exclude apps with the last maintenance was before 1^{st} January 2021. The purpose of these exclusion criteria is to select the last updated apps. By applying EC3, a total of 153 apps have been excluded (e.g., Huawei Health, Google Health Studies). Therefore, the number of the selected apps after employing EC3 is 128.
- **EC4:** Exclude mobile apps with a very limited number of reviews. The purpose of this exclusion criteria is to collect an interesting number of reviews for each app. By applying EC4, a total of 42 apps have been excluded (e.g., Health Tracker, Spectrum Health App). Therefore, the number of the selected apps after employing EC4 is 86.

Figure 2 represents the mHealth apps selection process and the number of the excluded and kept apps for each exclusion criteria.

In the collection of the user reviews from the selected 86 mHealth apps, we used the Appbot tool Appbot (2021) that is used in the App review & rating analysis for mobile

Table 3 A user feedback that affects different quality characteristics

User feedback	Terrible app, slow, keeps crashing and it is unstable. Passwords have been compromised by hackers once already.
Quality characteristics	Terrible app, slow, keeps crashing and it is unstable. **[Performance]**
	Passwords have been compromised by hackers once already. **[Security]**
User opinion	**[Negative]**

teams. The initial number of the user reviews extracted from mHealth apps is 2980. The collected reviews can be classified into different categories. Some reviews provide the general opinion of the user about the mobile app (e.g., "I love it!", "I don't like this app"). Other reviews may expose technical problem related to the app (e.g., The button in the page don't work), others ask for the addition of new functionality (e.g., "The app shall provide the functionality to retrieve the current location of the user"), while others propose to improve the app quality (e.g., "let's try another method to login"). In this paper, we will focus on relevant reviews that criticize or acknowledge mobile apps' quality or suggest some improvement (i.e., features). To select the most appropriate reviews to be used in this study, we excluded: eight duplicated reviews, 21 multi-language reviews (English and others), and 1270 irrelevant reviews (general opinion reviews).

3.3 Data annotation

A user review, expressed in natural language, may include different sentences, where each sentence affects a specific quality characteristic. Hence, each sentence must be evaluated and classified independently. Table 3 gives an example of user feedback that provides a Negative opinion on two quality characteristics (Performance and Security).

In this paper, data annotation was a challenging task. Although some reviews have been already annotated by the Appbot tool Appbot (2021), we decided to do this task manually with the help of an expert in software engineering. Data annotation is done by each author individually.

When there is a disagreement between the authors in the manual annotation, the final decision was made by the software engineering expert. The collected reviews are classified according to the ISO/IEC 25010 quality characteristics ISO/IEC (2010). Each review is labeled into its most suitable class (e.g., security, usability, performance); however, the decision regarding a single review could differ from one expert to another. Hence, the authors discussed the review classification carefully to guaranty their correctness.

Then, each review is annotated as positive, negative, or neutral. Each user review is rated by Appbot tool Appbot (2021) according to a scale of 1 to 5. The annotation of the selected user reviews into positive, negative, or neutral is done concerning the following rules:

- User reviews ranked 1 or 2 are Negative. The total number of Negative reviews is 754.
- User reviews ranked 3 are Neutral. The total number of Neutral reviews is 39.
- User reviews ranked 4 or 5 are Positive. The total number of Positive reviews is 888.

Table 4 presents the total number of user reviews for each quality characteristic and their classification into Positive, Negative, and Neutral classes. As provided in this

Table 4 User review dataset corresponds to eight quality characteristics from the ISO/IEC 25010

Quality characteristics	Number of reviews			
	Positive	Negative	Neutral	Total
Functional Suitability	332	229	28	589
Compatibility	27	94	2	123
Performance	78	80	1	159
Reliability	3	29	0	32
Portability	1	3	2	6
Security	19	177	2	198
Maintainability	11	66	1	78
Usability	417	76	3	496
Total	888	754	39	1681

table, the total number of the user reviews that will be used in this research is 1681, where 589 are Functional Suitability reviews, 123 are Compatibility reviews, 159 are Performance reviews, 32 are Reliability reviews, 6 are Portability reviews, 198 are Security reviews, 78 are Maintainability reviews, and 496 are Usability reviews, as illustrated in Fig. 3. Furthermore, 888 are Positive reviews, 754 are Negative reviews and 39 are Neutral reviews, as illustrated in Fig. 4.

Table 5 revealed some user reviews collected from the selected mHealth apps corresponding to some quality characteristics (Usability, Security, and Functional suitability).

3.4 Data pre-processing

When using the machine learning classifiers, the collected user reviews need to be preprocessed. Data pre-processing is a crucial step. It includes the following tasks: (i) remove punctuation and special characters, (ii) tokenize sentences (i.e., split reviews into tokens), and (iii) remove stop words.

