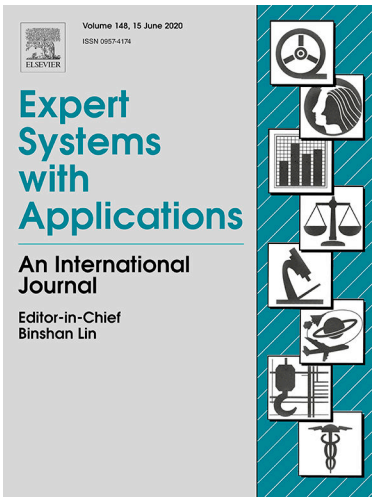

Discrepancy Detection between Actual User Reviews and Numeric Ratings of Google App Store using Deep Learning

Saima Sadiq, Muhammad Umer, Saleem Ullah, Seyedali Mirjalili, Vaibhav Rupapara, Michele NAPPI

PII: S0957-4174(21)00552-2
DOI: https://doi.org/10.1016/j.eswa.2021.115111
Reference: ESWA 115111

To appear in: *Expert Systems with Applications*

Received Date: 17 January 2021
Accepted Date: 21 April 2021

Please cite this article as: Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., NAPPI, M., Discrepancy Detection between Actual User Reviews and Numeric Ratings of Google App Store using Deep Learning, *Expert Systems with Applications* (2021), doi: https://doi.org/10.1016/j.eswa.2021.115111

Discrepancy Detection between Actual User Reviews and Numeric Ratings of Google App Store using Deep Learning

Saima Sadiq[a], Muhammad Umer[a,b,**], Saleem Ullah[a], Seyedali Mirjalili[c,d], Vaibhav Rupapara[e,*] and Michele NAPPI[f,*]

[a]Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, Pakistan
[b]Department of Computer Science & Information Technology, The Islamia University of Bahawalpur, Bahawalpur, 63100, Pakistan
[c]Center for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, Brisbane, QLD 4006, Australia
[d]Yonsei Frontier Lab, Yonsei University, Seoul, Korea
[e]School of Computing and Information Sciences, Florida International University, USA
[f]Department of Computer Science,University of Salerno,Fisciano, Italy

ARTICLE INFO

Keywords:
Review rating prediction
Unbiased rating prediction
Deep learning
CNN
Opinion Mining

ABSTRACT

Nowadays online reviews play a significant role in influencing the decision of consumers. Consumers show their experience and information about product quality in their reviews. Product Reviews from Amazon to Restaurant Reviews from Yelp are facing problems with fake reviews and fake numeric ratings. Online reviews typically consist of qualitative (text format) and quantitative (rating) formats. In the case of Google Play store fake numeric ratings can play a big role in the success of apps. People tend to believe that a high-star rating may be significantly attached with a good review. However, user star level rating information does not usually match with text format of review. Despite many efforts to resolve this issue, Apple App Store and Google Play Store are still facing this problem. This study proposes a novel Google App numeric reviews & ratings contradiction prediction framework using Deep Learning approaches. The framework consists of two phases. In the first phase, the polarity of reviews are predicted using sentiment analysis tool to build ground truth. In the second phase, star ratings are predicted from text format of reviews after training deep learning models on ground truth obtained in the first phase. Experimental results demonstrate that based on actual user reviews the proposed framework significantly predicts unbiased star rating of app.

1. Introduction

Mobile phones are a part of our lives by providing us with countless functionalities to perform daily tasks for example online shopping, getting location information and managing assets. As Android is rapidly developing among mobile frameworks Holla and Katti (2012), Android applications (referred as "mobile apps") are also growing in number and provide free and paid access to users through app stores such as official Google Play Store. On Google Play Store users have to choose from millions of mobile apps for them but sometimes downloaded apps are not useful enough. This online platform allows users to share their reviews. User review information is growing tremendously every day and different trends and statistical patterns can be explored from such data. Previous studies turn out that famous apps receive hundreds of reviews on a daily basis Licorish, Tahir, Bosu, and MacDonell (2015). In the era of huge amounts of user generated data, redundant and irrelevant information impedes users to target right needed information and interfere with their choices X. Liu et al. (2019). Users share their experience and opinion to help other users to make right decisions in the form of online reviews over online platforms. In recent times online reviews are considered a significant asset. To extract knowledge from user opinions is known as opinion mining and also called sentiment analysis. Ding, Liu, and Yu (2008); B. Liu and Zhang (2012); Popescu and Etzioni (2007).

App stores permit users to express their opinions numerically as well as in textual form in ratings and reviews respectively. Empirical studies proved that user reviews in app stores contain important information Tian, Nagappan, Lo, and Hassan (2015). Text form of a review may contain positive, negative or neutral sentiment about specific app posted by a user. These text reviews without considering numeric rating have been widely explored by researchers by performing categorization /classification Panichella et al. (2015) , summarization Anchiêta and Moura (2017), Analyzing Ciurumelea, Schaufelbühl, Panichella, and Gall (2017) and clustering Villarroel, Bavota, Russo, Oliveto, and Di Penta (2016) for market analysis or product feedback review etc.

In addition star rating is considered as an important factor in online reviews as it represents the quality of a topic. Usually users assign rating to a review using star rating by selecting the number of stars ranging from 1 to 5 according to the level of their liking. Higher rating of the star means the user liked it most. When a user needs to choose for an online service or product, he finds help from reviews and ratings of other users. Most people prefer the choice with a high star rating as they consider it to be attached with good

*Corresponding author
**Principal corresponding author
s.kamrran@gmail.com (S. Sadiq); umersabir1996@gmail.com (M. Umer); saleem.ullah@kfueit.edu.pk (S. Ullah); ali.mirjalili@laureate.edu.au (S. Mirjalili); vaibhav.rupapara.sept@gmail.com (V. Rupapara); mnappi@unisa.it (M. NAPPI)
ORCID(s): 0000-0002-2611-3738 (S. Sadiq); 0000-0002-6015-9326 (M. Umer)

reviews. Impact of star rating is also explored by researchers and they also proved its importance in users' choice Du, Rong, Wang, and Zhang (2020). Their experimental results proved that rating improves the prediction helpfulness in prioritizing product reviews.

