Towards Security Awareness of Mobile Applications using Semantic-based Sentiment Analysis

Ahmed Alzhrani, Abdulmjeed Alatawi, Bandar Alsharari, Umar Albalawi, Mohammed Mustafa
College of Computing and Information Technology, University of Tabuk, Tabuk, 71491, Saudi Arabia
Industrial Innovation and Robotics Center, University of Tabuk, KSA, Tabuk, 71491, Saudi Arabia

***Abstract*—With the rapid increase of smartphones and the growing interest in their applications, e.g., Google Play Apps, it becomes necessary to analyze users' reviews whether they are expressed as ratings or comments. This is because recent studies reported that users' reviews could provide us with useful clues and valuable features that can help in understanding the broad opinion about some applications in term of security awareness. Several techniques have been developed for this crucial task and significant progress have been achieved such as Semantic and Sentiment Analysis, Topic Modelling, and Clustering. The majority of the existing methods are mainly based on representing reviews' words in a Bag-Of-Words vector space with String-matched approaches without considering the common polysemy and synonymy problems of words. This is true due to the fact that users who make use of these applications are often from a diverse background and thus, different vocabulary. This paper proposes a new approach to classifying security opinions about applications from users' reviews while considering special features of synonymous and polysemous words. To achieve this task, the proposed model makes use of word embedding, topic modelling, Bi-LSTM, and n-grams approach. For the proposed model, a new dataset is built that contains reviews about 18 popular applications. The application's selection was primarily governed by making the dataset diverse in its domain. The experiment results showed that the proposed ensemble model which combines the prediction of the extracted features, which in turn captures synonymy, polysemy, and dependency of words-is significantly useful, and it achieves better results with an accuracy approaching 90% compared to the use of each technique separately. The model could contribute in preventing mobile users from unsafe applications.**

***Keywords*—Security awareness; semantic analysis; sentiment analysis; mobile applications; topic modelling; clustering**

I. Introduction

The history of smartphones dates to the 1980s; however, the revolution in the industry came when the first iPhone was introduced in the market. This revolution has completely changed the life of humanity and how people interact; and what seemed like a distinct dream has become a tangible reality. After 2007, users felt the power within their hands, with smartphones becoming an integral part of every life. Today, there are close to six billion smartphone users, which amounts to over 70% of the world population. It is estimated that access to smartphones would reach over 7.5 billion in the next five years [1], as shown in Fig. 1. Smartphones are also expected to affect all aspects of life on earth in the coming years. Smartphones have evolved over the last decade and have come a long way from being a prized possession to an ultimate necessity. When IBM introduced the first smartphone, it was more about the touchscreen, email, fax, and notes than anything fancy or smart power in hand. The first smartphone was never that powerful and looked like it would take ages to reach the common people. It was in the year 2007, however, when the first iPhone was introduced, and it captured the market by storm. It possessed a number of features, including the capacity to play music and capture images, not to mention that it was priced at something that people could think to own [2].

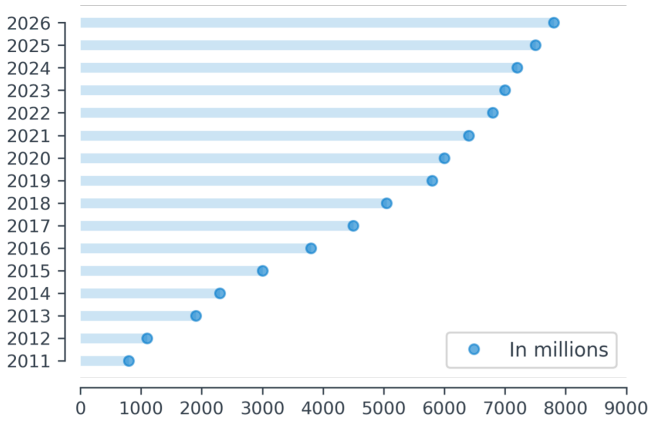

Fig. 1. Number of Smartphone Users.

Today, the 'New Generation' of smartphones have increasingly become more innovative and eye-catching. During the current pandemic (COVID-19), smartphone users have increased [3]. While in lockdown, such users have befittingly found solace in these gadgets, allowing them to stay connected with their families, gain access to information, and use them to join work platforms, such as Teams and Zoom, in addition to other users. Many companies started producing smartphones and application distribution platforms. For example, Samsung entered the market strongly and distinctively in 2009 through the Android operating system and has since been continuously developing its famously known application store 'Google Play' (GP). There are millions of different smartphone apps that are emerging as the number of users increases and technology develops. In 2017, the Google Play store had approximately 3.5 million apps [4], as shown in Fig. 2.

One of the advantages of the applications is that they allow users to review and evaluate. Therefore, any application downloaded by any of the operating systems can get

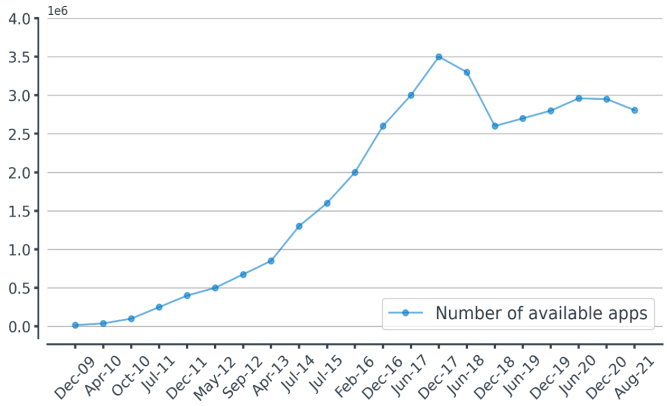

Fig. 2. Number of Available Applications in Google Play.

reviewed by its users, which lets others know about the app's benefits, drawbacks, or any security issues. User reviews and ratings are some of the most critical factors in determining whether or not an app will survive and prosper in the market. Furthermore, data scraping can be used with these reviews to benefit from them or to provide information. According to a survey published in 2018 [5], 94% of buyers avoided a business after reading unfavorable reviews about it online. Interestingly, 80% of clients would avoid doing business with a company if it had less than four stars, which is also relevant to the app industry. Hence, it is crucial to take app reviews into consideration. Through the advancements in new apps, software, and technologies, IT security is forced into continuously being upgraded to prevent it from breaching privacy. With so many apps, as well as social media platforms available to communicate and express ideas freely, technology has aided an explosion of data in recent years. This has created a loophole through which smartphone users' personal information can be easily accessed, raising larger security concerns and difficulties.