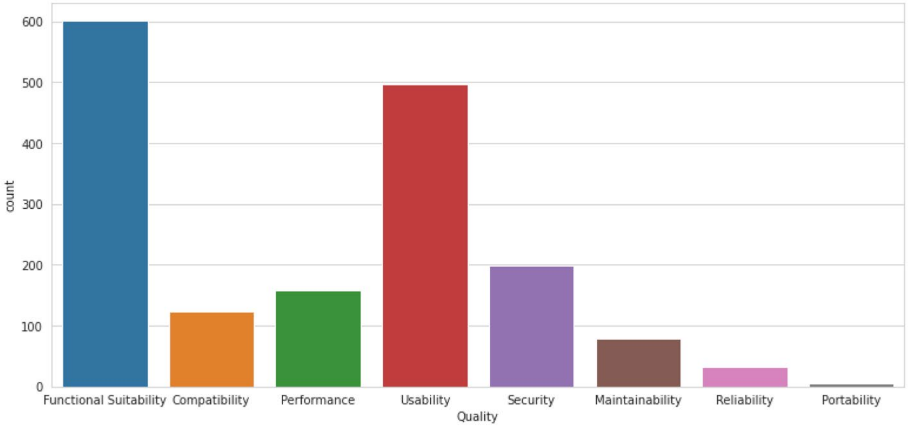


Fig. 3 User reviews number according to the quality characteristics

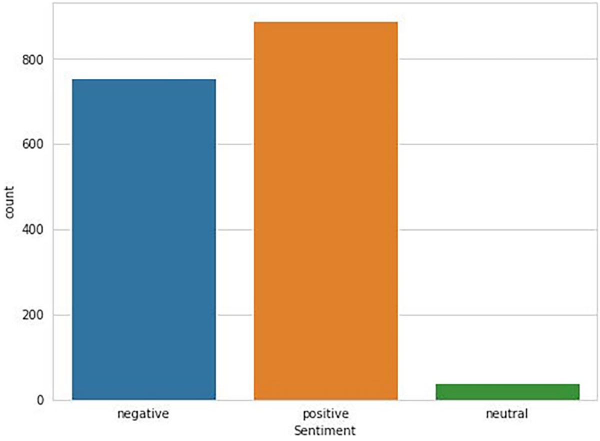


Fig. 4 The user review sentiment classes distribution

3.5 Data vectorization

In the data vectorization, we extracted unique clean tokens from the collected reviews. Then, we extracted Bigrams and applied the Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer to assign weight for each term Medina and Ramon (2015).

3.6 Machine learning algorithms

In this section, we present the implementation of six machine learning algorithms to firstly classify reviews with respect to the ISO/IEC 25010 quality characteristics and further more into Negative, Neutral or Positive opinions.

Among the different machine learning classifiers used in the literature, we selected the six classifiers: SVM, KNN, MNB, RF, DT, and SGD.

3.6.1 Support vector machine

Support Vector Machine (SVM) is a very popular and trendy supervised machine learning algorithm proposed by Vapnik in 1995 Vapnik (2013). SVM showed a promising classification performance with a justified dataset size. It aims to categorize data by finding the decision boundary (hyperplane, surface) that best separates the classes by maximizing the margin distance between nearest data-points that belong to different classes. Several kernel functions can be used (i.e., linear, polynomial, radial basis, and sigmoid) in order to obtain optimal hyperplane (e.g., line) which maximizes the margin distance that separates points that belong to each class. In our study, after several experiments, we used Support vector classifier (SVC) with linear SVM kernel function and we tested several hyper-parameters such as the penalty factor $C \in \{1, 10, 100\}$. Parameters that are the best ($C = 10$) will be chosen empirically.

Table 5 Some user reviews collected from the selected mHealth apps

User reviews	Quality characteristic	Sentiment Polarity
Can you change km into miles via the app?	Functional Suitability	Neutral
Very simple and easy for me	Usability	Positive
This is just an aweful app. I try to login, username/password incorrect. I press forgot username/password. They say that an email was sent to me. I got nothing. They reset my account. Way unsecure and less headache to deal with	Security	Negative
Terrible app, slow, keeps crashing and it is unstable	Performance	Negative
Very good app compared to others I've tried and steps always similar to hubby's who has an apple watch	Compatibility	Positive
New upgrades are not always a good thing. What my old phone had on it was just fine and even better in some cases. I do not like this new version. Wish I had my old Samsung health info back	Maintainability	Negative
Now it's not working at all...not showing any steps	Reliability	Negative
Chat is broken and throwing error messages on Android 10 in a foldable device	Portability	Negative

Table 6 Used parameter setting of the selected machine learning classifiers

Classifiers	Parameters/Values
SVM	Kernel function: Linear Function; C = 10
KNN	K=3
MNB	default (alpha = 1.0)
SGD	loss function = “hinge”, the penalty (regularization term) = “l2”, Number of iterations: max_iter = 5

3.6.2 Multinomial Naïve Bayes

Multinomial Naïve Bayes (MNB) is an approach of naïve bayes algorithms that is one of the most popular supervised machine learning algorithm. It tends to perform very well and achieve show significant results. MNB is a classification technique that makes probabilistic prediction of the class label given some observed features based on bayes theorem Ren et al. (2009). This algorithm assumes that attributes are independent even if they are related. It measures conditional probability of two or more events by calculating the occurring probability for each individual event Singh et al. (2019), this is why it is called naive. In this study, we used the default parameters (alpha=1.0).

3.6.3 Stochastic gradient descent

Stochastic Gradient Descent (SGD) is an approach of Gradient descent. It is an iterative optimization algorithm used to optimize the model function through minimizing the cost function. SGD classifier tends to find linear model parameters where the loss function is minimum by moving iteratively to minimum direction. SGD needs several hyper-parameters (e.g., loss function, regularization parameter and number of iterations). In our study, after several experiments, we used the following parameters: the loss function=“hinge”, the penalty (regularization term) =“l2”, and the number of iterations: max_iter=5.

3.6.4 K-Nearest neighbors

The K-Nearest Neighbors (KNN) classifier is a supervised machine learning algorithm that consists of calculating the Euclidean Distance between two feature vectors and the similarity between them. In our empirical test, for KNN classifier, we used K equal to 3, after some empirical tests of the value of K $\{1, 3, 5\}$.