In Aralikatte, Sridhara, Gantayat, and Mani (2018), authors inspected 8600 reviews of ten trendy android applications and found 20% inconsistency between star ratings and reviews. They also investigated the users' and app developers' points of view about finding a mismatch of review and rating. The authors conducted a survey to know whether users update their ratings after posting problems in the review. Results revealed that both app developers and users feel that it is beneficial to detect this mismatch of review and rating because user's choice to download an app depends upon existing ratings and reviews. They also claimed that this inconsistency affects the download rate of upcoming and small apps. Furthermore, when users find any problem in an app they just post a review but do not update the star rating of the app, which is also a significant cause of mismatch of rating and review.

Nevertheless users' rating information presented on online platforms does not always match with text reviews. An efficient way to sort out this issue is the intelligent system which assists users according to their need and interest. As star rating adds value to the review information and is valuable, it becomes necessary to investigate prediction of automated review rating system in order to predict unbiased rating for review. A classifier can predict rating from text format for reviews and it is known as review rating prediction Tang, Qin, Liu, and Yang (2015). In a nutshell, previous researches use reviews and ratings to perform classification into positive , negative and neutral categories but no work is done to predict biased and unbiased rating using deep learning models.

An efficient machine learning model is required to extract useful information in prediction. Deep learning is based on multi-layered neural architecture and provides stability, scalability and generalization when dealing with big data. It is performing well in multiple domains and is becoming first choice for high prediction accuracy. This study proposes a framework to predict discrepancy between user ratings and reviews using deep learning architecture after review analysis. In order to investigate review rating by considering the embedded features in the text of review. We consider sentiment of the review as one of the most significant features that could be used for analysis to solve this problem in a better way. Choosing this approach would guide to more accurate prediction of rating and help users to make the right decision.

The rest of the paper is organized as: Section 2 presents related work, section 3 discusses material and methods, section 4 presents experiment, section 5 elaborate results and discussion. Finally section 6 concludes the work.

2. Related Work

One of the most common and effective ways to extract useful information out of the user reviews is to conduct sentiment analysis by mining textual information. Sentiment analysis involves classifying structured text into positive, neutral or negative categories to explore the user's opinion about product, service or application Cambria, Das, Bandyopadhyay, and Feraco (2017). A number of research works in the field of sentiment analysis has been performed by using machine learning algorithms on user reviews and achieved promising resultsAhmad, Aftab, Muhammad, and Ahmad (2017); Singla, Randhawa, and Jain (2017).

Many studies have shown that app users and their reviews are key factors for software development Maalej and Nabil (2015). A detailed study containing 290 publications on impact of user involvement is presented by Bano et al.Bano and Zowghi (2015). App reviews pose additional challenges for being three to four times shorter than twitter message in text analysis Jakob, Weber, Muller, and Gurevych (2009). A study conducted by Pagano et al. have shown that 84% reviews contain short text (contain less than 160 characters) Pagano and Maalej (2013). In literature researchers analyzed user reviews in different aspects. Chandy et al. utilized a latent model to classify spamming reviews. Chandy and Gu (2012). Mark et al. investigated to search related reviews.Vu, Nguyen, Pham, and Nguyen (2015). Guzman et al. applied Latent Dirichlet Allocation (LDA) and SentiStrength for sentiment analysis of app reviews.Guzman and Maalej (2014). Martin et al. investigates the sampling bias effects and techniques during app review mining.Martin, Harman, Jia, Sarro, and Zhang (2015). Our work differs in the way that we classify each review sentence by comparing numeric rating into either "Biased rating" or "Unbiased rating" instead of simply "Positive" or "Negative" in terms of emotional perspective.

Dhinakaran et al. Dhinakaran, Pulle, Ajmeri, and Murukannaiah (2018) used uncertainty based active learning techniques. They classified reviews as features, rating, bugs and user experience. They used 4400 reviews in their experiment and considered both binary classification and multi class classification. For binary classification task, they considered review from the corresponding class as positive and review from any other classes as negative and for multi class classification tasks, they considered all four classes. They achieved higher accuracy with active learning. They proved active learners more effective than passive learners.

Suleman et al.Suleman, Malik, and Hussain (2019) predicted app ranking using different machine learning models on Goolgle Play store and different datsets. Tree based algorithm outperformed to predict app ranking in their experiment. Categories of their dataset include app_category, number of downloads, type, size, version, rating are used as input for app ranking prediction. Their machine learning models include DT, LR, SVM, NB, K-mean clustering, KNN, PCA, ANN. Their effort was to improve optimization of app stores,to manage trends and improve rating systems.

Authors in Martens and Johann (2017) experimented on

Apple app reviews to explore the importance of emotional sentiment as an informative feature. They explored how sentiment can play a significant role in the analysis of app reviews of app stores. They claimed that users mostly do not express their emotions in app reviews neither positive sentiment nor negative and rating is the best indicator to view user satisfaction instead of sentiment i.e emoticons etc. Due to weak correlation of sentiment features with reviews in different aspects, when sentiment is calculated it often diverges.

Panichella et al. Panichella et al. (2015) performed review classification based on specific categories that are software maintenance and evolution. They used NLP to detect user intentions in review. They also investigated how much NLP, Text Analysis and Sentiment Analysis based features play a role in finding relevance of reviews. They chose the seven most popular apps of the year 2013 from different categories. They proved that by combining NLP, TA and SA results are improved with 75% Precision, and 74% Recall and individually without combining features, precision was 70% and recall was 67%.

Review classification is performed by many researchers to understand the user's needs. Pratama, Utami, and Sunyoto (2019) applied lexicon based approach using domain specific features such as special words and star ratings. They claimed that these features improved the performance of the classifier. They experimented on Apple App store and Google Play store. They extracted features from star ratings and used SentiWordNet as a lexical approach. They classified reviews in four classes (Positive, Neutral, Negative and Average). They achieved the highest accuracy and F1 score with 60% by combining SentiWordNet and Star rating.

Islam et al. Islam (2014) also included star rating to check sentiment polarity. They observed that sometimes rating has much difference than reviews of users. To solve this problem they proposed a uniform approach by considering star rating and numeric rating. They calculated the average of star rating and numeric rating to find sentiment polarity value. They proposed a diverse approach that can be implemented for sentiment classification of different domains as reviews vary among different apps.