Smartphones invading human life has produced multiple apps that are related to almost everything a user does in their routine life, indicating that the user of a smartphone is never alone and always being watched by someone or the other. Today, there are apps developed for everything: job-hunting, reading books, watching movies, or anything else throughout the day [6]. These apps give their users the freedom and ease to complete the requirements ranging from simple data collection or information gathering to entertainment or any further assistance. According to [7], the number of individuals using mobile banking applications has surpassed two billion, accounting for around 40% of the global adult population. A few fact checks of the security issues in these apps [8] are as follows:

- In the second quarter of 2018, mobile applications and mobile browsers accounted for 71% of all fraud transactions.
- In 2018, tools and leisure applications accounted for 54% of malicious mobile apps.
- The United States accounted for 25% of all mobile malware incidence, India for 23%, and China for only 3 %.

Modern mobile operating systems include a variety of security features. Installed apps can only access files in their own sandbox folders by default, with user privileges preventing them from changing system files. However, mistakes made by developers while designing and coding for mobile applications leave security vulnerabilities that adversaries may exploit. A lack of multi-factor authentication is one of the most frequent risks to mobile app security. Without this protection, a hacker only needs a tiny amount of personal information to access the data.

The rest of this paper is organized a follows: The Background and related works are described in Section II. The database collection of this paper is summarized in Section III. Section IV discusses the datasets analysis. In Section V, the proposed model is discussed. Section VI illustrates the experiment results, while Section VII concludes the paper and briefly discusses areas of future work.

II. Background and Related Work

Several methods and techniques have been proposed in the literature for the task of how to extract sentiments in user's review texts. Some of these methods were built on top of supervised learning, in which a large set of data is being labelled, whereas others were based on unsupervised learning. However, the common theme among the provided studies is how to extract useful features that can help in identifying sentiments or opinions. Besides the use of conventional classification techniques that are based on the traditional frequency of both positive and negative words, the utilized approaches for feature extraction that are related to this study can be informally categorized into three main categories, these are Sentiment-based, Semantic-based and Topic-based techniques. Next section discusses each of which in more detail.

A. Sentiment-Based Analysis

Prior studies have proposed various sentiment analysis approaches. The authors [9] used text mining to summarize user evaluations and extract key features from Android app reviews. They then utilized a natural language processing (NLP) technique to write rules. They used SAS® Enterprise MinerTM 7.1 to summarize reviews and pull-out features, and SAS® Sentiment Analysis Studio 12.1 is utilized to perform sentiment analysis. Reviews from two Apps from two different categories were used in the experiment; namely: "Where is my Perry" from "Brain & Puzzle" category and "Beautiful widgets" from "Personalization" category. Data was gathered from the Google Play Android App Store. Textual reviews with rich content were collected from the App Store website. Six hundred reviews were extracted, A corpus of 500 reviews was used for text mining, constructing sentiment models, and developing sentiment rules, while a testing dataset of 100 reviews was used. Each textual review is categorized into a positive or negative directories based on overall ratings depending on the 5-star rating scale used by Google Play. Rule based models were applied to the testing datasets. The NLP rule-based model outperformed the SAS® Sentiment Analysis Studio 12.1 default statistical model in predicting sentiment

in test data. In analyzing customer sentiment, NLP rule-based models give deeper insights than statistical models.

In the research [10], A series of comparison studies were carried out using classification algorithms to classify emotions. Review length and feature representation. The experiments were carried out on over 1400000 mobile app reviews based on four characteristics: 1) short average length; 2) large span of length; 3) power-law distribution; 4) Significant difference in polarity. This research makes a comparison between two classification techniques; namely: Support Vector Machine (SVM) and Naïve Bayes. The experimental results stated that: 1) Bayesian method is proved to perform better than the SVM on the classification link; 2) N-Gram is applied (N=2) for the best result on feature representation after the Chinese word segmentation; 3) Feature extraction process can improve the sentiment classification accuracy when reviews have more than 150 words; 4) Short reviews are more easily classifiable then long ones.

Zhang et al. [11] built a novel system for named MoSa (Mobile Sentiment Analysis) for data analysis. The proposed system employs algorithms for detecting and analyzing different types of application big data, such as news comments. They proposed a sentiment analysis approach relying on a mixed sentiment dictionary and new algorithms for calculating comment sentiment scores and categorizing comment sentiment tendencies. The mixed sentiment dictionary is more effective at classifying short texts according to experimental data. The average word length should be less than 2.3 when utilizing a sentiment dictionary to analyze brief texts. They also suggest certain unique statistical models that describe public behavior. In doing so, they discovered that standard deviation is a better indicator of public sentiment than other statistics.

The study [12] proposed a security-related and evidence-based ranking scheme, called SERS. This approach is utilizing the principles of Subjective Logic operations, theory of evidence, Sentiment Analysis techniques, Static taint analysis, and NLP. SERS achieves a holistic rank ordering of related apps and generates insights linked to apps available on the Google Play Store, in both structured and unstructured contexts. They tested their method on publicly available apps from the Google Play Store and compared their results to common ranking methods like average star ratings.

The review paper [13] focuses on the sentiment analysis (SA) for users' review of mobile apps to extract user requirements for developing new apps or improving existing ones, i.e., requirements evolution. They have investigated the approaches employed in the literature during the years 2009 to 2015 to answer the following research questions: 1) "Are there any publications about using SA of mobile app users' reviews for requirements evolution?", 2) "What are the methods used?", and 3) "What are the tools used?". Results demonstrate the value of using SA to analyze user reviews and reports, automated methods, and tools for evaluating reviews with features and sentiments.