The summary of the used parameters setting of each machine learning classifier is shown in Table 6.

4 Experiments results and discussion

In this section, we present the results of our conducted experimentation. All those experiments have been evaluated on an NVIDIA GEFORCE GTX core i7 processor system and have been implemented using python language. We randomly split the dataset

Table 7 Reference and introduced labels for the evaluation of the proposed models

		Model Label	
		Positive	**Negative**
Referenced	**Positive**	Positive - Positive **(TP)**	Positive - Negative **(FN)**
label	**Negative**	Negative - Positive **(FP)**	Negative - Negative **(TN)**

into 80:20 to create respectively the training and the testing sets. We used the following metrics to evaluate our models: Accuracy (Eq. 1), Precision (Eq. 2), Recall (Eq. 3), and F1-score (Eq. 4).

$$Accuracy\,(\%) \;= \frac{|TP+TN|}{|TP+TN+FP+FN|} \tag{1}$$

$$Precision(\%) = \frac{TP}{TP+FP} \tag{2}$$

$$Recall(\%) = \frac{TP}{TP+FN} \tag{3}$$

$$F1-score(\%) = \frac{2*Recall*Precision}{Precision+Recall} \tag{4}$$

The references and introduced labels for the evaluation of the proposed models are provided in Table 7.

In Table 8, we provide the performance evaluation results for the classification of the mHealth apps user reviews according to the ISO/IEC 25010 quality characteristics model using the six machine learning classifiers. As it is illustrated in this table, SGD achieved the best overall accuracy of 82.00%, followed by RF, DT, SVM, KNN, and MNB with 80.00%, 74.04%, and 73.45%, 70.50%, and 67.55%, respectively. In addition, for a multi-class classification problem, accuracy, precision, recall and F1-score do not always provide a complete performance evaluation of our classifier. For this reason, we used also Cohen's Kappa score as given by the following equation.

$$\kappa = \frac{P_o - P_e}{1 - P_e}. \tag{5}$$

where P_o the observed overall agreement, and P_e the expected mean proportion of agreement due to chance. The proposed system performance showed that the system achieved a higher value than the cutoff value (0.74 in quadratic-weighted kappa).

Table 8 The performance evaluation results (Accuracy) for the classification of mHealth apps reviews according to the ISO/IEC 25010 quality characteristics using six selected machine learning classifiers

Classifier	**MNB**	**KNN**	**SVM**	**SGD**	**RF**	**DT**
Accuracy	67.55%	70.50%	73.45%	82.00%	80.00%	74.04%

Table 9 The performance evaluation results (Precision, Recall, and F1-score) for the mHealth apps reviews classification by quality characteristic using SGD

Quality characteristics	Precision	Recall	F1-score
Compatibility	68.00%	65.00%	67%
Functional Suitability	77%	91%	83%
Maintainability	91%	67%	77%
Performance	88%	66%	75%
Portability	100%	100%	100%
Reliability	75%	38%	50%
Security	85%	74%	79%
Usability	90%	89%	89%
weighted avg	83.00%	82.00%	82.00%

Since SGD classifier provided the best accuracy result, we selected this classifier to further evaluate the system using other metrics such as Precision, Recall, and F1-score. In Table 9, we present the performance evaluation results for the mHealth apps reviews classification by quality characteristic using SGD classifier. As it is showed in this table, the weighted average values of precision, recall, and F1-score are 83.00%, 82.00% and 82.00% respectively.

Table 10 presents the performance evaluation results for the classification of the mHealth apps user reviews according to the sentiment polarity using the six machine learning classifiers. As it is illustrated in this table, SVM and RF provided the best overall accuracy of 90.50%, followed by SGD, MNB, DT and KNN with 90.20%, 90.00%, 86.05% and 81.30%, respectively. For a more appropriate evaluation, we assessed the performance of our classifier using the Cohen's Kappa score. The proposed system performance showed that the system achieved a higher value than the cutoff value (0.83 in quadratic-weighted kappa).

Since SVM and RF classifiers provided the best accuracy results, we selected these classifiers to further evaluate the system using other metrics such as Precision, Recall, and F1-score. In Table 11, we present the performance evaluation results for the mHealth apps reviews classification by sentiment polarity using SVM and RF classifiers. As it is shown in this table, the best weighted average values of precision, recall, and F1-score are 90.00%, 91.00%, and 90.00%, respectively provided using RF classifier.

Figure 5 presents the confusion matrix for sentiment polarity. As it is illustrated in this figure, 95.00% of the Negative reviews have been correctly classified as Negative, whereas, 90.00% of the Positive reviews have been correctly classified as Positive. Our system made mistakes mostly in classifying Neutral reviews, where 50% have been classified as Negative, and 50% have been classified as Positive.

Table 12 presents the performance evaluation results (Accuracy) for each quality characteristic by sentiment polarity using the six selected machine learning classifiers.