Shah et al. Shah, Sirts, and Pfahl (2018) used lexical features to perform classification of app reviews. The main hypothesis of their research was to compare a simple BOW model with more complex linguistic features based model. They used reviews of 17 apps of different categories such as books, music, communication and games. They assigned mutually exclusive labels that are feature request, bug report, feature evaluation, praise and other. Then they used both BOW features and linguistic features to train the Maximum Entropy model. Then on the second step, they also compared their simple BOW model with a deep learning based CNN model. CNN model performed slightly worse than their simple BOW model. They concluded that use of more complex linguistic features did not improve the performance and there is no need to use a more expensive CNN model for sentiment classification of reviews.

Deep learning models have been explored in text classification Sadiq et al. (2021), image classification Umer, Sadiq, et al. (2020), gait phases and event recognition Di Nardo, Morbidoni, Cucchiarelli, and Fioretti (2020), speech recognition Yang, Li, Wang, and Tang (2019), estimating click through rate of advertisement Qiang et al. (2020), early fault detection Mao, Zhang, Tian, and Tang (2020) and showing robust results. Authors in Zhou, Chen, and Wang (2013) performed sentiment classification using active deep network that is a semi-supervised approach. It is built using restricted Boltzmann machines with unsupervised learning using unlabeled reviews and labeled reviews. Exponential loss with gradient decent is used to tune the model. An information active deep network is proposed for the labeled review selection by applying information density. Experiments are performed on five different datasets for sentiment classification. Results reveal that active deep network and information active deep network achieve higher results than that of classical semi-supervised learning models as well as deep learning models applied for sentiment classification. The authors proposed SenticNet by integrating logical reasoning in deep learning models for sentiment analysis Cambria, Li, Xing, Poria, and Kwok (2020). Authors proposed attention based (CNN-RNN) model for sentiment analysis Basiri, Nemati, Abdar, Cambria, and Acharya (2020). A stacked-ensemble model based on deep learning was proposed to detect intensity of sentiments and emotions Akhtar, Ekbal, and Cambria (2020).

Reviews and ratings of google play store have been analyzed for category prioritization in Mahmud, Niloy, Rahman, and Siddik (2019). Recently in our previous task Umer, Ashraf, Mehmood, Ullah, and Choi (2020) different features TF-IDF (Unigram, Bigram and Trigram) and ensemble learning models have been explored to classify Google Play Store user ratings using ensemble machine learning models. TextBlob analysis is performed to determine sentiment of review and then compared with star rating value to check its authenticity. Machine learning models achieved reasonable performance. Machine learning based models require massive and good quality of data for training with reasonable feature extraction technique. In some cases data is imbalanced and models cannot train well and eventually become susceptible to errors. While deep learning models make use of word embedding technique to represent learned features and achieve good results in challenging natural language processing problems. This the reason of the use of deep learning models in this study.

The analysis of existing approaches reveals that different techniques have been applied to explore user reviews in many aspects. Several models have been designed for classification, summarizing, analyzing and predicting star rating using different features. However, rarely any research direction found to explore text user reviews and star rating for biased rating prediction. This study attempts to classify rating as biased or unbiased based on actual user reviews using a deep learning model.

Table 1
Attributes of the used dataset

Attribute	Description
App-id	The unique app-id assigned from Google play store.
App-name	It shows the actual name of Google play store app.
App-category	Represents category of Google play store to which the app belongs.
App-rating	The rating given by an individual user to Google play store app.
App-review	The review given by individual user for a specific app.

3. Material & Methods

This section describes a dataset, preprocessing steps and deep learning framework for numeric rating prediction. Figure 4 presents the proposed architecture used in our experiment for review rating prediction.

3.1. Dataset Description

Dataset of 14 different categories of mobile applications is derived from Google Play Store. Dataset containing 502, 658 records is scrapped by using BeautifulSoup (BS) web scraper. Different attributes of the dataset involve App_id, Appname, App_category, App_review and App_rating as shown in Table 1. Each app category contains at least four thousand reviews in the dataset. Figure 1 represents the name and percentage ratio of each app. From the Google Play Store dataset we analyzed App_review and App_rating in our research.

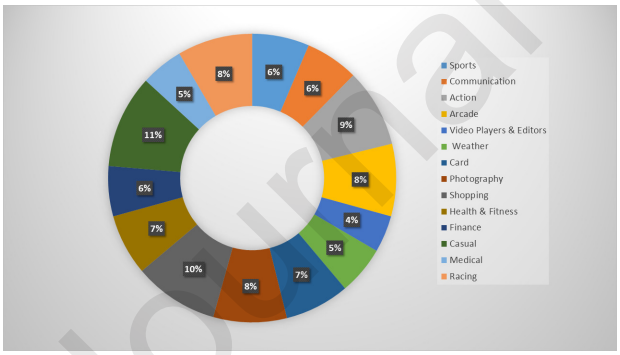

Figure 1: Percentage of each category in the dataset.

3.1.1. Visualization of dataset

We also visualize the dataset to observe numeric star ratings assigned by users. It can be observed clearly in Figure 2 that users assigned rating 5 more frequently. It seems that this rating can be fake or biased. Numeric rating assigned to apps by anonymous users is a significant problem to be analyzed Aralikatte et al. (2018). We noticed the frequency of numeric star rating in each category and find a casual category of mobile apps with high numeric rating and video players and editors with lowest numeric rating assigned by user as shown in Figure 3.

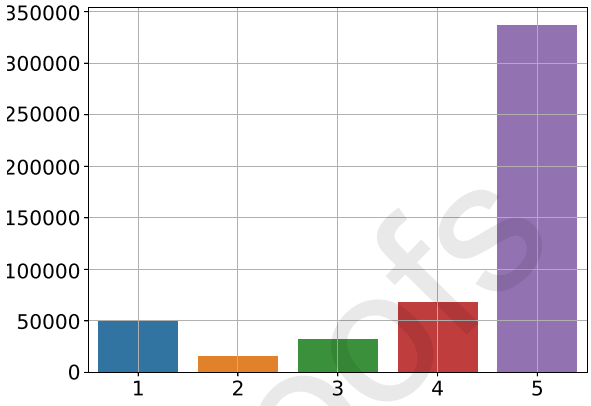

Figure 2: Most frequent ratings from users.

3.2. Preprocessing

To prepare data for training phase unstructured and noisy data need preprocessing. Noise is something that does not contain useful information in prediction and interfere next process by increasing overhead of training by increasing time of training and often reduce accuracy e.g HTML tags, duplicate punctuation and stop words and white spaces etc. Literature reveals that preprocessing play a significant role in accurate prediction and leads to good results Feldman and Sanger (2007).