The research published by Rizk et al. [14] utilized the naïve Bayes method and LDA algorithm for determining the customer sentiment on mobile banking apps and what aspects need to be maintained or improved in the app. The data used for the experiment were collected from the user reviews on Google Play store. Data were manually labelled and two classes (positive and negative) were generated. Naïve Bayes is utilized for the sentiment analysis process, LDA is used for the process of topic modelling. The experiment's findings revealed that the Nave Bayes approach had a high level of accuracy, recall, and precision. The value of k=5, which equals 86 % accuracy, 93 % recall, and 92 % precision, has the highest accuracy, recall, and precision. The most common topics in negative classes, according to the LDA method, are OTP code delivery constraints, application login problems, and network connection issues. The most often common topics in positives classes, on the other hand, were ease, simplicity, and helpfulness.

Lavanya et al. [15] proposed an opinion mining algorithm utilizing the word alignment model for extracting opinion targets and opinion words from online reviews extracted from Twitter. The proposed project aimed to design opinion words and opinion target predicting algorithms that mine user reviews posted on Twitter to analyze the market status of a particular product. The results of the experiments show that the pro-posed method achieves higher accuracy in a more efficient manner. Furthermore, the project's ultimate goal is to create possible consumer-oriented items such as mobile phones, laptops, and so on.

The study [16] has merged three techniques; namely, 1) Sentiment analysis (SA), 2) NLP and 3) Text Analysis (TA), to classify the reviews of mobile apps into categories related to software maintenance and evolution. This study provides a high-level taxonomy of sentences' categories included in user reviews that are related to the maintenance and evolution of mobile applications. Moreover, it provides a novel approach using NLP techniques to extract the user Intentions expressed in user reviews. The study's findings revealed that combining NLP, TA, and SA methods allows app developers to discover meaningful phrases with higher precision (75.2%) and recall (74.2%) than utilizing each methodology separately. The authors also demonstrated that increasing the size of the training set improves accuracy and recall in specific setups. They also discovered that a classifier that is trained using both structure (NLP) and sentiment (SA) characteristics performs considerably better than the one that is just trained with text (TA) features.

B. Semantic-based Analysis

The authors [17] proposed a framework for automatically analyzing the differences and similarities between app reviews from the Google Play Store and tweets based on the semantics of the words. They demonstrated that the framework can automatically identify similarities and differences using statistical tests and human expert review. This system may be used instead of the costly and unreliable crowd sources (due to the evaluation of non-experts). By filtering similar and different issues, it can decrease redundant information and group the main points. The results of multiple experiments, which were compared to expert evaluation, showed that it may be used to find similarities and differences between extracted topics, n-grams, and user comments.

Yadav et al. [18] proposed a novel framework that employs Word2Vec word embeddings to incorporate semantics into

app user feedback analysis. They use Google Play, Store and Twitter as a case study to see if their method can detect similar/different comments in the two well-studied types of bug reports and feature requests in the literature. Statistical analysis and human expert evaluation both validated the result. The result demonstrated that this approach could measure the semantic differences between users' comments in both groups automatically. The framework may be used to create intelligent tools that combine user input from various platforms while also allowing for automated analysis of reviews.

C. Topic Modelling and Clustering

The authors [19] conducted a study employing machine learning techniques to compare national cybersecurity strategies (NCSs) for different nations. Topic modeling and clustering methods were utilized to investigate the similarity and differences between NCSs and to identify the underlying topics that are appearing in them. A total of 60 NCSs developed between 2003 and 2016 were gathered and examined. They discovered that membership in international intuitions could be a determinant factor for NCS harmonization and integration using institutional theories. The study indicated that quantitative analytical methods such as LDA and clustering can be used to acquire a wider picture and insights during the formation of NCS while analyzing qualitative data such as textual policies strategies, and legislations. The clustering method's results assisted us in gaining a better understanding of the overall similarities between NCSs. The findings suggested that members of a group such as NATO or other like-minded allies have established more integrated and coordinated NCS.

Moubayed et al. [20] introduced a tool that summaries, categorizes, and models such data sets, as well as a search engine that allows users to query the model created from the data. The tool is based on a technique known as probabilistic topic modeling, which goes beyond document lexical analysis to model the intricate relationships between words, documents, and abstract concepts. It will help academics query the papers' underlying models and access the library of documents, allowing them to be sorted thematically. LDAVis, a third-party tool, is also included to provide a more in-depth understanding of the LDA model's inner structure. The tool has simple and intuitive navigation features that display the words inside a subject as well as the relationships between topic and words. This is a very useful tool for security experts to assess the quality of the created model and fine-tune the search queries they provide to the search engine.

D. Analyzing the Current State of the Research in Users' Reviews (Security)

To build a well-designed architecture, firstly existing studies are analyzed briefly, in terms of whether the used model can capture the unique characteristics of English language that often present in reviews.The main four features are whether they are covered by the described studies or not. These features are synonymy, polysemy, contextual information inside text, hidden and latent variables in texts, and word dependency. Table I briefly discusses the outcomes and the main shortcomings in each of the listed studies.

III. Database Collection

Existing corpora are either distributed under some license policy (not accessible) or they do not suite this study, which is the classification of English reviews for some of the most popular applications from a security perspective. For this reason, we decided to collect our own corpus with an eye to make it public for researchers in the near future. English reviews were crawled by making use of a quantitative approach from the mobile app metadata of Google Play Store during the period of January 2021 to September 2021. In particular, several tools including the most popular Google Play Scrapper are used for this purpose. Accordingly, a Python application is built based on the API of Google Play Scrapper. One of the elegant features of Google Play Store is that it does not require any dependencies. In the process, however, a list of some popular keywords in security and cryptography domains was prepared by an expert in the field. Examples for the employed tokens include words like cryptography, confidentiality, breach, crack, etc.