Among the quality characteristics, we excluded the Portability because its number of user reviews is very limited (6 reviews in total). As it is illustrated in this table, the SGD classifier provides better results compared to other classifiers for all the quality characteristics except for the Reliability, where SVM and MNB provided the best Accuracy

Table 10 The performance evaluation results (Accuracy) for the mHealth apps reviews classification by sentiment polarity using six selected machine learning classifiers

Classifiers	MNB	KNN	SVM	SGD	RF	DT
Accuracy	90.00%	81.30%	90.50%	90.20%	90.50%	86.05%

Table 11 The performance evaluation results (Precision, Recall, and F1-score) for the mHealth apps reviews classification by sentiment polarity using SVM and RF classifiers

Sentiment	SVM			RF		
polarity	Precision	Recall	F1-score	Precision	Recall	F1-score
Negative	87.00%	95.00%	91.00%	85.00%	97.00%	90.00%
Neutral	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Positive	94.00%	90.00%	92.00%	96.00%	88.00%	92.00%
Weighted avg	89.00%	91.00%	90.00%	90.00%	91.00%	90.00%

values. Usability quality characteristic is considered as a referential concept for the quality of mHealth apps Idri et al. (2018). As provided in Table 12, the performance of Usability reviews sentiment classification given by SGD classifier is about 98.00%, which is the best result compared to other quality characteristics.

Since SGD classifier provided the best Accuracy results, we further evaluated our model using precision, Recall, and F1-Score metrics (see Table 13). The number of Neutral reviews in the majority of quality characteristics is very limited (e.g., Compatibility, Reliability, Security). This justifies that the system could not consider those reviews as a class (None in Table 13). As an example, for the usability quality characteristic, the weighted average values of 98.00% were given for precision, recall, and F1-score metrics.

Figure 6 presents two confusion matrices for sentiment polarity for (a) functional suitability and (b) performance. As it is illustrated in this figure, for the functional suitability quality characteristic, 91.00% of the Negative reviews have been correctly classified as Negative, whereas 91.00% of the Positive reviews have been correctly classified as Positive. Our system made mistakes mostly in classifying Neutral reviews, where 83% have been classified as Negative, and 17% have been classified as Positive. Moreover, for the performance quality characteristic, 78% of the Negative reviews have been correctly classified as Negative and 22% are classified as Positive, whereas, 100% of the Positive reviews have been correctly classified as Positive. Our system made mistakes mostly in classifying Neutral reviews, where all of them (100%) have been classified as Positive.

Fig. 5 Confusion matrix for sentiment polarity

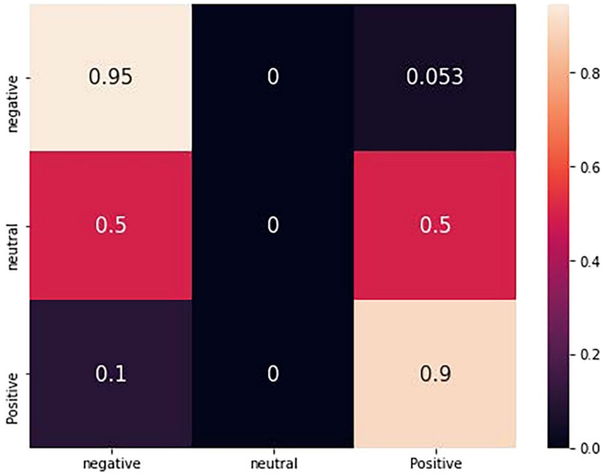


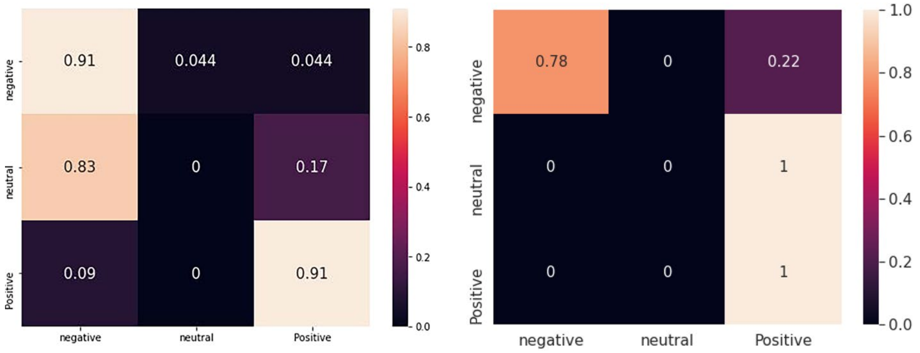
Table 12 The performance evaluation results (Accuracy) for each quality characteristic by sentiment polarity using the four selected machine learning classifiers

Quality characteristics	MNB	KNN	SVM	SGD
Functional suitability	77.11%	72.03%	83.05%	**86.44%**
Compatibility	92.00%	88.00%	**96.00%**	**96.00%**
Reliability	**100%**	93.75%	**100%**	93.75%
Security	95.00%	**97.50%**	95.00%	**97.50%**
Maintainability	87.50%	87.50%	87.50%	**91.66%**
Usability	90.00%	95.00%	91.00%	**98.00%**
Performance	84.37%	75.00%	78.12%	**87.50%**

Bold values refer to the best results obtained by each ML classifier for every characteristic

Table 13 Performance evaluation results for each quality characteristic by sentiment polarity using SGD classifier

Quality characteristics	Polarity	Metrics		
		Precision	**Recall**	**F1-score**
Functional suitability	**Positive**	95.00%	91.00%	93.00%
	Negative	79.00%	91.00%	85.00%
	Neutral	0.00%	0.00%	0.00%
	Weighted-Avg	84.00%	86.00%	85.00%
Compatibility	**Positive**	100%	50.00%	67.00%
	Negative	96.00%	100%	98.00%
	Neutral	None	None	None
	Weighted-Avg	96.00%	96.00%	95.00%
Reliability	**Positive**	0.00%	0.00%	0.00%
	Negative	100%	94.00%	97.00%
	Neutral	None	None	None
	Weighted-Avg	100%	94.00%	97.00%
Security	**Positive**	100%	50.00%	67.00%
	Negative	97.00%	100%	99.00%
	Neutral	None	None	None
	Weighted-Avg	98.00%	97.00%	97.00%
Maintainability	**Positive**	100%	50.00%	67.00%
	Negative	91.00%	100%	95.00%
	Neutral	None	None	None
	Weighted-Avg	88.00%	92.00%	89.00%
Usability	**Positive**	98.00%	100%	99.00%
	Negative	100%	85.00%	92.00%
	Neutral	None	None	None
	Weighted-Avg	98.00%	98.00%	98.00%
Performance	**Positive**	83.00%	94.00%	88.00%
	Negative	93.00%	87.00%	90.00%
	Neutral	0.00%	0.00%	0.00%
	Weighted-Avg	81.00%	80.00%	78.00%