Therefore, various steps of preprocessing have been performed before starting the training phase. We transformed all text to lowercase, then in the second step we removed HTML tags and stopwords. Then performed stemming to convert words into simpler form and then tokenized remaining text to feed deep learning models for training.

3.3. Deep Learning Models

This research used deep learning models to predict numeric ratings of Google play store apps using actual user reviews. These models have been extensively used in text classification tasks and achieved great success in sentiment classification Rebiai, Andersen, Debrenne, and Lafargue (2019). Efficiency of deep learning architecture is incumbent on its capability to learn the way of representation of text from data without rigorous feature engineering. These deep learning approaches have the ability to identify semantic relationships of text in a better way than conventional machine learning approaches. We evaluate the performance of deep learning models and discuss it in brief.

CNN achieved remarkable results in text classification task Kim (2014). First input layer of CNN, is composed of word2vec embedding, process data at first step. Then Convolutional layer with variable filter sizes perform classification and followed by pooling layer and then output is

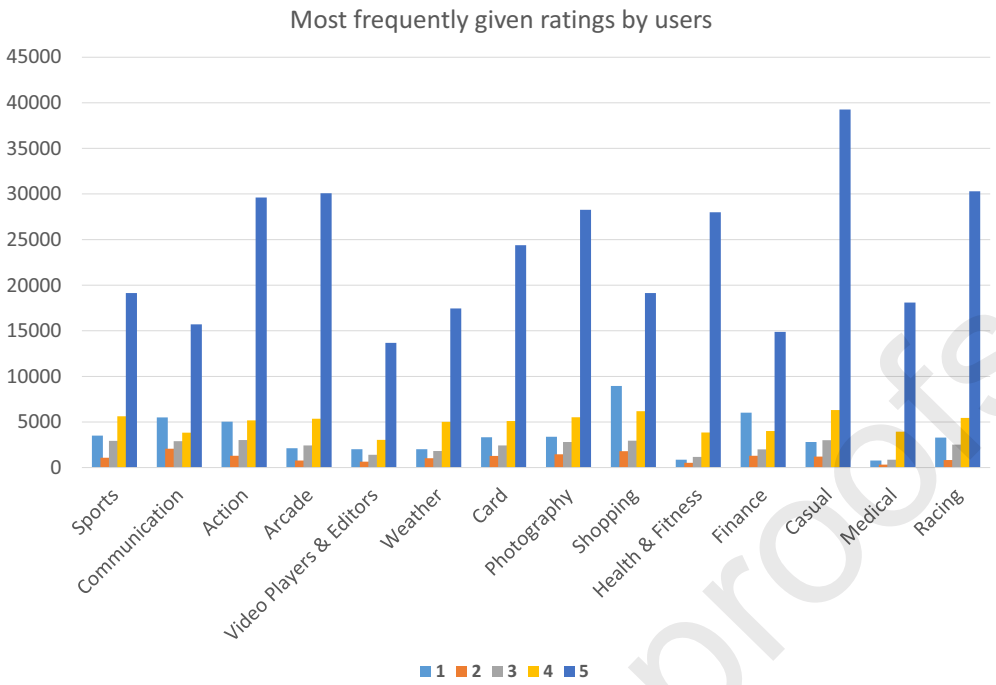

Figure 3: Each category ratings from users.

obtained with SoftMax layer. CNN has been performing efficiently in text classification but it does not deal with long term dependencies of words in a sentence just focusing on local features.

In contrast, RNN Elman (1990) is a specialized neural network that uses recurrent connection to remember everything. RNN takes current input and also takes in focus previous inputs. Such ability makes it very efficient in sequential classification. But RNN also faces a problem of vanishing gradients that hinders it to learn long term dependencies and their correlation Hochreiter, Bengio, Frasconi, Schmidhuber, et al. (2001).

LSTM, that is a variant of RNN designed by Hochreiter and Schmidhuber (1997) to deal with the challenges of long term dependencies. It has a better capability to model sequential data by learning long term dependencies using a gating mechanism. LSTM obtained output from the current iteration to use as input in the next iteration. Its architecture considers the backward context of text and classification is finally performed by last layers.

BiLSTM also known as bi-directional LSTM is proposed by Dyer, Ballesteros, Ling, Matthews, and Smith (2015). The idea behind its design was to provide a solution for temporal flow of information in both directions to deal with faster learning on the problem. It trains neural networks by looking at both sequences of context of text from front(future context) to back (past context).

GRU Cho, Van Merriënboer, Bahdanau, and Bengio (2014) is also an improved variant of LSTM in terms of performance and network structure. Both GRU and LSTM deal with vanishing gradient problems to learn long term dependencies but differ in a structure. GRU has two gates (update & reset) and LSTM has three gates (input, output & forget). GRU is unidirectional and controls information flow of previous iteration when dealing with current input using reset gate. While the update gate combines previous iteration with current iteration and passes it to the next step.

3.4. Proposed Framework

This section describes all phases of the proposed approach framework and its modules utilized in the experiment. Figure 4 elaborates the proposed framework. At first, review text has been cleaned by preprocessing and then the polarity of reviews is predicted. The division of data is 70% for training while 30% for testing. The training has been performed on "unbiased" textual data to predict the actual rating of apps based on "unbiased" reviews and ratings. Then deep learning models have been applied to predict numeric ratings from reviews. The proposed approach has two phases, each phase has been explained separately.

3.4.1. Phase I: Polarity Prediction

First phase consists of a polarity prediction model in which the polarity of review is predicted whether it is positive or negative. Reviews are labelled according to formula;

Google Apps Store

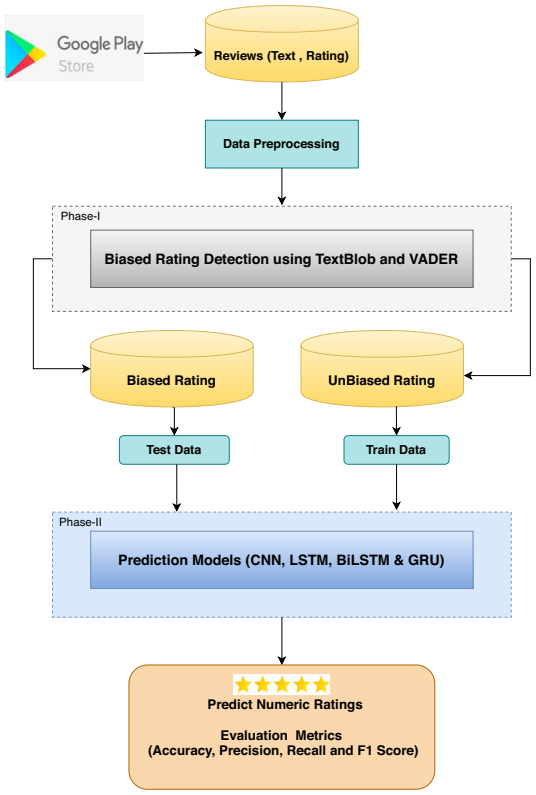

Figure 4: Architecture of the proposed Framework.