Recently, the research [21] showed that racing game applications are often downloaded by users and their reviews are frequently reviewed. The same study also showed that education apps are the most dominant in Google Store. Motivated by the arguments and with an eye on analyzing security in several domains, six categories have been chosen in the most popular domains and from which the majority of the applications are often downloaded. These are games, finance, education, shopping, entertainment, and social media. The domains were specified from mobile app metadata.

This study focus is concentrated around 18 common and well-known applications, each of which is under a specific category. In particular, in each domain, three applications are chosen among the top popular ones. The intended period was from 2013 to 2021. Given these constrains, about 250435 reviews are collected. Due to the messy theme of the raw reviews (and before the features were extracted), the data was firstly gone through various preprocessing. The main of this processing is to populate text into a standard and canonical form and thus, it can be analyzed. Those preprocessing steps are as follows:

1) Awkward Reviews: the data was filtered to remove awkward reviews and those were in mixed languages (i.e., English and French) and emojis.
2) Punctuation Marks and Symbols: to have clean text, all punctuations, marks, and symbols were eliminated.
3) Normalization: texts were normalized, trimmed, and cases folded, started with removing stopwords. Stopwords are often eliminated to compress corpus size. Additionally, they are non useful and of little importance because they are present approximately in the majority of the reviews. However, since the data had many misspelled words, a list of additional awkward stopwords is identified that were removed during this cleaning phase. For example, the phrase doesn't (does not) has been removed.
4) Porter Stemmer: After that, the texts in the dataset were stemmed using Porter stemmer. Porter stemmer is a light stemmer that can chop off the suffixes to render different forms from a single stem.

TABLE I. THE COMPARISON STUDY OF SENTIMENT AND SEMANTIC-BASED ANALYSIS

References	Contribution	Approach	Potential shortcoming
[9]	A sentiment feature-based model that combines text mining techniques with NLP. In particular, the learned features were automatically used to extract keywords.	Text clustering techniques from text mining plus Part-Of- Speech (POS) in NLP.	- weighting method used for clustering was based on simple Frequency. - Contextual information was not exploited. - Dataset size was relatively small. - POS needs large data to correctly identify sentiment in reviews. - The impact of stopwords removal was not investigated.
[13]	Investigating the importance of employing sentiment analysis techniques in providing a picture about the mobile applications.	A general review paper.	- No empirical analysis for the employed approaches and techniques. - The tools that were covered by the study were not assessed.
[14]	Identifying the weaknesses and strengths in mobile banking apps through the analysis of topics in users' reviews comparing results to those obtained through Naïve Bayes classifier.	Latent Dirichlet Allocation (LDA) topic model.	- Neither Hidden Topic Markov Model (HTMM) nor N-gram language models were used in the study. - Additionally, the use of LDA is not a good option when dealing with short texts, especially those with messy nature like reviews. In such cases, the use of NMF is the most adequate approach.
[15]	An opinion mining algorithm was proposed to extract opinion words and target opinion from Some online reviews extracted.	The IBM-1 alignment model, which is originally used for translation task in NLP.	IBM model needs a considerable large amount of parallel data to extract lexical weights. However, a small size of sentences were used in the study.
[16]	Classify the reviews of mobile apps into groups related to software maintenance and evolution.	Sentiment analysis, NLP, Text Analysis.	The study needs to be extended to a larger number and variety of apps.
[18]	Integrating users' reviews and feedback about specific application(s) from different platforms.	Google's word2vec Pre-trained word embedding model.	- Topicality has not been modeled in the developed approach. - Dependency of words in text was not investigated and thus, words were handled as if they were independent inside the texts.
[20]	Building a topic-based (semantic-based) search engine to tackle the change in security threats.	Latent Dirichlet Allocation (LDA) topic model.	Bag-Of-Words topic modelling does not consider Markovian relations between topics. Neither HTMM nor N-gram language models were used in the study. Thus, topics may be incoherent.

TABLE II. DISTRIBUTION OF COLLECTED REVIEWS IN DIFFERENT APPLICATIONS AND DOMAINS NORMALIZATION PROCESSES

Category	Application	Number of Reviews
Games	Pubg Mobile	58139
	Call of Duty	
	Among us	
Education	Google Classroom	9160
	Duolingo	
	Google Play Books	
Shopping	Amazon Shopping	24228
	AliExpress	
	Wish	
Social Media	Snapchat	28534
	Tik Tok	
	Twitter	
Entertainment	YouTube	38168
	Prime Video	
	Netflix	
Finance	PayPal	18907
	AlRajhi	
	Investing	
Total		177136

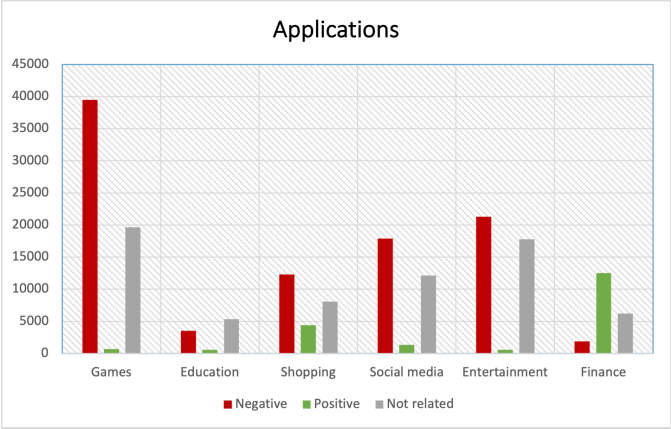

Fig. 3. The Distribution of the Three Classes According to Category.