(a) (b)

Fig. 6 The confusion Matrices for sentiment polarity for quality characteristics examples. **a** Confusion Matrix for Functional Suitability, **b** Confusion Matrix for Performance

Table 14 depicts a performance comparison between the Unigram and Bigram bag-of-words variation for both quality and sentiment classifications using all selected machine learning algorithms. As illustrated in this table, Bigram provides better performances than Unigram for both quality and sentiment classification.

5 Discussion and comparative evaluation

In this study, we empirically evaluated the machine learning algorithms that could be successfully used in the classification of mHealth apps user reviews according to the ISO/IEC 25010 quality characteristics ISO/IEC (2010), and help practitioners incorporate user feedback faster and more accurately. The implication of this study is summarized as follows:

1. The most addressed quality characteristics by the users of mHealth apps is Functional suitability, followed by usability. In fact, since the users of these apps usually have some health issues, they certainly need apps providing several functionalities and are easy to use, and with low complexity.
2. The majority of the users are not satisfied with the functionality provided by mHealth apps and their reliability, while they are mostly satisfied with their usability. In fact, the reliability of mHealth apps is important. For instance, using an app that *"Doesn't work half the time"* is not beneficial.

Table 14 Comparison (Accuracy) of bag-of-words models

Classifiers	**Bag of words**			
	Quality		**Sentiment**	
	Unigram	**Bigram**	**Unigram**	**Bigram**
MNB	64.60%	67.55%	89.61%	90.00%
KNN	70.20%	70.50%	81.00%	81.30%
SVM	65.19%	73.45%	90.50%	90.50%
SGD	79.94%	82.00%	89.31%	90.20%

3. With a limited amount of data, machine learning models could provide sufficient results for classification problems.

The automatic classification of user reviews will help developers identifying the quality issues of their mobile apps based on the users' experiences, and hence, the quality characteristics that should be improved in the next release. In addition, many people are using mHealth apps, and hence improving the quality of these apps, will certainly increase the number of their users and improve user satisfaction. The proposed model in this paper could be used also to improve the quality of other apps categories such as gaming apps and education apps.

It is difficult to draw a direct and fair comparison to the existing works, since the differences of the used categories and protocols in their empirical evaluation, except Al Kilani et al. (2019), and Uddin and Khomh (2019). Al Kilani et al. (2019) classified the mHealth app reviews into Bugs, new features, sentimental, general Bug, security, performance, and usability. Among those categories, only usability, security, and performance are considered as quality characteristics. The results obtained by Al Kilani et al. (2019) proved that the main quality characteristics addressed by the mHealth apps' users are: usability, performance, and security with respectively 839, 555, and 96 reviews. On the other hand, our collected dataset proved that the main quality characteristics addressed by users are: functional suitability, usability, security, and performance with respectively 589, 496, 198, and 159. Hence, if we exclude the functional suitability as it is literally related to the functionality provided by the mHealth apps, we can conclude that the main quality characteristics addressed by users are usability, security, and performance. In addition, using the SGD model, we obtained better results than Kilani et al., for Performance, Usability, and Security (see Table 15). On the other hand, Uddin and Khomh (2019) classified the APIs' reviews into 11 categories; among them, five are considered as quality characteristics (performance, usability, security, compatibility, and portability) using SVM/RF. This study showed that the main quality characteristics addressed by the APIs users are: usability, performance, and security with respectively 1457, 357, and 163. Compared to this study, using the SGD model, we obtained better results for the five quality characteristics (see Table 15).

In this paper, we also used sentiment polarity to analyze the user reviews on mHealth apps, and classify them into positive, negative, or neutral opinions. As illustrated in Table 4 and Fig. 4, the percentage of positive, negative and neutral tweets was 52.82%, 44.85% and 2.32%, respectively. Practitioners should not handle the negative, the neutral, and the positive reviews in the same manner. In fact, negative reviews should be addressed by developers in order to improve the quality of their apps. Positive reviews indicate that users praise the app's functionality, and are generally satisfied with it. Analyzing positive reviews allows the developers to identify the strengths of their own mHealth apps. In addition, positive reviews on other mHealth apps help developers in the maintenance and evolution of their own apps. For instance, they could identify features that should be implemented in the next release to increase the deployment of their apps. Finally, the number of neutral reviews is limited compared to the positive and negative reviews. Moreover, neutral reviews usually provide new feature request. Hence, the neutral and negative reviews should be addressed with the same importance. In fact, negative reviews address a quality characteristics that should be improved, whereas, neutral reviews suggest new feature. Positive reviews and negative reviews have an important impact on the mHealth apps rates.