$$Review_{(o-n)} = \begin{cases} Negative, & 0 \text{ to } \frac{n}{2} \\ Positive, & \frac{n}{2}+1 \text{ to n} \end{cases} \quad (1)$$

Where n is the level of rating or number of stars which are from 1 to 5. The reviews with value equal to or greater than 3 belong to positive class such as 3,4 & 5 while reviews with less rating value (less than 3 stars (1 & 2) belongs to negative class. A Python library TextBlob (sentiment function) returns two properties, subjectivity and polarity. Polarity is float and its range is [-1,1] where 1 represents positive statement and -1 represents a negative statement. VADER Hutto and Gilbert (2014) is a sentiment analysis tool which is lexicon and rule based. It gives negativity or positively scores as well as describing how negative or positive a sentiment is. These tools assist us to check the ground truth to calculate biasness of reviews. We used 2 tools TextBlob and VADER to analyze polarity of reviews. Flowchart of Biased rating detection method using TextBlob and VADER in shown in Figure 5.

TextBlob finds polarity of reviews in two classes positive and negative and VADER also analyzes sentiment intensity. The polarity of each review is calculated in the dataset and compared it to its relevant star numeric rating. If the numeric rating is greater than 3 and polarity value is less than -0.5, then such rating will be considered as biased and otherwise unbiased.

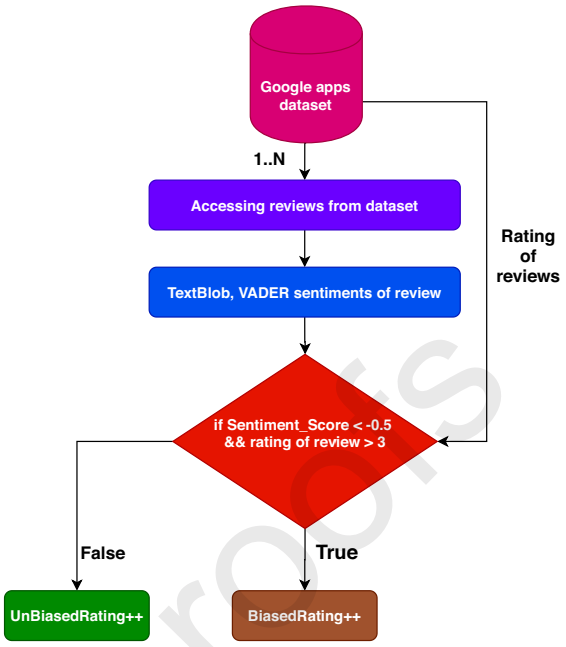

Figure 5: Flow chart of Biased Rating Detection Method.

from textblob import TextBlob
t = TextBlob("She is not a trustworthy person. i hate her")
t.sentiment shows
Sentiment(polarity=-0.8, subjectivity=0.9)

from nltk.sentiment.vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()
a="She is not a trustworthy person. i hate her"
sid.polarity_scores(a) shows
'neg': -0.9, 'neu': 0.0, 'pos': 0.0

3.4.2. Phase II: Rating Prediction

The second Phase consists of deep learning models utilized to predict numeric rating from unbiased user reviews obtained in Phase-I. Results of these deep learning models are compared with the aggregate rating of those apps. Original rating of Google Play Store apps is the aggregate rating of apps. Main objective of this experiment is to find discrepancies between user reviews and rating.

The advantage of the proposed framework is the use of the word embedding technique to build the semantic similarity relationship between the words and make use of deep learning models especially CNN which performs better by using multiple convolutional operations with max-pooling as a supporting layer. The challenge of the proposed framework is the dataset that we used in this research is imbalanced in terms of few apps categories. The sequence generation prediction models like LSTM, RNN, GRU, and Bi-LSTM do not perform well on the imbalanced dataset Li (2017).

Table 2
Summary of the hyper-parameter values for all models

Parameter	Value
Embedding dimension	(100)
Batch size	64
Pooling	2*2
Epochs	13
Optimizer	SGD
Function	Categorical cross entropy

Table 3
Layers structure of the DL models

CNN	LSTM	BiLSTM	GRU	RNN
Conv (7×7, @64) Max Pooling (2×2) Conv (7×7, @64) GlobalMax Pooling (2×2) Dropout (0.5) Dense (32 neurons) Softmax(5)	LSTM (100 neurons) Dropout (0.5) Dense (32 neurons) Softmax(5)	BiLSTM (100 neurons) Dropout (0.5) Dense (32 neurons) Softmax(5)	GRU (100 neurons) Dropout (0.5) Dense (32 neurons) Softmax(5)	RNN (100 neurons) Dropout (0.5) Dense (32 neurons) Softmax(5)

4. Experiment

In order to estimate performance of the models, extensive experiments have been performed on the Google Play Store dataset with state-of-the-art deep learning models that are CNN, RNN, LSTM, BiLSTM and GRU. The reason for choosing these deep learning models is to validate the proposed framework in all aspects of training like sequence generation and extraction of significant features. This section represents the experimental setting and evaluation metrics in detail.

4.1. Experimental Setup

In phase-I Biased rating detection algorithm is implemented in Python. In phase-II for numeric rating prediction sigmoid has been used as an activation function on output layer to perform prediction task, Cross entropy has been utilized as loss function to measure difference between predicted and actual values. To optimize parameters of models, Adam optimizer has been used. Detail of hyper parameter tuning is shown in Table 2 and layered structure is in Table 3. Layered architecture of the proposed CNN model is presented in Figure 6.

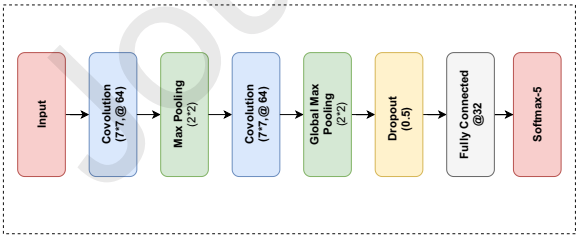

Figure 6: Layered architecture of the proposed CNN-based Deep learning model.