Following this, the extracted reviews were stored into a CSV format file containing their corresponding metadata. After the above normalization processes, Table II illustrates the distributions and the total number of reviews of each application with its category. For Training phase, the dataset needs to be labeled. Accordingly, we decided to classify the sentiments of the review texts into three main categories (Positive, negative and Neutral). The word neutral means the review is actually not related, but it was collected because of one of its constituent words is on the keywords list. One might ask why to eliminate such reviews. After a deep discussion, we decided to keep them so as to mimic what actually happening in the real world. From that perspective, a human judgment process was conducted to label the reviews in one of the three aforementioned classes. Fig. 3 shows the distributions of the three classes according to category.

IV. DATASETS ANALYSIS

At first, the distribution of labelled classes from two prospective: users' ratings and the class labelling is investigated. In all reviews that were collected from different domains, the user's rating is of 1 to 5 stars. Fig. 4 plots the distribution of different ratings using a pie chart. In the figure, it is obvious that the higher rating for all applications together is found to be of user rating level 1.

This indicates that the overall satisfaction in terms of the selected is high. However, users' ratings may not be perfect. Therefore, the correlation between human labelled classes and these users' rating levels is studied. In particular, exploring whether the sentiment polarity is correlated with users' ratings. Therefore, the ratings levels is categorized as shown in Table III. In the same table, the total number of different review categories plus labelling on the same categories, are also provided.

From Table III, it is evident that the distribution between the two types of labelling (users' rating vs. our labelling)

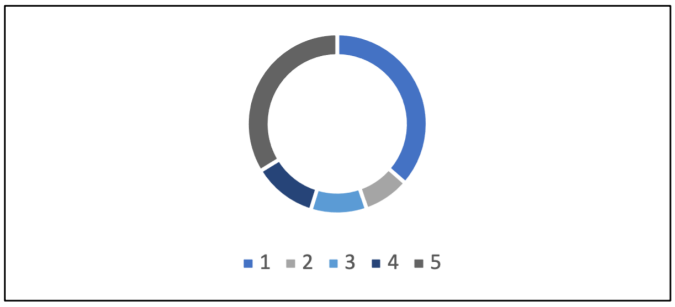

Fig. 4. The Destruction of the users' Ratings for All Categories Together.

TABLE III. THE CATEGORIZED LABELS FOR USERS' RATING AND OUR LABELLING

Rating	Categorized Label	Users' Rating Class	Our Labelling
≥ 4	Positive	73774	17216
$\doteq 3$	Natural	86357	66406
≤ 2	negative	17005	93514

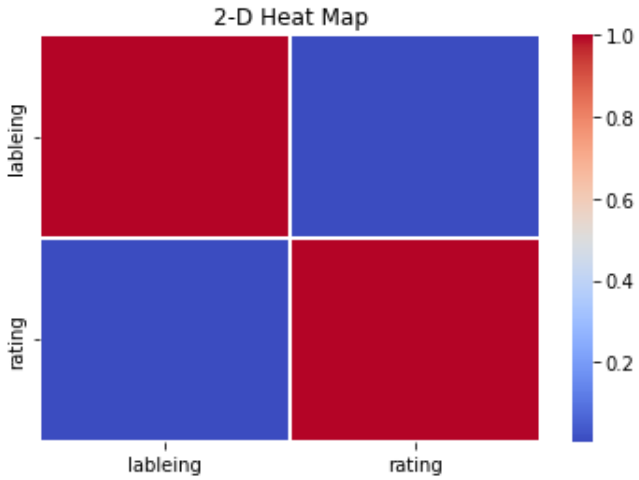

Fig. 5. The Destruction of the users' Ratings and Human Judgment Labeling.

are significantly inconsistent and extremely varied. We cannot claim that the labelling is perfect, but it has been judged by the same team, unlike the users' ratings, which are judged by a large number of users, and thus, we can claim that the labelling (human labelling for distinguishing purposes) is better than the identified classes of human labels. This is an important conclusion because often researchers attempt to avoid the boring and time-consuming process of labelling. Therefore, unless the ratings are checked statistically, we cannot depend on users' ratings, which are very subjective among a large number of users.

Fig. 5 shows the correlation between the two identified types of labels (Human labelling vs user-rating labels) using HeatMap. It is clear that the two types of ratings are highly uncorrelated. This is an indication for the fact that users' ratings are not aligned with human judgment labelled about reviews. The conclusion is aligned with Table III. Next step is to analyze why there are many unrelated reviews. We went through several reviews, not related to the study but they were crawled. From the proposed analysis, it is showing that the main problem for this phenomenon is the polysemy of words. Polysemy means a single word that have different meanings. For example, the word apple in the information technology domain has a very different meaning from its meaning in the agriculture domain. In the former field, it is likely related to apple phone, whereas in the latter, it refers to fruits. This polysemy nature of words in English texts cannot be disambiguated unless contextual information (topicality) and homonymes inside the texts are captured. There are many topical model and approaches that can be utilized to mitigate this problem, e.g., Latent Dirichlet Analysis (LDA), Words' Embeddings, and BERTopic (Bidirectional Encoder Representations of Transformers).

Nevertheless, the use of such models may not resolve this problematic phenomenon and the availability of contextual information is doubtable. This is because texts in reviews are often too short with relatively weak co-occurrence associations among words. As a consequent result, the texts may not be interpreted (even the hidden topics are captured) due to lack of contextual semantics and topicality. This is one of the main targets for this research, to extract some useful features based on hidden and contextual information in reviews' texts. In Fig. 3, it is also evident that the probability distribution of the collected reviews is imbalanced. This bias can easily make the proposed model, as it will be described later, biased towards a specific category and/or classes. For this reason, in the experiments, a random up-sampling is implemented for each category to be aligned with the main category in the dataset.

V. PROPOSED MODEL

A. Motivation

Traditional approaches for representing texts in documents/reviews/tweets attempt to create some degree of association between words and documents (reviews in this case) to build some co-occurrence word-document matrix to standard weighting schemes for normalized and stemmed words. In such models, the order of words does not have any impact on the entire process, known as Bag-Of-Words (BOW) model. The main drawback that may occur here is that the model does not reveal the structural semantics among words and the hidden meanings of the entire context inside text. This means that the linguistic contexts of words are only eliminated due to their frequency in each document and among documents within an unordered set of words.