Table 15 Comparison between the bag-of-words models and state-of-the-art models

Proposal	Total Reviews	Method	Categories	Results (%)		
				Precision	Recall	F1-score
Al Kilani et al. (2019)	7500	Naive Bayes	Performance (555 reviews)	12.00	9.00	11.00
			Usability (839 reviews)	36.00	14.00	21.00
			Security (96 reviews)	8.00	27.00	12.00
Uddin and Khomh (2019)	4428	SVM	Performance (357 reviews)	77.80	45.50	56.20
			Usability (1457 reviews)	53.20	74.90	62.00
			Security (163 reviews)	77.50	57.80	60.20
			Compatibility (95 reviews)	50.00	7.80	13.30
			Portability (73 reviews)	62.90	62.90	60.80
Our proposal	1681	SGD	Performance (159 reviews)	88.00	66.00	75.00
			Usability (496 reviews)	90.00	89.00	89.00
			Security (198 reviews)	85.00	74.00	79.00
			Compatibility (123 reviews)	68.00	65.00	67.00
			Portability (6 reviews)	100.00	100.00	100.00

SVM and RF models represent 0% accuracy for the Neutral class. Actually, the two models handle three-class classification (Positive, Negative and Neutral); however, the total number of user reviews with Neutral polarity is limited to six reviews. SVM and RF classified three reviews as Positive and three reviews as Negative. Neutral reviews usually provide suggestions on how to improve the app such as *"Could you PLEASE make the back button NOT close the app from every screen"*; however, for some Neutral reviews, users used at the same time words for positive opinion and words for negative opinions, such as *"..., it's incredibly* ***annoying****... Also, how about being able to expand a single day's worth of stats instead of only the main scroll screen and the weekly metrics,...* ***thank you*** *very much and everyone that needs detailed hart monitoring,* ***this a must****. (****Hard to use****.)."*

On the other hand, some researchers used deep learning models to detect quality aspects in software reviews such as Application Programming Interface (API) reviews (cf., Uddin and Khomh (2019); Yang et al. (2022), etc.). For a fair comparison with those, we applied BERT, RoBERTa, DistilBERT, and DistilBERT ML models. In Table 16, we compare between the results provided by the best performing ML models (SVM/RF and SGD) with deep learning models (BERT, RoBERTa, DistilBERT, and DistilBERT ML). As illustrated in this table, for quality classification, DistilBERT ML achieved the best overall accuracy of 76.00% followed by RoBERTa, DistilBERT, and BERT with 58.00%, 56.00%, and 51.00% respectively. For sentiment classification, BERT achieved the best overall accuracy of 85.00% followed by RoBERTa, DistilBERT ML, and DistilBERT with 83.00%, 81.00%, and 80.00% respectively. This could be explained by the fact that deep learning models need a large data for training, which is not available in our dataset.

Table 16 Comparison (Accuracy) of bag-of-words with deep learning models

Bag-of-words models		
Classifiers	**Quality**	**Sentiment**
SVM	73.45%	**90.50%**
RF	**80.00%**	90.50%
SGD	**82.00%**	90.20%
Deep learning models		
Models	**Quality**	**Sentiment**
BERT	51.00%	**85.00%**
RoBERTa	58.00%	83.00%
DistilBERT	56.00%	80.00%
DistilBERT ML	**76.00%**	81.00%

Bold values refer to the obtained best results

In Table 17, we compare the obtained results using SGD and deep learning models (BERT, RoBERTa, DistilBERT, and DistilBERT ML) with the state-of-the-art models. As illustrated in this table, for our proposed approach, in most cases, SGD model provides better results than deep learning models. Compared to Yang et al. (2022), we obtained competitive results only for Usability and Performance quality characteristics using DistilBERT ML. This is explained by the number of reviews used in the dataset. In fact, deep learning models provide better results with a large dataset.

6 Threats to validity

Threats to the validity of our study are related to internal validity and external validity.

- **Internal validity:** the main threat to the internal validity of our study is the dataset collected from the 86 mHealth apps provided by Google Play store using the Appbot tool Appbot (2021). In fact, the collected reviews were automatically classified by Appbot tool into different topics (e.g., Performance, Use cases, Bug, Feature Requests). Some of those topics respect the ISO/IEC 25010 quality characteristics model, while others do not. Hence, we classified manually the collected reviews in our dataset based on the ISO/IEC 25010 quality standard ISO/IEC (2010). We conducted these classifications carefully to guarantee their correctness and feasibility in machine learning algorithms. This issue is revealed also when examining the sentiment polarity. While collecting and classifying user reviews, the ISO/IEC 25010 quality model was appropriate for the kind of reviews we get from the mHealth apps. In fact, except Neutral reviews, all the collected reviews could be easily classified according to the several quality characteristics of the ISO/IEC 25010 model.
- **External validity:** deals with the generalization of the results of this study to other subsets of mobile applications. The method that we proposed in this paper is not explicitly applied to mHealth apps. It could be generalized to apply it to any subset of mobile apps. We believe that the proposed approach in this paper can be used to evaluate the quality of other mobile apps categories (e.g., gaming, kids, education). Moreover, user feedback can not only be classified according to the ISO 25010 quality characteristics (e.g., portability, performance) but also according to the quality sub-characteristics (e.g., availability, flexibility).