Early stopping and checkpoint call back technique has been applied at the end of each epoch. When the desired metric no longer keeps on improving, early stopping will automatically stop to train the model. Contrary, checkpoints save weights of the best epoch by calling it back.

Table 4
Examples of biased app rating.

User review	User app rating
Movies take so much to get buffered. Ads come up after every 5min	4
Since the recent update, it is impossible to pass level 620.	
The balls glitch through the cushions. You literally can't complete it	5
I installed this application but it will suck every time on Google map	
location tracing layout didn't detected my location kindly fix this issue thanks	5
The graphics of the game are little bit blurry	4
So booring game not nice but ok	4

Table 5
Biased ratings calculated using TextBlob.

App rating	Rating count	Biased count	Ground truth
1	49,608	-	49,608
2	15,605	-	15,605
3	32,238	14,725	17,513
4	68,419	20,479	47,940
5	336,781	89,034	247,747

5. Results and Discussion

To predict the authenticity of numeric ratings using deep learning models, 14 categories of apps, every category containing 12 apps were selected. Dataset consists of a total 168 apps. App categories selected on the basis of app popularity among users, as represented on aggregate rating on Google Play Store. Apps with a high number of reviews and highest collection of ratings are chosen in each category. Figure 3 presents the number of apps in each category and the selected categories of apps and in the dataset. It also specifies numeric ratings obtained by each app. Numeric rating scores assigned by users to apps of Google App Store may be exaggerated and often do not match with the reviews. Our effort in this work is to find this discrepancy.

The proposed framework comprises of two phases. Phase-I is the polarity (sentiment) prediction phase. Numeric ratings given by the users to the Google Play Store apps may be overrated or biased just to get the attraction of new users. Contradiction between user reviews and numeric rating can be seen clearly with few examples in Table 4.

To justify this hypothesis, there should be a systematic way. In the phase-I we use TextBlob and VADER to find polarity of each review to find out mismatch between review text and numeric rating. If the polarity value obtained is less than -0.5 and rating given by user is greater than 3 then such rating is considered as biased otherwise it is unbiased. Result of phase-I is presented in Table 5 for TextBlob and Table 6 for VADER. In the experiment we find mismatch of review and rating in app rating of 3, 4 and 5 star value that also reveal the importance of higher star rating value. Re-

Table 6
Biased ratings calculated using VADER.

App rating	Rating count	Biased count	Ground truth
1	49,608	-	49,608
2	15,605	-	15,605
3	32,238	15,225	17,013
4	68,419	21,279	47,140
5	336,781	89,307	247,474

Table 7
Classification result of all deep learning models.

Models	Accuracy	Precision	Recall	F1-Score
CNN	89%	82%	89%	86%
LSTM	87%	79%	83%	81%
RNN	83%	78%	81%	80%
BiLSTM	86%	81%	85%	83%
GRU	77%	74%	79%	77%

sults reveal that TextBlob shows total 124,238 ratings are biased out of 502,658 ratings given by users to specific categories of apps which shows that biased rating percentage is 24.7162% and unbiased rating is 75.2838%. VADER results shows that 125,811 ratings are biased out of 502,658 ratings given by users to specific categories of apps which represent bias rating percentage is 25.0291% and unbiased rating is 74.9708%. VADER shows a more biased count of reviews than TextBlob. VADER performs better than TextBlob in sentiment analysis of social media text as it is based on lexicon and it is faster than Textblob Bonta and Janardhan (2019). We consider VADER results as ground truth in our experiment.

In phase-II we take unbiased user reviews predicted by VADER of Google app store for training of deep learning models to predict the numeric rating. After that we compared their results with the aggregate rating (Actual rating of apps on Google Play Store) of those apps. Reason for this comparison is to find mismatch between user reviews and user ratings. Table 7 presents the classification result of all deep learning models. It can be observed that CNN outperformed among all other deep learning models such as LSTM, RNN, BiLSTM and GRU with 89% classification accuracy. CNN also showed robust results in terms of precision, recall and f1 score with 82%, 89% and 86% respectively.

Performance of the best performing model CNN is evaluated on each app category. Experiment reveals that prediction accuracy can be increased by training models on each category separately. Table 8 presents the results attained by training CNN on each category of Google play store apps. For the "Health & Fitness" category CNN achieved highest accuracy and precision 94% and 90% respectively, which is higher than accuracy and precision obtained by training on all combined categories. CNN model has also achieved higher results regarding recall and F1 Score in individual app categories.

Table 9 demonstrate predicted numeric ratings from the deep learning models used in the experiment. It can be seen clearly in Table 9, for 8 out of 14 categories CNN, the best performing model, predict lowest numeric rating value. While for "Gun Shot", "Teen Patti Gold", "B612" and "Pharmapedia Pakistan" LSTM predict lowest rating. For "MX Player", "Seven" and "Phone Pc" BiLSTM predict lowest rating. RNN and GRU predict high star rating values but these ratings are still lower than aggregate ratings. CNN performed well and give accurate predictions of numeric rating of application of all categories. Results obtained from extensive deep learning models seem useful in the evaluation of review and rating. Results also prove that given numeric ratings and reviews are not similar as ratings predicted by all deep learning models are lower than aggregate rating which proves clear biases in aggregate rating. Identification of contradictions and mismatch observed form deep learning model results can be used to help users for app selection and companies to make business decisions.

The accuracy of all deep learning models especially CNN actually represents a mismatch between user review and numeric rating. Result shows that the numeric rating of users in the app store is higher than obtained from the CNN model.

It is analyzed from Table 9 that the numeric rating given by the users on Google playstore is approximately 25% higher than the ratings predicted by the deep learning classifiers. It can also be observed in Table 9 that the CNN model with word embedding features performed excellently as compared to the other deep learning model in all categories application review rating prediction. The reason for CNN performance is that all other deep learning models used in this research work are work on the principle of sequence generation. CNN makes use of multiple convolutional filters with max-pooling supporting layers to extract the significant features and make predictions on the basis of those significant features. Sequence generation is used in cases where the dataset is balanced in all classes. While the dataset we used in this research work is a little bit imbalanced in terms of few apps categories. That's the big reason for CNN performance and it proves the discrepancy between the user given numeric rating and reviews.