The problem here is that words in human language are polysemous by their nature. The term polysemy means a single word which can have several meanings and senses. On the other hand, synonymy is very prevalent in natural human languages. Synonymy refers to the opposite in which different words may have the same meaning. It is evident that traditional techniques for representing words are limited towards problematic synonymy and polysemy; and susceptible to mismatching problems. Synonyms and polysemy cannot be adequately captured unless three main features in the texts are considered:

1) How to represent words? it is a essential question. In traditional approaches, words are represented using

standard TF*IDF weighting. In such a case, it cannot be predicted the best word from its surrounding text.

2) Capturing hidden topics and latent semantics and variables. Discovering such hidden structures is extremely valuable because more functions (more than classification or clustering technique) can be utilized for different purposes.
3) Capturing the dependence between texts' words. This means that the order is a matter here. This is not an easy task, because the dependency can be captured bidirectionally (in a forward or backward direction). This is be-cause human usually does not begin their thoughts from scratch, there are always some dependencies that are represented inside text. For example, individual words can be understood according to their previous and next words.

B. The Proposed Architecture

Inspired by the arguments above, the proposed model is a carefully designed architecture that can predict reviews while considering their contextual information, ordering and hidden topics. In a brief description as shown in Fig. 6, the proposed model uses:

1) Traditional n-grams TF*IDF, which is the n-gram occurrence of a "n-word" phrase. In this case, it will capture which words are more likely to occur in reviews and which words are more likely occur together.
2) Distributional Representation: the proposed model will use word embeddings. The latter allows us the next representation of words that can understand their meanings.
3) Bi-Directional Long-Short Term Memory (Bi-LSTM), which is a Recurrent Neural Network (RNN) model used to capture dependencies inside texts.
4) For topic modelling, the proposed model will use the Non-Negative Matrix Factorization (NMF). NMF is a model used to find hidden and latent topics, which generate text according to some hidden topic distribution. One main feature of the NMF is its ability to handle short texts as those present in reviews.

VI. Experimental Results

Down-sample technique is used to balance the distributions across different classes in the dataset. Accordingly, the labelling classes with the highest frequencies were down sampled to the lowest one. The outputs were then merged and then randomly sampled together (shuffled) resulting into 17829 for each class instead of 91283, 17829, and 68024 for negative, positive, and neutral classes, respectively. Following this, the dataset was split into training, testing, and validation partitions with 80%, 10%, and 10% of the entire dataset, respectively, as illustrated in Table IV.

A. Traditional N-Gram (TF-IDF) Model

The model starts to create various features with word n-grams model with n [1, 3]. Hence, the entire reviews were converted into a Bag-Of-Words model. Since the number of n-grams was found to be extremely high, the model tries to use top k-grams with k to be determined empirically. Following this, the features were fed to four different classifiers one by one, these are Logistic Regression, Naive Bayes, Decision Tree and Random Forest. The best result was achieved by employing the bigrams model on the top 296633 features with the logistic regression model. In particular, the overall accuracy performance was found to be 88%, 76%, and 77% for training, validation, and testing datasets. Fig. 7 plots the accuracy for different n-grams.

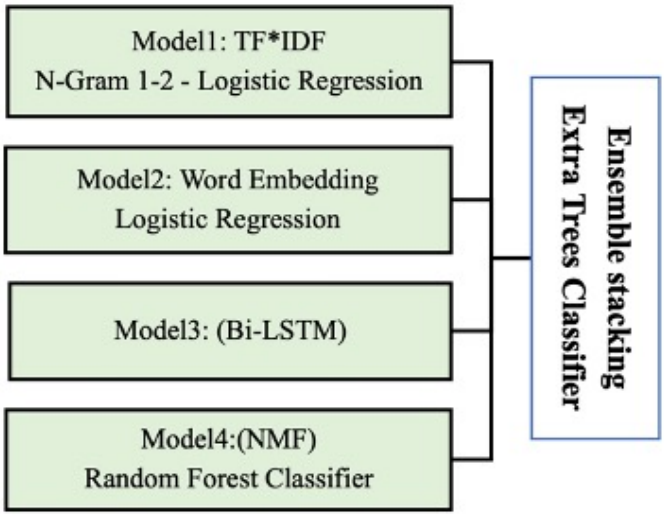

Fig. 6. The Proposed Ensemble Stacking Model.

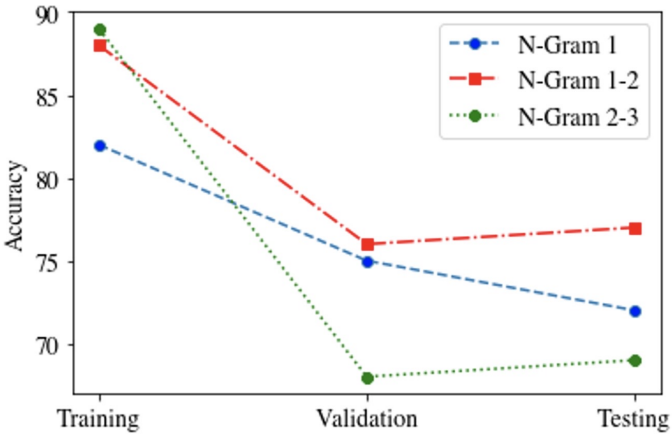

Fig. 7. The Accuracy of Logistic Regression Classifier with n-grams-based Features.