Table 17 Comparison between the deep learning models and the state-of-the-art models

Proposal	**Total Reviews**	**Categories**	**Method**	**Results (%)**		
				Precision	**Recall**	**F1-score**
Yang et al. (2022)	4522	Performance	BERT	96.30	96.20	96.20
		(348 reviews)	RoBERTa	96.60	96.50	96.50
		Usability	BERT	79.90	79.50	79.50
		(1437 reviews)	RoBERTa	80.40	79.50	79.70
		Security	BERT	98.60	97.40	97.80
		(163 reviews)	RoBERTa	98.30	98.50	98.30
		Compatibility	BERT	97.30	98.10	97.50
		(93 reviews)	RoBERTa	96.40	98.00	97.10
		Portability	BERT	99.10	99.00	99.00
		(70 reviews)	RoBERTa	99.20	99.20	99.20
Our proposal	1681	Performance	SGD	88.00	66.00	75.00
		(159 reviews)	BERT	15.00	9.00	11.00
			RoBERTa	35.00	24.00	28.00
			DistilBERT ML	92.00	71.00	80.00
		Usability	SGD	90.00	89.00	89.00
		(496 reviews)	BERT	60.00	69.00	64.00
			RoBERTa	67.00	68.00	68.00
			DistilBERT ML	93.00	71.00	81.00
		Security	SGD	85.00	74.00	79.00
		(198 reviews)	BERT	44.00	61.00	51.00
			RoBERTa	64.00	70.00	67.00
			DistilBERT ML	72.00	78.00	75.00
		Compatibility	SGD	68.00	65.00	67.00
		(123 reviews)	BERT	48.00	39.00	43.00
			RoBERTa	39.00	48.00	43.00
			DistilBERT ML	56.00	74.00	64.00
		Portability	SGD	100.00	100.00	100.00
		(6 reviews)	BERT	00.00	00.00	00.00
			RoBERTa	00.00	00.00	00.00
			DistilBERT ML	00.00	00.00	00.00

7 Conclusion

User reviews vary from relevant reviews providing ideas for mHealth improvement to reviews complaining about the app's issues or giving complaints. In the herein presented work, we used six supervised machine learning to evaluate the quality of 86 mHealth apps according to the ISO/IEC 25010 quality model based on user feedback. We collected 1682 reviews including positive, negative, and neutral opinions from the Google play store. We applied natural language processing techniques and machine learning in the process of review analysis. The evaluation results proved that the SGD classifier provided the best accuracy of 82.00% in classifying user reviews according to the quality characteristics, whereas SVM and RF classifiers provided the best accuracy of 90.50% in classifying user reviews according to the sentiment polarity.

For future work, we will propose to identify functional requirements from user feedback for a future release of mHealth apps. In addition, prioritizing relevant user reviews could be helpful in identifying which quality characteristic should be improved first. Moreover, we are planning to enlarge the dataset and use deep learning to improve the classification results.

Acknowledgements The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IF-PSAU-2021/01/17750).

Data availability The data that support the findings of this study as well as the code are publicly available through the github account of the author: RaouiaMokni. Link: https://github.com/RaouiaMokni/Machine-learning-for-mHealth-apps-quality-evaluation-An-approach-based-on-user-feedback-analysis.

Declarations

Ethical approval This article does not contain any studies with Human participants and/or animals performed by any of the authors.

Informed consent Not applicable

Human participants and/or animals Not applicable

Competing interests Not applicable

References

Agrawal, V. (2020). 3 ways digital marketers can use machine learning how a difficult industry can be made easier through the use of artificial intelligence.

Al-Hawari, A., Najadat, H., Shatnawi, R. (2020). Classification of application reviews into software maintenance tasks using data mining techniques. *Software Quality Journal* pp. 1–37.

Al Kilani, N., Tailakh, R., Hanani, A. (2019). Automatic classification of apps reviews for requirement engineering: Exploring the customers need from healthcare applications. In: *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pp. 541–548. IEEE.

Alpaydin, E. (2020). Introduction to machine learning. MIT press.

Appbot. (2021). App review & ratings analysis for mobile teams. Retrieved 11 February 2021. Available: https://appbot.co/

Araujo, A., Golo, M., Viana, B., Sanches, F., Romero, R., Marcacini, R. (2020). From bag-of-words to pre-trained neural language models: Improving automatic classification of app reviews for requirements engineering. In: *Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional*, pp. 378–389. SBC.

Aslam, N., Ramay, W. Y., Xia, K., & Sarwar, N. (2020). Convolutional neural network based classification of app reviews. *IEEE Access, 8*, 185619–185628.

Davalbhakta, S., Advani, S., Kumar, S., Agarwal, V., Bhoyar, S., Fedirko, E., Misra, D., Goel, A., Gupta, L., Agarwal, V. (2020). A systematic review of the smartphone applications available for coronavirus disease 2019 (covid19) and their assessment using the mobile app rating scale (mars). medRxiv

Dewi, M. R., Ngaliah, N., Rochimah, S. (2020). Maintainability measurement and evaluation of myits mobile application using iso 25010 quality standard. In: *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp. 530–536. IEEE.

Falih, N., Firdaus, A. (2019). Measuring performance, functionality and portability for mobile hybrid application. In: *2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, pp. 195–200. IEEE.

Guzman, E., El-Haliby, M., Bruegge, B. (2015). Ensemble methods for app review classification: An approach for software evolution (n). In: *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 771–776. IEEE.

Hebb, D. O. (1949). The organization of behavior: a neuropsychological theory. J. Wiley; Chapman & Hall.

Herrera, M., Moraga, M.Á., Caballero, I., Calero, C. (2010). Quality in use model for web portals (qiu-wep). In: *International Conference on Web Engineering*, pp. 91–101. Springer.

Idri, A., Bachiri, M., Fernández-Aleman, J. L., Toval, A. (2017). Iso/iec 25010 based evaluation of free mobile personal health records for pregnancy monitoring. In: *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, pp. 262–267. IEEE.