6. Conclusion

Online reviews play a vital role in consumer purchasing decisions by providing them information about quality of product or service. However, it is very hard for consumers to find genuine and unbiased information from large amounts of reviews. Review is qualitative measure and rating is quantitative measure. Reviews of Google Play Store can be biased and overrated to get user attention.

In this study, a novel framework was proposed to predict numeric ratings of Google apps from user reviews using a deep learning model. The framework consisted of two phases. The first phase was to find the polarity of each review using TextBlob and VADER to calculate ground truth to find mismatch between user reviews and aggregate numeric ratings. VADER showed that 25.03% of user ratings are biased, and in the second phase rating was predicted from text

Table 8
Accuracy of application categories using CNN.

App category	Number Of reviews	Accuracy	Precision	Recall	F1-Score
Sports	32280	82%	78%	84%	81%
Communication	30000	87%	75%	83%	79%
Action	44141	91%	83%	87%	85%
Arcade	40751	92%	87%	94%	90%
Video players & editors	20781	84%	76%	79%	79%
Weather	27324	81%	78%	87%	82%
Card	36520	86%	82%	89%	85%
Photography	41440	88%	86%	93%	93%
Shopping	47840	82%	75%	80%	77%
Health & fitness	34415	94%	90%	91%	90%
Finance	28220	83%	76%	84%	79%
Casual	52560	89%	86%	85%	85%
Medical	24002	90%	84%	87%	85%
Racing	42384	89%	87%	84%	85%

Table 9
Numeric rating prediction using deep learning classifiers.

App Name	App reviews	CNN	LSTM	RNN	BiLSTM	GRU	Aggregate rating
Billiards City	4,480	4.01	4.21	4.42	4.14	4.46	4.47
UC-Browser	3,000	3.64	3.99	4.10	3.67	3.89	4.20
Gun Shot	3,000	3.30	3.22	3.43	3.47	3.31	3.69
Temple Run 2	3,000	3.70	3.85	3.93	3.99	3.74	4.45
MX Player	3,000	3.29	2.98	3.06	2.83	2.95	3.92
Weather & Clolck Widget	4,480	2.95	3.04	3.24	3.04	3.13	4.33
Teen Pati Gold	4,480	3.81	3.79	3.99	3.09	3.83	4.61
B612	4,000	3.46	3.39	3.75	3.91	3.53	4.64
Flipkart	4,480	2.33	2.46	2.70	2.45	2.60	3.80
Seven	4,424	2.92	3.18	3.37	2.90	3.51	4.43
PhonePe	4,000	2.78	2.97	3.13	2.78	3.18	3.98
Candy Crush Saga	4,480	3.73	3.87	3.94	3.83	3.90	4.56
Pharmapedia Pakistan	4,133	3.27	3.20	3.29	3.48	3.30	4.72
Beach Buggy Racing	4,480	3.81	4.02	4.13	3.84	4.17	4.61

of reviews based on ground truth calculated in phase-I. Extensive experiments were conducted to test and analyze the performance of deep learning models. Results demonstrated that CNN outperformed in numeric rating prediction and we found aggregated ratings higher than ratings predicted by deep learning models. Hypothesis also proved right that sentiment numeric rating may be biased and mismatch with user reviews.

Conflict of Interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results

CRediT authorship contribution statement

Saima Sadiq: Writing - Original draft preparation, Conceptualization of this study. **Muhammad Umer:** Writing - Original draft preparation, Conceptualization of this study, Methodology, Software. **Saleem Ullah:** Final manuscript review, Project Supervision. **Seyedali Mirjalili:** Final manuscript review. **Vaibhav Rupapara:** Project Supervision. **Michele NAPPI:** Final manuscript review, Funding Acquisition.

References

Ahmad, M., Aftab, S., Muhammad, S. S., & Ahmad, S. (2017). Machine learning techniques for sentiment analysis: A review. *Int. J. Multidiscip. Sci. Eng*, *8*(3), 27.

Akhtar, M. S., Ekbal, A., & Cambria, E. (2020). How intense are you? predicting intensities of emotions and sentiments using stacked ensemble [application notes]. *IEEE Computational Intelligence Magazine*, *15*(1), 64–75.

Anchiêta, R. T., & Moura, R. S. (2017). Exploring unsupervised learning towards extractive summarization of user reviews. In *Proceedings of the 23rd brazillian symposium on multimedia and the web* (pp. 217–220).

Aralikatte, R., Sridhara, G., Gantayat, N., & Mani, S. (2018). Fault in your stars: an analysis of android app reviews. In *Proceedings of the acm india joint international conference on data science and management of data* (pp. 57–66).

Bano, M., & Zowghi, D. (2015). A systematic review on the relationship between user involvement and system success. *Information and Software Technology*, *58*, 148 - 169. Retrieved from http://www.sciencedirect.com/science/article/pii/S0950584914001505 doi:

https://doi.org/10.1016/j.infsof.2014.06.011

Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2020). Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, *115*, 279–294.

Bonta, V., & Janardhan, N. K. N. (2019). A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology*, *8*(S2), 1–6.

Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2017). Affective computing and sentiment analysis. In *A practical guide to sentiment analysis* (pp. 1–10). Springer.

Cambria, E., Li, Y., Xing, F. Z., Poria, S., & Kwok, K. (2020). Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. In *Proceedings of the 29th acm international conference on information & knowledge management* (pp. 105–114).

Chandy, R., & Gu, H. (2012, 04). Identifying spam in the ios app store. In (p. 56-59). doi: 10.1145/2184305.2184317

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Ciurumelea, A., Schaufelbühl, A., Panichella, S., & Gall, H. C. (2017). Analyzing reviews and code of mobile apps for better release planning. In *2017 ieee 24th international conference on software analysis, evolution and reengineering (saner)* (pp. 91–102).

Dhinakaran, V. T., Pulle, R., Ajmeri, N., & Murukannaiah, P. K. (2018). App review analysis via active learning: reducing supervision effort without compromising classification accuracy. In *2018 ieee 26th international requirements engineering conference (re)* (pp. 170–181).

Di Nardo, F., Morbidoni, C., Cucchiarelli, A., & Fioretti, S. (2020). Recognition of gait phases with a single knee electrogoniometer: A deep learning approach. *Electronics*, *9*(2), 355.

Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining* (pp. 231–240).