B. Word Embedding Model

In the second run of the experiments, the model extracts features using the word2vec skip gram pretrained model of distributional representation of reviews' words. Hence, the embedding of each word in each review is extracted from the pretrained model and the average of all embeddings of the voter words is computed for each review. In the experiments, 300 features were used for each word. The computed vectors

TABLE IV. DISTRIBUTION OF COLLECTED REVIEWS IN DIFFERENT APPLICATIONS OF THEIR DOMAINS AFTER BEING BALANCED

Category	Application	# of Reviews	After balancing process			
			All	Negative	Positive	Natural
Games	Pubg Mobile	58139	13171	7375	642	5154
	Call of Duty					
	Among us					
Education	Google Classroom	9160	2542	625	520	1397
	Duolingo					
	Google Play Books					
Shopping	Amazon Shopping	24228	8574	2423	4094	2057
	AliExpress					
	Wish					
Social Media	Snapchat	28534	7259	3080	1142	3037
	Tik Tok					
	Twitter					
Entertainment	YouTube	38168	9086	3981	536	4569
	Prime Video					
	Netflix					
Finance	PayPal	18907	12855	345	10895	1615
	AlRajhi					
	Investing					
Total		177136	53487	17829	17829	17829

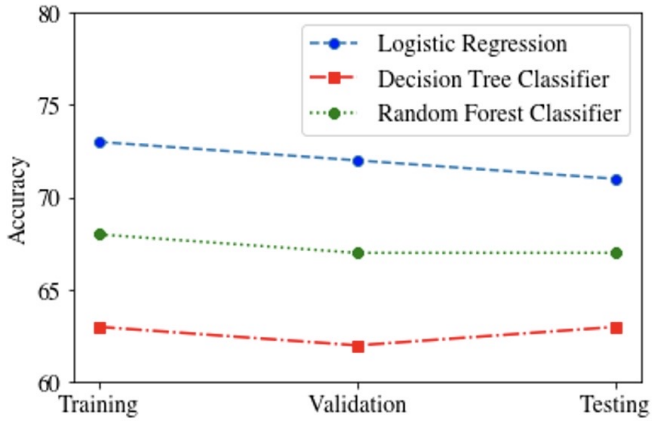

Fig. 8. The Accuracy of the Three used Classifier Word Embedding Model.

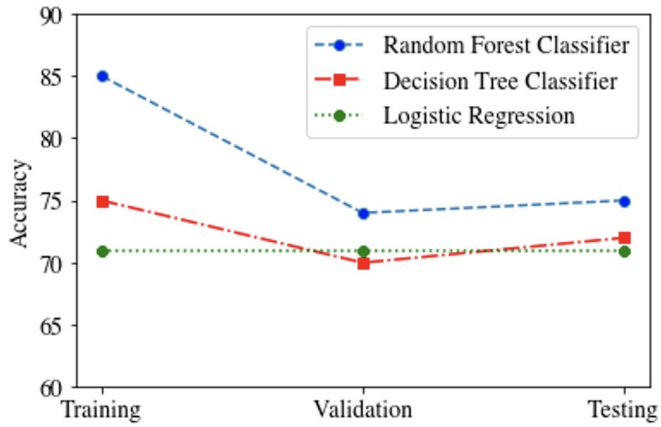

Fig. 9. The Accuracy of the used Classifiers with NMF Topic Modelling Features.

are then fed to logistic regression, decision tree and random forest classifier. Logistic regression classifier also obtains the best accuracy 73%, compared to decision tree and random forest, which both result in almost similar results - 72% and 71%, respectively. The Fig. 8 shows the results of words' embedding using the three classifiers.

C. Bi-Directional Long-Short Term Memory (LSTM) Model

The fundamental concept of LSTM is the feedback connection. At first, the model tokenizes the corpus with a maximum unique word = 25000 words out of +26000 words in the whole corpus, then fitted the tokenizer to the train dataset, and transform all datasets. After that, it pads the tokenized dataset to keep all records with the same length and only needed and effective maximum sequence length which equals 120 features. After that, the model builds the Bi-LSTM architecture, train the Recurrent Neural Network (RNN) model on the padded training dataset, and evaluate the RNN through prediction accuracy. The best model is a simple Bi-LSTM NN with 1 embedding, 1 bidirectional layer, 1 dense layer with dropout 30% with 120 features the maximum sequence length used in the padding step with accuracy 85%, 77% and 78% for training, validation and testing datasets.

D. Non-Negative Matrix Factorization (NMF) Model

In the forth experiment, the model extracts hidden topics that cause the texts in review to occur. For this purpose, it uses NMF. NMF is an unsupervised model that has the ability to discover such latent variables from a large amount of data and the end product is a probability distribution for every topic over words and how likely a word belongs to each topic is also extracted as a distribution over topics. One accredited feature of NMF is that the model has shown to be more effective - compared to other topic models - when it is used for short texts, as those present in reviews. From that prospective, the frequency of the words were extracted using the countvectorizer. the model attempted a different number of hidden topics and empirically concluded that the best number of topics to be extracted is 150. Results, which are plotted in Fig. 9, show that the best result by this method is obtained when a random forest classifier was used with an accuracy approaching 86% in the training dataset and 75% in the testing dataset.

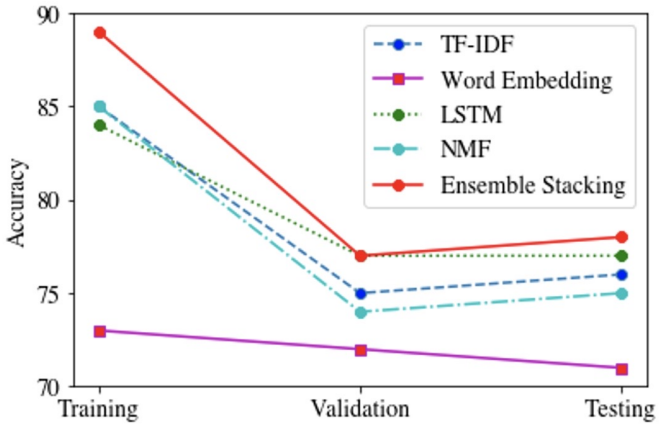

Fig. 10. The Accuracy of the Models Compared to the Proposed Ensemble Model.

E. Ensemble Stacking Model

In the final set of the experiments there is an ensemble model that employs predictions of the base model that created to build a new model. Therefore, instead of averaging all extracted features from different models, we used the prediction output to feed the ensemble stacking model. Accordingly, the parameters of the model were estimated, and the results showed that the accuracy was increased significantly (it almost approaches 89%) compared to the base models. Fig. 10 plots the accuracy of all models when they were compared to the proposed ensemble model.