Idri, A., Sardi, L., Alemán, J. L. F. (2018) Quality evaluation of gamified blood donation apps using iso/iec 25010 standard. In: *HEALTHINF*, pp. 607–614.

ISO/IEC. (2010). 25010 system and software quality models. Tech. rep.

ISO/IEC. (2016). 25022 systems and software engineering – systems and software quality requirements and evaluation (square) – measurement of quality in use. Tech. rep.

Lu, M., Liang, P. (2017). 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. ACM

Maalej, W., Kurtanović, Z., Nabil, H., & Stanik, C. (2016). On the automatic classification of app reviews. *Requirements Engineering, 21*(3), 311–331.

Maalej, W., Nayebi, M., Johann, T., & Ruhe, G. (2015). Toward data-driven requirements engineering. *IEEE Software, 33*(1), 48–54.

Medina, C. P., & Ramon, M. R. R. (2015). Using tf-idf to determine word relevance in document queries juan. *New Educational Review, 42*(4), 40–51.

Messaoud, M. B., Jenhani, I., Jemaa, N. B., Mkaouer, M. W. (2019). A multi-label active learning approach for mobile app user review classification. In: *International Conference on Knowledge Science, Engineering and Management*, pp. 805–816. Springer.

Mobius MD. (2019). The medical workflow company, 11 surprising mobile health statistics.

Panichella, S., Di Sorbo, A., Guzman, E., Visaggio, C. A., Canfora, G., Gall, H. C. (2015). How can i improve my app? classifying user reviews for software maintenance and evolution. In: *2015 IEEE international conference on software maintenance and evolution (ICSME)*, pp. 281–290. IEEE.

Pospieszny, P., Czarnacka-Chrobot, B., Kobylinski, A. (2018). An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems and Software*, pp. 184–196.

Ren, J., Lee, S. D., Chen, X., Kao, B., Cheng, R., Cheung, D. (2009). Naive bayes classification of uncertain data. In: *2009 Ninth IEEE International Conference on Data Mining*, pp. 944–949. IEEE.

Singh, G., Kumar, B., Gaur, L., Tyagi, A. (2019). Comparison between multinomial and bernoulli naïve bayes for text classification. In: *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pp. 593–596. IEEE.

Tamjeed, M. (2020). Accessibility in user reviews for mobile apps: An automated detection approach.

Turner, A. (2020). How many smartphones are in the world?

Uddin, G., & Khomh, F. (2019). Automatic mining of opinions expressed about apis in stack overflow. *IEEE Transactions on Software Engineering, 47*(3), 522–559.

Vapnik, V. (2013). The nature of statistical learning theory. Springer science & business media.

Yang, C., Xu, B., Khan, J. Y., Uddin, G., Han, D., Yang, Z., Lo, D. (2022). Aspect-based api review classification: How far can pre-trained transformer model go. In: *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE Computer Society*.

Zhang, D., Tsai, J. J. P. (2002). Machine learning and software engineering. In: *14th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2002)*, 4-6 November 2002, Washington, DC, USA, p. 22.

Zulfa, F., Munawaroh, H., Rochimah, S. (2020). Portability characteristics evaluation of myits mobile using iso/iec 25010 quality standard. In: *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pp. 537–542. IEEE.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mariem Haoues Ph.D in Computer Science-Software Engineering. Member of the "Multimedia, InfoRmation systems and Advanced Computing Laboratory". She received her B.S. (2010) and M.S. (2013) in "Computer Science and Multimedia" from ISIMS - Univ. of Sfax, Tunisia. In 2018, she received her Ph.D degree in "Computer Science" from FSEG - Univ. of Sfax, Tunisia. Dr. Haoues has over 22 publications in the field of software engineering. Her research interests include software evolution, software quality, requirements engineering, healthcare, and machine learning.

Raouia Mokni Ph.D in Computer Science. She received her B.S. (2009) in "Information System". In 2018, she received her M.S. (2012) and Ph.D degree (2018) in "Computer Science" from FSEG - Univ. of Sfax, Tunisia. Since 2019, she has been Assistant professor at Prince Sattam Bin Abdulaziz University, Alkharj, Kingdom of Saudi Arabia. She is an IEEE member. Dr. Mokni has over 18 publications in the fields of Computer engineering, Artificial intelligence, etc. Her research interests include Artificial Intelligence, healthcare, biometrics, pattern recognition and image processing.

Dr. Asma Sellami is teaching at the University of Sfax in Tunisia. Her current research interest includes broadly measurement in Software Engineering, software quality and software project management. Dr. Sellami is also working on ISO standards for measuring the functional size of software, and has been involved in developing case study of ISO 19761 (COSMIC FSM Method). She published more than 40 refereed conferences, journals, and technical reports. She is currently member of COSMIC Advisory council in Tunisia.

Authors and Affiliations

Mariem Haoues[1,2] · Raouia Mokni[3,4] · Asma Sellami[2]

✉ Mariem Haoues
m.haoues@psau.edu.sa

Raouia Mokni
r.mokni@psau.edu.sa

Asma Sellami
sellami.asma@isims.usf.tn

[1] Department of Software Engineering, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University Al-Kharj, Al-Kharj 11942, Saudi Arabia

[2] Mir@cl Laboratory, University of Sfax, ISIMS, BP 242. 3021., Sfax, Tunisia

[3] Department of Information System, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University Al-Kharj, Al-Kharj 11942, Saudi Arabia

[4] University of Sfax, Sfax, Tunisia