Du, J., Rong, J., Wang, H., & Zhang, Y. (2020). Helpfulness prediction for online reviews with explicit content-rating interaction. In *International conference on web information systems engineering* (pp. 795–809).

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., & Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press.

Guzman, E., & Maalej, W. (2014). How do users like this feature? a fine grained sentiment analysis of app reviews. In *2014 ieee 22nd international requirements engineering conference (re)* (p. 153-162).

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.* A field guide to dynamical recurrent neural networks. IEEE Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Holla, S., & Katti, M. M. (2012). Android based mobile application development and its security. *International Journal of Computer Trends and Technology*, *3*(3), 486–490.

Hutto, C. J., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international aaai conference on weblogs and social media.*

Islam, M. R. (2014). Numeric rating of apps on google play store by sentiment analysis on user reviews. In *2014 international conference on electrical engineering and information & communication technology* (pp. 1–4).

Jakob, N., Weber, S., Muller, M.-C., & Gurevych, I. (2009, 01). Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. *International Conference on Information and Knowledge Management, Proceedings*. doi: 10.1145/1651461.1651473

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Li, H. (2017, 09). Deep learning for natural language processing: advantages and challenges. *National Science Review*, *5*(1), 24-26. Retrieved from https://doi.org/10.1093/nsr/nwx110 doi: 10.1093/nsr/nwx110

Licorish, S. A., Tahir, A., Bosu, M. F., & MacDonell, S. G. (2015). On satisfying the android os community: User feedback still central to developers' portfolios. *2015 24th Australasian Software Engineering Conference*, 78-87.

Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data* (pp. 415–463). Springer.

Liu, X., Su, X., Ma, J., Zhu, Y., Zhu, X., & Tian, H. (2019). Information filtering based on eliminating redundant diffusion and compensating balance. *International Journal of Modern Physics B*, *33*(13), 1950129.

Maalej, W., & Nabil, H. (2015). Bug report, feature request, or simply praise? on automatically classifying app reviews. In *2015 ieee 23rd international requirements engineering conference (re)* (p. 116-125).

Mahmud, O., Niloy, N. T., Rahman, M. A., & Siddik, M. S. (2019). Predicting an effective android application release based on user reviews and ratings. In *2019 7th international conference on smart computing & communications (icscc)* (pp. 1–5).

Mao, W., Zhang, D., Tian, S., & Tang, J. (2020). Robust detection of bearing early fault based on deep transfer learning. *Electronics*, *9*(2), 323.

Martens, D., & Johann, T. (2017). On the emotion of users in app reviews. In *2017 ieee/acm 2nd international workshop on emotion awareness in software engineering (semotion)* (pp. 8–14).

Martin, W., Harman, M., Jia, Y., Sarro, F., & Zhang, Y. (2015). The app sampling problem for app store mining. In *2015 ieee/acm 12th working conference on mining software repositories* (p. 123-133).

Pagano, D., & Maalej, W. (2013, 07). User feedback in the appstore: An empirical study.. doi: 10.1109/RE.2013.6636712

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)* (p. 281-290).

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).

Popescu, A.-M., & Etzioni, O. (2007). Extracting product features and opinions from reviews. In *Natural language processing and text mining* (pp. 9–28). Springer.

Pratama, B. T., Utami, E., & Sunyoto, A. (2019). The impact of using domain specific features on lexicon based sentiment analysis on indonesian app review. In *2019 international conference on information and communications technology (icoiact)* (pp. 474–479).

Qiang, B., Lu, Y., Yang, M., Chen, X., Chen, J., & Cao, Y. (2020). sdeepfm: Multi-scale stacking feature interactions for click-through rate prediction. *Electronics*, *9*(2), 350.

Rebiai, Z., Andersen, S., Debrenne, A., & Lafargue, V. (2019). Scia at semeval-2019 task 3: sentiment analysis in textual conversations using deep learning. In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 297–301).

Sadiq, S., Mehmood, A., Ullah, S., Ahmad, M., Choi, G. S., & On, B.-W. (2021). Aggression detection through deep neural model on twitter. *Future Generation Computer Systems*, *114*, 120–129.

Shah, F., Sirts, K., & Pfahl, D. (2018, 01). Simple app review classification with only lexical features. In (p. 112-119). doi: 10.5220/0006855901120119

Singla, Z., Randhawa, S., & Jain, S. (2017). Sentiment analysis of customer product reviews using machine learning. In *2017 international conference on intelligent computing and control (i2c2)* (pp. 1–5).

Suleman, M., Malik, A., & Hussain, S. S. (2019). Google play store app ranking prediction using machine learning algorithm. *Urdu News Headline, Text Classification by Using Different Machine Learning Algorithms*, 57.

Tang, D., Qin, B., Liu, T., & Yang, Y. (2015). User modeling with neural network for review rating prediction. In *Twenty-fourth international joint conference on artificial intelligence.*

Tian, Y., Nagappan, M., Lo, D., & Hassan, A. E. (2015). What are the characteristics of high-rated apps? a case study on free android applications. In *2015 ieee international conference on software maintenance and evolution (icsme)* (pp. 301–310).

Umer, M., Ashraf, I., Mehmood, A., Ullah, D. S., & Choi, G. S. (2020, 07). Predicting numeric ratings for google apps using text features and ensemble learning. *ETRI Journal*. doi: 10.4218/etrij.2019-0443

Umer, M., Sadiq, S., Ahmad, M., Ullah, S., Choi, G. S., & Mehmood, A. (2020). A novel stacked cnn for malarial parasite detection in thin blood smear images. *IEEE Access*, *8*, 93782–93792.

Villarroel, L., Bavota, G., Russo, B., Oliveto, R., & Di Penta, M. (2016). Release planning of mobile apps based on user reviews. In *2016 ieee/acm 38th international conference on software engineering (icse)* (pp. 14–24).

Vu, P. M., Nguyen, T. T., Pham, H. V., & Nguyen, T. T. (2015). Mining user opinions in mobile app reviews: A keyword-based approach (t). In *2015 30th ieee/acm international conference on automated software engineering (ase)* (p. 749-759).

Yang, L., Li, Y., Wang, J., & Tang, Z. (2019). Post text processing of chinese speech recognition based on bidirectional lstm networks and crf. *Electronics*, *8*(11), 1248.

Zhou, S., Chen, Q., & Wang, X. (2013). Active deep learning method for semi-supervised sentiment classification. *Neurocomputing*, *120*, 536–546.