VII. Conclusion

Several methods and techniques have been developed for the vital problem of extracting and classifying sentiment inside users' reviews of mobile applications. The problem is challenging because in human language, users can express their thoughts in different vocabularies and words, which are synonymous and polysemous by their nature. In particular, polysemy, in which a single word can have several senses, is significantly difficult. Additionally, texts convey contextual information that cannot be captured using conventional techniques like word-document matrices. In other words, words often tend to be generated according to some hidden topic distributions and they are also dependent inside text.

Inspired by these arguments, this paper attempts to contribute to security threat control by classifying automatically mobile applications based on their accompanied reviews. Accordingly, the proposed model combined the words embedding' features that are extracted from reviews of approximately 18 applications with hidden contextual information inside their texts, which were extracted using NMF topic model. However, to capture the dependency among words, the proposed model employed the use of Bi-LSTM in which the text will be analyzed in both forward and backward directions. We also add another type of features, which are those extracted from word embeddings as the belief is that similar words tend to occur in similar texts.

For the experiment, we built the dataset and the reported experiments showed that the proposed model outperforms traditional models for classifying user reviews and it is able to capture the announced features in review text. In particular, the proposed model accuracy is above 89% and the improvement compared to the proposed base model is significantly high. The experiments showed that NMF model topic model is able to capture hidden topics inside short texts of users' reviews. On the other hand, both Bi-LSTM and word embeddings are extremely useful for handling synonymous and polysemous words.

In the future, we are going to investigate the impact of stemming in improving model performance. In addition, it is really interesting to look into the effect of the dataset word embedding, therefore we plan to train the model using the dataset instead of employing pretrained word embedding models. We are also going to expand the model to Arabic reviews. the latter language is rich in its vocabulary and the use of synonymy and polysemy is widespread.

Acknowledgment

The authors would like to thank The University of Tabuk for providing research support and facilities.

References

[1] Ericsson, "Ericsson mobility report november 2021," november, 2021. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/november-2021

[2] SimpleTexting, "History and evolution of smartphones," July, 2021. [Online]. Available: https://simpletexting.com/where-have-we-come-since-the-first-smartphone

[3] H. S. Maghded, K. Z. Ghafoor, A. S. Sadiq, K. Curran, D. B. Rawat, and K. Rabie, "A novel ai-enabled framework to diagnose coronavirus covid-19 using smartphone embedded sensors: design study," in *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*. IEEE, 2020, pp. 180–187.

[4] Statista, "Number of available applications in the google play store from december 2009 to september 2021," October, 2021. [Online]. Available: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store

[5] T. Manifest, "4 reasons why you need mobile app reviews," October, 2018. [Online]. Available: https://themanifest.com/app-development/blog/benefits-mobile-app-reviews

[6] M. Hatamian, "Engineering privacy in smartphone apps: A technical guideline catalog for app developers," *IEEE Access*, vol. 8, pp. 35 429–35 445, 2020.

[7] ptsecurity, "Mobile application vulnerabilities and threads 2019," July, 2019. [Online]. Available: https://www.ptsecurity.com/upload/corporate/ww-en/analytics/Mobile-Application-Vulnerabilities-and-Threats-2019-eng.pdf

[8] CPO, "How hacker attack mobile apps," April, 2019. [Online]. Available: https://www.cpomagazine.com/cyber-security/how-hackers-attack-mobile-apps

[9] J. Liu, M. K. Sarkar, G. Chakraborty *et al.*, "Feature-based sentiment analysis on android app reviews using sas® text miner and sas® sentiment analysis studio," in *Proceedings of the SAS Global Forum 2013 Conference*, vol. 250. Citeseer, 2013.

[10] L. Zhang, K. Hua, H. Wang, G. Qian, and L. Zhang, "Sentiment analysis on reviews of mobile users," *Procedia Computer Science*, vol. 34, pp. 458–465, 2014.

[11] Y. Zhang, W. Ren, T. Zhu, and E. Faith, "Mosa: A modeling and sentiment analysis system for mobile application big data," *Symmetry*, vol. 11, no. 1, p. 115, 2019.

[12] N. S. Chowdhury and R. R. Raje, "Sers: A security-related and evidence-based ranking scheme for mobile apps," in *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 2019, pp. 130–139.

[13] N. M. Rizk, A. Ebada, and E. S. Nasr, "Investigating mobile applications' requirements evolution through sentiment analysis of users' reviews," in *2015 11th International Computer Engineering Conference (ICENCO)*. IEEE, 2015, pp. 123–130.

[14] M. E. Permana, H. Ramadhan, I. Budi, A. B. Santoso, and P. K. Putra, "Sentiment analysis and topic detection of mobile banking application review," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*. IEEE, 2020, pp. 1–6.

[15] T. Lavanya, M. J. P. JC, and K. Veningston, "Online review analytics using word alignment model on twitter data," in *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1. IEEE, 2016, pp. 1–6.

[16] 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 *2015 IEEE international conference on software maintenance and evolution (ICSME)*. IEEE, 2015, pp. 281–290.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[18] A. Yadav, R. Sharma, and F. H. Fard, "A semantic-based framework for analyzing app users' feedback," in *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2020, pp. 572–576.

[19] F. Kolini and L. Janczewski, "Clustering and topic modelling: A new approach for analysis of national cyber security strategies," 2017.

[20] N. Al Moubayed, D. Wall, and A. S. McGough, "Identifying changes in the cybersecurity threat landscape using the lda-web topic modelling data search engine," in *International Conference on Human Aspects of Information Security, Privacy, and Trust*. Springer, 2017, pp. 287–295.

[21] P. B. P. Reddy and R. Nallabolu, "Machine learning based descriptive statistical analysis on google play store mobile applications," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2020, pp. 647–655.